Evaluation and Comparison of Scheduling Strategies for the Scheduling of Electric Vehicles at Capacity-Constrained Charging Stations

Ashkan Amouzandeh[1] , **Valentin Robu**[1]
[1]TU Delft

Abstract

As the popularity of electric vehicles (EVs) increases, congestion at charging becomes a more imminent problem. Congestion at a charging station can lead to long waiting queues and failure of EV owners to charge their vehicles fully before their departure from the station. To combat this issue, this paper explores several candidate scheduling strategies that can be applied for the charging prioritization of EVs at a single station. Through extensive simulations, the efficacy of these strategies is studied under three performance metrics. From the set of strategies studied, we find that earliest deadline first (EDF) and shortest job first (SJF) are the best options in the case that adherence to deadline or a shorter waiting time is most valued, respectively.

1 Introduction

Electric vehicles (EVs)[1] have experienced a rise in popularity over the past few years. Their popularity is expected to further increase in the foreseeable future as well [Muratori *et al.*, 2021]. This rise in popularity is due to, amongst other factors, technological progress and maturation, environmental benefits, decrease in battery costs, greater availability of charging infrastructure and consumer acceptance [Muratori *et al.*, 2021].

The increased adoption of EVS, combined with the fact that EVs take a significantly longer time to charge than the fueling of traditional combustion engine gasoline vehicles ([Qin and Zhang, 2011]), might possibly lead to congestion in the charging infrastructure, more specifically, at the public charging stations. To combat this issue, [De Weerdt *et al.*, 2015] proposes an Intention-Aware Routing System (IARS): "A novel navigation system that predicts congestion at charging stations based on dynamic information about current and future demand for charging". Comparing this system with state-of-the-art benchmark routing algorithms, they show, in some cases, an improvement of over 80% in waiting times at charging stations and a more than 50% reduction in overall journey times.

There is, however, still some further work to be done on this matter, the first of which is a principled comparison between IARS and reservation-based systems. Secondly, the authors would like to combine their work with a pricing model to open the possibility for finding more efficient solutions. Lastly, and most importantly in the context of this paper, the order in which the vehicles can be best prioritised for charging at the individual charging stations is not yet explored.

This paper, therefore, contributes to the aforementioned work by exploring the possible scheduling strategies for EVs at an individual charging station. The strategies (described in Section 3) are studied and compared via extensive simulations. More precisely, we use several performance metrics to asses the efficacy of the strategies and conclude which strategy is best suited for which performance criterion.

The scheduling of EVs at charging stations has been studied before, but often with different objectives or under different metrics. For instance, in [Subramanian *et al.*, 2012], the performance of, amongst others, EDF and LLF is explored via simulation studies. It is concluded that EDF is optimal in the absence of power constraints, but the metrics used are required reserve energy and reserve capacity. Moreover, in [Chen and Tong, 2012], the performance of EDF and LLF are studied again. However, both scheduling algorithms are slightly modified to include a simple admission policy. This admission policy checks whether the system remains underloaded upon arrival of each vehicle. The authors also propose an online scheduling algorithm named TAGS that is a greedy scheduling strategy that performs simple threshold checks on profitability for its admission policy. Then, simulations are used to evaluate the performance of TAGS against EDF and unmanaged charging (UC), where UC is interchangeable with FCFS. LLF is not considered in the simulations. The performance metric used is the average (monetary) profit of the charging service provider, which is a different direction than the direction of this paper. Furthermore, in [Chen *et al.*, 2011] a more theoretical approach to optimality is taken and no simulation analysis is conducted. The authors propose an online scheduling algorithm referred to as DSC (Deadline Schedul-

[1]"EVs are defined as vehicles that are powered with an on-board battery that can be charged from an external source of electricity. This definition includes plug-in hybrid electric vehicles (PHEVs) and battery electric vehicles (BEVs). EVs often are referred to as plug-in electric vehicles (PEVs)" [Muratori *et al.*, 2021].

ing with Commitment) and prove that its competitive ratio is the maximum ratio achievable for online preemptive scheduling with commitment, therefore proving its optimality.

In more recent work, namely [Kong *et al.*, 2016], experimental analysis of multiple scheduling strategies is carried out. The performance is measured under varying scales of charging resources and varying charging load of vehicles. This approach is similar to that of this paper, where we consider different charging station capacities and different total number of vehicles arriving as well. However, different from our work, the performance metric considered is the social welfare, which the authors define as the sum utility of the EVs, where utility is a measurement of user satisfaction. The utility depends on both the price for electricity and the state-of-charge of an EV. Another difference is the fact that LLF is not considered in their simulations. They reason that due to the frequent preemption of the algorithm, it is impractical for charging EVs.

The remainder of the paper is structured as follows: First, Section 2 describes the model used and formulates the performance metrics. Then, Section 3 introduces the scheduling strategies that are evaluated and compared. Next, Section 4 describes the experimental setup and presents and discusses the results obtained from the experiments. Thereafter, Section 5 mentions the ethical aspects concerning this study but also research in general. It discusses the importance of reproducability as well. Finally, Section 6 concludes.

2 EV Scheduling at Charging Stations

In this section, we first present the model used for scheduling EVs at charging stations in Section 2.1. Then, in Section 2.2, we describe the performance metrics used to asses and compare the performance of the scheduling strategies.

2.1 Model

To model the problem of scheduling EVs at a charging station, we assume a combined zone, park-and-charge system [Huang *et al.*, 2012]. In such a system, as opposed to a separate zone system, EVs are parked and charged in the same area. Each parking spot has an associated charging point of which its power is regulated by a centralized controller. We will view this centralized controller as the charging station itself hereafter. Furthermore, a station will have a fixed capacity m which is the maximum number of charging points that can simultaneously supply power to charging EVs. In other words, in the context of a multiprocessor system, m would be the number of processors available.

Furthermore, each EV i, which can be viewed as a charging job, is modeled as a triplet EV_i where $EV_i = (a_i, s_i, d_i)$. Here, the first parameter a_i is the arrival time of EV i. This is the time the vehicle arrives at the charging station and therefore, it is the earliest time the EV can be charged. The next parameter, s_i, is the state of charge that EV i has upon arriving at the charging station. The last parameter, d_i, is the latest time at which the charging of EV i should be finished, also called the deadline. In section 4, we describe how we generate these stochastic parameters for the simulation. Furthermore, a charging station will only be able to view the parameters of an EV after it has arrived (i.e., after its arrival time has passed). In other words, the charging station will use an online scheduling algorithm to decide the order in which to charge the vehicles.

2.2 Performance Metrics

Our objective is to compare and evaluate multiple scheduling policies. In order to do this, we need to consider appropriate performance metrics. These metrics are listed below.

i The average waiting time

ii The maximum tardiness

iii The average tardiness

We define the waiting time as the total amount of time an EV spends at a charging station. This includes both the waiting time in the queue and the charging time. More formally, given the arrival time a_i and finish time f_i of an EV i, the waiting time is $W_i = f_i - a_i$. Furthermore, tardiness is defined as the amount of time an EV is finished charging later than its deadline. Tardiness (as opposed to lateness) is non-negative. More formally, given the finish time f_i and deadline d_i of an EV i, the tardiness of EV i is $T_i = max\{0, f_i - d_i\}$.

As a result, our definition of best performing depends on the specific metric considered: For metric i, we are interested in minimising the average waiting time. Therefore, the policy with the minimum average waiting time would be the best suited. For metric ii, we prefer to minimise the maximum tardiness. Consequently, the policy with the minimum maximum tardiness is the ideal policy. Lastly, for metric iii, we would like to see highest utilization possible. For that reason, we consider the maximum utilization as the best suited for this metric.

Lastly, the intuition behind these metrics is as follows. Through metric i, we can evaluate how well a scheduling strategy is suited for creating shorter queue times at a station. This metric does not consider the deadlines of the vehicles while evaluating performance. On the other hand, metrics ii and iii evaluate strategies taking into account the deadlines. Via the last two metrics, therefore, we are able to compare strategies based on how well they are able to adhere to deadlines.

3 Scheduling Strategies

This section first introduces the well-known scheduling strategies that are considered in this paper. Thereafter, it describes two new algorithms that are additionally considered. Lastly, it mentions for which algorithms both a preemptive and non-preemptive version are considered.

3.1 Well-Known Scheduling Algorithms

The first scheduling strategy we consider is a well-known strategy, namely earliest deadline first (EDF) [Liu and Layland, 1973]. EDF prioritises EVs with earlier deadlines over EVs with later deadlines. Secondly, we consider least laxity first (LLF) [Mok, 1983]. LLF prioritises EVs with the smallest amount of slack time (i.e., the time remaining until the deadline is reached after charging is completed). More formally, the laxity of EV i with deadline d_i and finish time f_i is $L_i = d_i - f_i$. Thirdly, we consider shortest job first (SJF).

This strategy prioritises EVs that have the shortest charging time. Furthermore, similar to [De Weerdt *et al.*, 2015], we use a first come first serve (FCFS) scheduling strategy as well. This strategy will also function as a benchmark.

3.2 EDSJF and LLSJF

In addition to the the scheduling algorithm mentioned in the previous section, we also study the performance of two new algorithms. These algorithms, named earliest deadline shortest job first (EDSJF) and least laxity shortest job first (LLSJF), are slight modifications of EDF and LLF, respectively. Here, as opposed to EDF and LLF, ties are not broken arbitrarily but by the EV with the shortest charging time.

We hypothesise that these algorithms will perform similar to their corresponding counterparts under the maximum and average tardiness metrics. However, we believe they will perform better when considering average waiting times, especially in congested scenarios. The reason behind this intuition is that, since EDSJF and LLSJF sort on deadlines and laxity at first, respectively, they should not perform worse than their counterparts when considering tardiness metrics. However, when considering the waiting time, they have an edge on their respective counterparts which should lead to performance improvements. Moreover, since the probability that a tie occurs increases as more vehicles arrive at the station, a congested setting should lead to a more apparent improvement.

3.3 Preemption

Another interesting aspect of scheduling strategies that we study is the impact of preemption. To this end, we implement both the preemptive and non-preemptive versions of two algorithms. To be exact, these algorithms are EDF and SJF. The preemptive versions of these algorithms are named EDF(pre) and SJF(pre) for EDF and SJF, respectively. On the other hand, we do not consider preemptive LLF for the same reason as in [Chen and Tong, 2012]. That being the impracticability of the algorithm due to its nature of frequently preempting. However we still find it interesting to implement the non-preemptive version for evaluation. Consequently, we are able to retrieve some insight in the performance of LLF while nevertheless considering a practical algorithm. Moreover, we note that EDSJF and LLSJF, described in Section 3.2, are preemptive as well.

As for our expectations, we anticipate the preemptive versions outperforming their corresponding counterparts, but only under their correlated metric. For SJF, that is the average waiting time and for the other strategies, that is both the maximum and average tardiness.

4 Experimental Setup and Results

In this section, we compare the scheduling strategies discussed in Section 3 using a simulation. We explain how this simulation is set up and we describe the parameters used in the simulation in Section 4.1. Furthermore, we describe the probability distributions used to initialise the arrival times and deadlines in Section 4.2. Lastly, we present and explain the results obtained from the simulation in Section 4.3.

Algorithm 1 Main Simulation Loop

```
for each scheduling algorithm do
    currIteration ← 0
    while currIteration < numIterations do
        reset the charging station cs
        initialise arrival times
        initialise deadlines
        initialise EVs
        currStep ← 0
        while all EVs are not charged yet do
            for each ev do
                if ev arrives at this step then
                    enqueue ev at the charging station
                end if
            end for
            cs.charge(currStep)
            currStep++
        end while
    end while
end for
```

4.1 Simulation Setup

We use a fixed-increment time progression simulation to compare the scheduling strategies. Algorithm 1 shows the main simulation loop. To explain, for every scheduling algorithm, the simulation runs the charging station scenario using that scheduling method for a total of $numIterations$ times. Since $numIterations$ is set to 4000, the simulation will run for 4000 iterations for every strategy. In every iteration, the charging station will be reset first (queues emptied). Thereafter, the arrival times, deadlines and EVs will be initialised and lastly, $currStep$ will be set to 0. Then, for every step of the simulation (which represents a time slice), we check which EVs arrive during that period of time and we enqueue an arriving vehicle at the charging station for charging. Since we already obtain all the required data for our results after all vehicles have been charged, we can stop the current iteration and proceed to the next one when we are know all EVs have been fully charged. Furthermore, in every step, we call the method $cs.charge(currStep)$ which will charge a number of vehicles equal to the charging capacity, and in the order determined by the current scheduling strategy.

To explain the simulation environment more precisely, but also for reproducibility purposes, we will list and explain all the parameter values used in the simulation below.

- $numIterations$: The number of iterations that every scheduling strategy is ran. This number is set to 4000 so we can take the average of a large number of iterations which will mitigate the effect of random errors made due to the stochastic nature of the simulation.
- $numEVs$: The number of EVs that are initialised every iteration and that will eventually arrive at the charging station to be charged. This number is set to 100.
- $stationCapacity$: The number of EVs the station can charge simultaneously, earlier named m in Section 2.1. This number is set to 3.

- $maxCharge$: The number of steps it takes to fully charge an EV from 0 to 100%. In other words, this corresponds to the charging duration. This parameter is set to 10.
- $minutesPerStep$: The number of minutes that one step in the simulation represents. This parameter is set represent 5 minutes.
- $simulationDuration$: The duration of one whole iteration (in hours). This number is set to 24, representing a full day. Given the value of this parameter, we can compute the number of total steps the simulation will run per iteration as follows:

$$numSteps = \frac{simulationDuration}{minutesPerStep} \times 60$$

- $arrivalsDistribution$: The probability distribution used to generate the arrival times of the EVs. Further explained in Section 4.2.
- $deadlinesDistribution$: The probability distribution used to generate the deadlines of the EVs. Further explained in Section 4.2.

4.2 Arrival Time and Deadline Probability Distributions

To initialise the EVs, we need to generate the arrival times and deadlines first. For this, we use different probability distributions under which we can compare the performance of the scheduling strategies. These probability distributions are set using the $arrivalsDistribution$ and $deadlinesDistribution$ parameters. We use three different distributions for the arrival times and one distribution for the deadlines. For the arrival times, we use a uniform distribution in the range $[1, numSteps]$, where $numSteps$ corresponds to the last time slice of the simulation (see section 4.1). This is similar to the method the authors in [Kong *et al.*, 2016] use to generate arrival times. Secondly, we use a Poisson process with a mean number of arrivals per day of 1000. This is the same method [Huang *et al.*, 2012] use to generate the arrival times of their EVs. Lastly, we use a normal distribution with mean at 9am and a standard deviation of 1 to simulate an office scenario where employees arrive roughly at 9am to start their work shift. Considering these distributions, we remark that they result in different degrees of congestion. Sorted on the least to the most amount of congestion, we have the uniform distribution, followed by the Poisson distribution, followed by the normal distribution.

For the deadlines of the vehicles, we use a uniform distribution for each EV i in the range $[a_i, numSteps]$, where a_i is the arrival time of EV i. Therefore, we make it impossible for EVs to be initialised with a deadline earlier than their arrival time. This is because we are interested in the tardiness that the scheduling strategy of the charging station produces, and initialising EVs that already have a tardiness upon arrival works against this.

4.3 Results

We begin with discussing the performance of the scheduling algorithms under different probability distributions (Figures 1 - 5). Thereafter, we will proceed to discuss the performance of the algorithms under varying parameters (Figures 6 - 14). Before proceeding to the results discussion, we would like to clarify the following terms used in the upcoming paragraphs: When using the term "EDF and its versions" or "EDF versions" we refer to the strategies: EDF, EDF(pre) and EDSJF. When using the term "tardiness algorithms" we refer to EDF, EDF(pre), EDSJF, LLF and LLSJF. The intuition behind this last term is the fact that those algorithms all primarily take the deadline into consideration when deciding upon prioritisation. The deadline is used to calculate tardiness and therefore these algorithms are all connected to tardiness, hence the term.

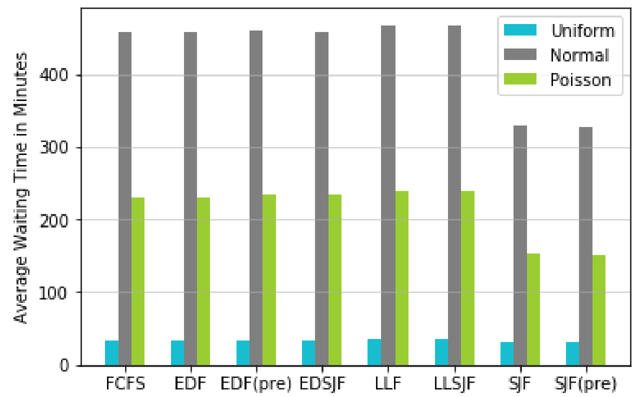

Figure 1: Average waiting times of the scheduling strategies under three arrival probability distributions (where pre means preemptive).

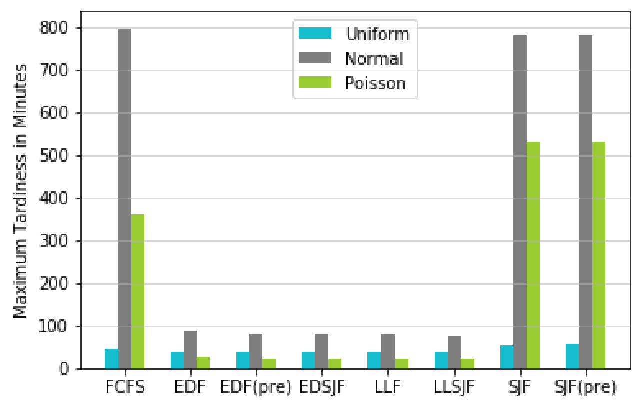

Figure 2: Maximum tardiness of the scheduling strategies under three arrival probability distributions (where pre means preemptive).

1) Performance evaluation under different probability distributions: **Average waiting time.** Figure 1 shows the average waiting times of scheduling methods. From this figure, we first observe that, with the exception of SJF (both the preemptive and non-preemptive version), all the strategies result in roughly the same amount of average waiting time under all probability distributions. SJF, on the other hand, has a significant shorter (34% under Poisson) average waiting time under the normal and Poisson distributions, but under the uni-

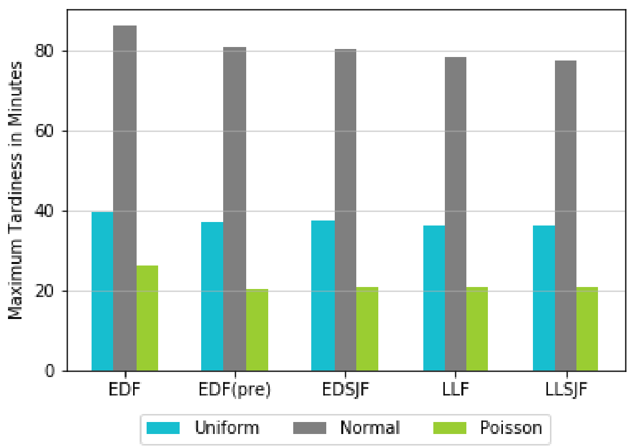

Figure 3: Maximum tardiness of the tardiness strategies under three arrival probability distributions (where pre means preemptive).

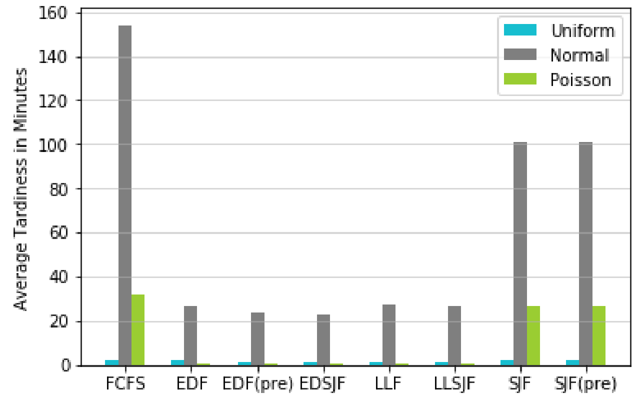

Figure 4: Average tardiness of the scheduling strategies under three arrival probability distributions (where pre means preemptive).

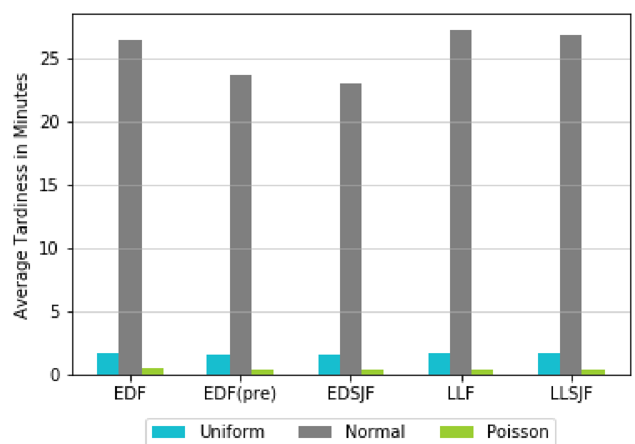

Figure 5: Average tardiness of the tardiness strategies under three arrival probability distributions (where pre means preemptive).

form distribution, the performance is equal to the others. The fact that SJF performs better than its counterparts under the normal and Poisson distributions is to be expected. This is due to the fact that the other strategies either focus solely on the deadline of the EVs (i.e., the tardiness algorithms) or focus on no particular criterion at all (i.e., FCFS), whereas SJF prioritises using charging times, a criterion that is linked to the waiting time metric. As for the uniform distribution case, SJF's performance is similar to the others because at no point during the day will there arrive a lot of EVs at the same time. Therefore, every EV arriving can begin charging immediately which leads to shorter waiting times. Next, we observe that there is no significant (only 2 minutes) difference between the performance of preemptive EDF and non-preemptive EDF, under all three distributions. The same holds for SJF and SJF(pre). The fact that there is no difference between EDF and EDF(pre) is to be expected because EDF prioritises exclusively on deadlines, so the its preemptive version would not improve when considering average waiting times. However, the results of SJF are interesting because we hypothesised that SJF(pre) would outperform SJF in regards to the average waiting time. Lastly, we observe that there is no performance difference between EDF and EDSJF, and between LLF and LLSJF. This is unexpected as well since we hypothesised that EDSJF and LLSJF would outperform their corresponding counterparts.

Maximum tardiness. The next figure, Figure 2, displays the maximum tardiness over all the EVs charging at the charging station. We observe that, in contrast to Figure 1, SJF now performs significantly worse that the other strategies, resulting in a 9 times increased tardiness compared to EDF for instance. This follows naturally from the fact that SJF does not consider deadlines when scheduling. Moreover, the tardiness algorithms all perform the best here, as expected. FCFS performs similar to the tardiness strategies when the arrival times are generated uniformly, but when they are generated using a normal or Poison distribution, the performance cannot match theirs anymore. The same holds for SJF. This is presumably because of the same reason mentioned earlier, namely that EVs can charge immediately upon arrival under a uniform distribution. Consequently, EVs are able to finish charging before their deadlines more often which leads to less tardiness. On the other hand, when we consider the normal and Poisson distributions for the arrival times, more congestion will take place at the charging station because of the colliding arrival times. This in turn will make it necessary for more EVs to wait in the queue, increasing their tardiness.

In order to observe the performance differences more clearly in regards to the maximum tardiness, Figure 3 shows only the tardiness strategies. We discover the following facts: First, the preemptive version of EDF performs slightly better than the non-preemptive version under all three distributions. To be precise, under the normal distribution where the difference is the most pronounced, the preemptive version has 5 minutes less maximum tardiness, which is roughly a 6% improvement. This is in line with our hypothesis. Second, EDSJF performs similar to EDF(pre). This is due to the fact that EDSJF is preemptive as well and we are not considering waiting times in this figure, therefore sorting on the shortest job aspect of EDSJF can not be evaluated. Third, LLF and

LLSJF have no significant performance difference. However, both perform slightly better than EDF and its two versions.

Average tardiness. In respect to the average tardiness, Figure 4 demonstrates that FCFS has 52% increased average tardiness than SJF and SJF(pre) which both in turn have roughly a 4 times increased average tardiness than the tardiness algorithms, under a normal distribution. The same relations in performance hold under the Poisson distribution. Only under the uniform distribution, the performance differences are not as pronounced, where the congestion at the charging station is limited. From these facts, we can conclude that FCFS, SJF and SJF(pre) perform significantly worse than the alternative algorithms when considering average tardiness in a congested setting.

Once more, to observe the differences more clearly, only the lateness strategies are presented in Figure 5. Here, we observe the following facts under the normal distribution: First, preemptive EDF performs 10% better than its non-preemptive version which is to be expected. Second, there is no significant performance difference between preemptive EDF and EDSJF and the same is true for LLF and LLSJF which is expected as well for the same reason as mentioned wile considering the maximum tardiness. Third, LLF and LLSJF perform similar to EDF but worse than EDF(pre) and EDSJF. This difference, however, can be explained by the fact that EDF, LLF and LLSJF are all non-preemptive whereas EDF(pre) and EDSJF are preemptive. This is nevertheless an interesting observation because it contrasts the findings under the maximum tardiness metric from Figure 3. There, LLF and LLSJF perform better than their counterparts, even the preemptive ones. This finding suggests that EDF is a better option when considering the average tardiness whereas LLF is better when we consider the maximum tardiness.

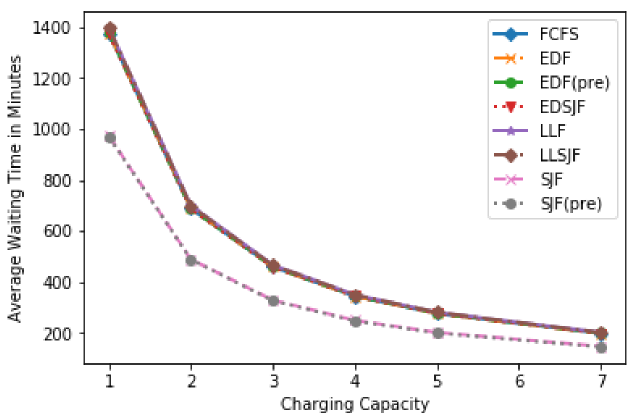

Figure 6: Average waiting times of the scheduling strategies under varying charging station capacities (where pre means preemptive).

2) Performance evaluation under varying parameter values: For the remainder of the results in this section, a normal distribution is used to generate arrival times. **Impact of charging capacity.** Moving on to the next set of figures, we start with Figures 6, 7 and 8. These figures show the performance of the algorithms under varying charging station capacities. In all three figures, the performances increase as the capacity increases. This is to be expected since an increased capacity leads to decreased queue sizes. In regards to average waiting times, Figure 6 demonstrates that SJF and SJF(pre) always perform better than their counterparts, regardless of the capacity. In respect to the maximum tardiness, Figure 7 shows that FCFS, SJF and SJF(pre) improve their performance slowly as capacity increases, whereas the lateness algorithms improve more rapidly. Lastly, considering the average tardiness, we observe that FCFS performs the worst under all capacities. Furthermore, a line crossover presents itself around the charging capacity mark of 2. Before this crossover, the tardiness algorithms perform significantly worse than SJF and SJF(pre). This performance relation switches after the crossover. From that point on, the lateness strategies keep performing the best, as expected. This is an interesting observation. One possible explanation is the following: when the capacity decreases to 1, the load on the charging station increases significantly. Presumably, under such high loads, prioritising EVs with the shortest charging times is more beneficial than sorting based on their deadlines.

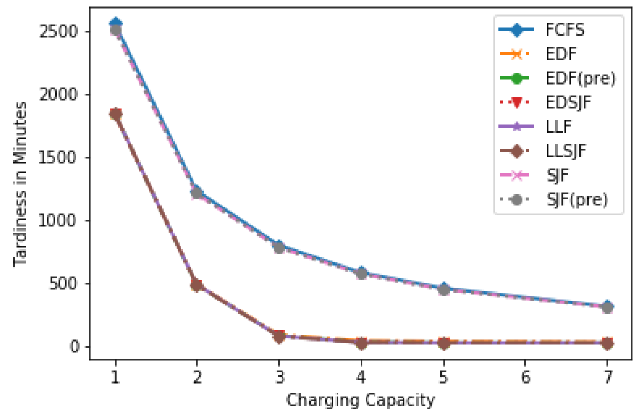

Figure 7: Maximum tardiness of the scheduling strategies under varying charging station capacities (where pre means preemptive).

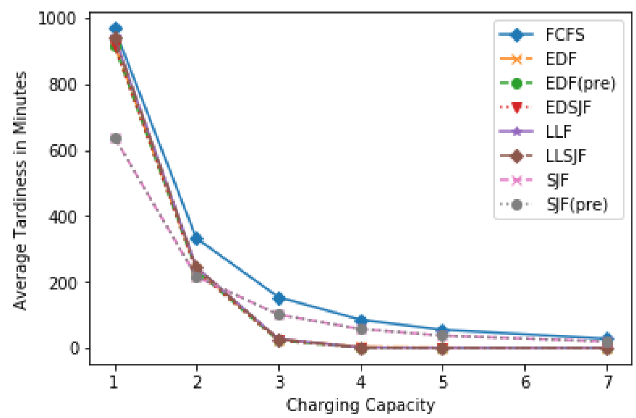

Figure 8: Average tardiness of the scheduling strategies under varying charging station capacities (where pre means preemptive).

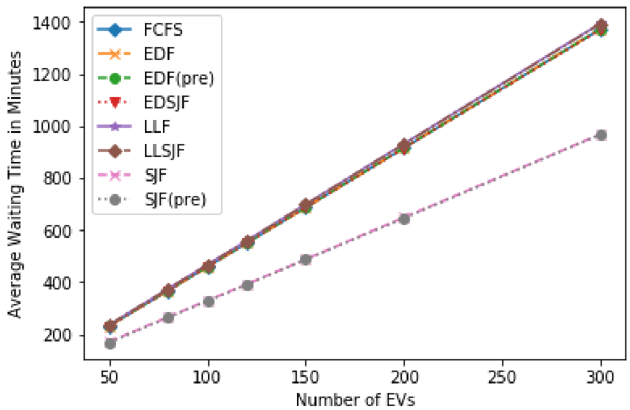

Figure 9: Average waiting times of the scheduling strategies under varying numbers of EVs arriving (where pre means preemptive).

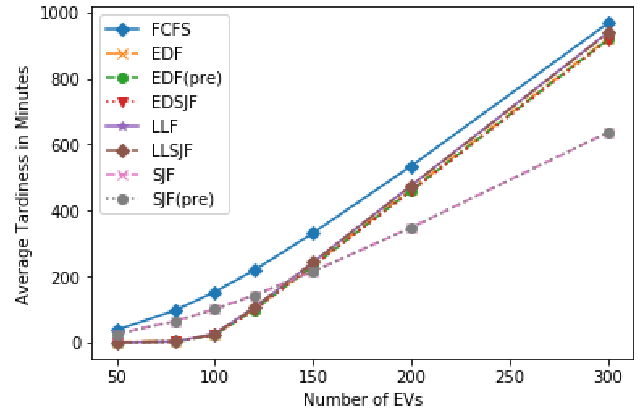

Figure 11: Average tardiness of the scheduling strategies under varying numbers of EVs arriving (where pre means preemptive).

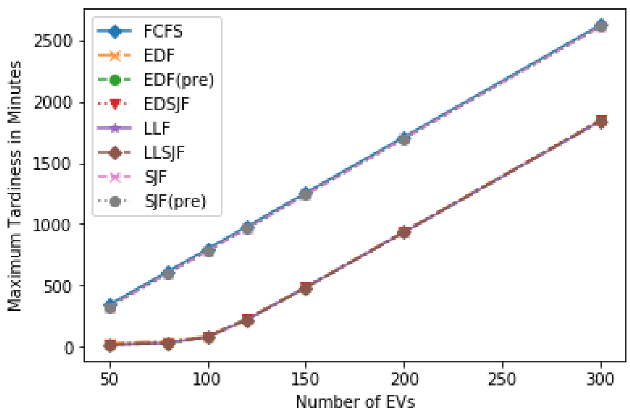

Figure 10: Maximum tardiness of the scheduling strategies under varying numbers of EVs arriving (where pre means preemptive).

We will come back to this thought again when studying the number of EVs and charging duration results. Another observation, one that holds for all three metrics, is that there exists no performance difference between SJF and its preemptive version SJF(pre). For EDF and preemptive EDF case, there do exist some small performance differences when considering the maximum and average tardiness, as mentioned earlier. However, these figures now show that the differences do not become more apparent as the charging capacity varies.

Impact of vehicle number. Figures 9, 10 and 11 display how the scheduling algorithms perform as the number of EVs arriving increases. We observe the following facts: First, all metrics considered, SJF and SJF(pre) perform the same constantly, as seen before. Furthermore, the performance of all strategies decreases as the number of EVs arriving increases. This is because there are more EVs arriving to be charged, which leads to a longer queue at the charging station which in turn leads to more waiting and more EVs missing their deadlines. Second, regarding waiting times, as more vehicles arrive, SJF and SJF(pre) increasingly perform better than the tardiness algorithms. Moreover, as the number of vehicles arriving increases, the performance of LLF and LLSJF slightly diverges from the performance of the EDF versions and becomes worse. This is an interesting observation. We attempt to explain this as follows: On average, a shorter charging duration results in an earlier finish time. Consequently, considering the equation for laxity $L = d - f$ (described in Section 3), when the finish time decreases, the slack time increases resulting in less priority for LLF and LLSJF whereas the EV would be prioritised by SJF. Thus, laxity prioritisation works against SJF prioritisation. Since EDF does not have this issue, the performances diverges when considering average waiting times. Third, in regards to the maximum tardiness, all scheduling strategies hold the same relation to one another as more vehicles arrive. More precisely, SJF, SJF(pre) and FCFS persistently perform worse than the tardiness algorithms. Fourth, in view of the average tardiness, FCFS continuously performs worse than all its counterparts. In addition, at around 140 EVs arriving, the tardiness algorithm lines and the SJF and SJF(pre) lines cross over. From that point on, the tardiness algorithms perform increasingly worse and approach the same performance as FCFS. This observation is similar to the one made when discussing varying charging capacities under the average tardiness. Once more, we presume that as the load on the station significantly increases as the number of EVs increase, allowing the shortest charging times first becomes the best order.

Impact of charging duration. Figures 12, 13 and 14 demonstrate the performance of the algorithms under varying charging durations. First, we observe that all algorithms perform worse as the charging duration increases. This is to be expected because a longer charging duration naturally results in longer charging times. If EVs need more time to charge, other EVs in the queue have to wait longer on their turn. This leads to longer waiting times and to more EVs missing their deadlines. Second, considering the waiting times, Figure 12 shows that SJF and SJF(pre) perform best at all charging durations and the performance gap increases as the charging duration increases. Since SJF and SJF(pre), in contrast to their counterparts, prioritise using the charging time, this is to be expected. However, between these two algorithms, no signif-

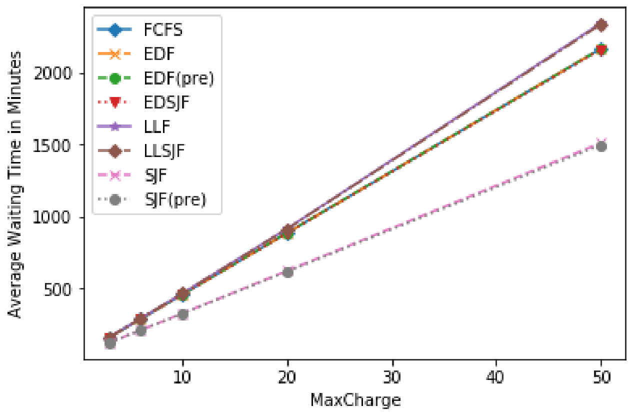

Figure 12: Average waiting times of the scheduling strategies under varying charging durations (where pre means preemptive).

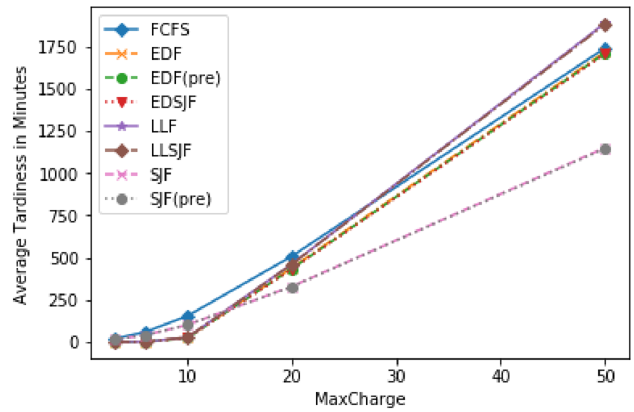

Figure 14: Average tardiness of the scheduling strategies under varying charging durations (where pre means preemptive).

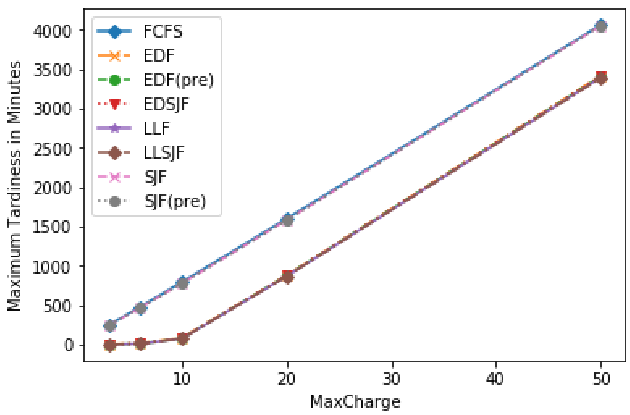

Figure 13: Maximum tardiness of the scheduling strategies under varying charging durations (where pre means preemptive).

icant difference can be observed. This is, again, an interesting observation since we would expect SJF(pre) to outperform its counterpart. Furthermore, non-preemptive EDF does not perform significantly different than preemptive EDF or the preemptive EDSJF. This, on the other hand, is not an unexpected result, since EDF and its versions all prioritise using the deadlines of the EVs which form no connection to the waiting times. Moreover, as the charging duration increases, the performance of LLF and LLSJF decreases more rapidly than that of the EDF versions. The reason for this is the same as discussed before. As to why this performance gap increases, we hypothesize that due to a longer charging duration, charging takes longer and therefore, the effects of a single non-optimal schedule are amplified. Next, concerning the maximum tardiness, Figure 13 shows that SJF and SJF(pre) perform similar to FCFS. This is not unexpected since the aforementioned strategies all do not consider deadlines in their scheduling policy. On the other hand, all the strategies that do incorporate deadlines into their policy, have a similar performance profile. These are the tardiness algorithms. Lastly, in regards to the average tardiness, Figure 14 makes way for some interesting observations. Once more, we observe SJF outperforming the lateness algorithms after a certain point, here 14, as the load on the station increases. This figure is no exception to the rule since an increased charging duration leads to more congestion at the station. Moreover, we observe that LLF and LLSJF perform increasingly worse than the EDF versions as the charging duration increases. This finding might suggest that under high loads, prioritising on deadlines directly is better than prioritising on slack time.

5 Responsible Research

In this section, we mention some of the aspects concerning ethicality and reprodicibilty of research. Thereafter, we discuss the ethicality and reproducibilty of our own research.

5.1 Ethical Aspects

When doing research, one should carefully consider the ethical aspects of the research. It is of paramount importance that the integrity of the research is up to par and that the research is done in a responsible way.

Related to our study, the most notable ethical aspect to consider is the realistic traffic data that is collected in the original paper ([De Weerdt *et al.*, 2015]) which this paper builds upon. In that paper, the authors use origin-destination pairs with the corresponding departure times in their experimental evaluation. They collect this data from a Dutch National Survey [Centraal Bureau Voor De Statistiek (CBS) / Rijkswaterstaat (RWS), 2012]. The collection of data could bring up privacy concerns and therefore, when using this data, one should consider if no privacy rights are violated. One way to check if this is indeed the case is by making sure the used data is anonymised.

In this study specifically however, we do not use any historical real life data. This is discussed further as a possibility for future work in section 6. Moreover, this study does not collect any new data which might present any privacy concerns. As a result, we can safely assume that no privacy rights are violated.

5.2 Reproducibility

Another important aspect of research is reproducibility. The results presented in a study or paper should be relatively easy to obtain by others. To achieve this, great care should be put into describing the steps and tasks necessary to perform in order to achieve the results. Since results are often obtained experimentally via simulations in the field of Computer Science, researchers should make sure all the details of their simulation environment (e.g., simulation parameters, computation resources available during the simulation, and the simulation (pseudo)code) are provided and explained where possible.

To this end, this paper includes the link to the repository[2] where the code of the simulation used in this paper is stored. Furthermore, it gives a pseudocode of the main simulation loop and explains the code as well. Lastly, it describes all the parameters used and provides all the values used for these parameters. With all this information, we are quite sure a reader could produce nearly identical results as those presented in this paper (section 4.3).

6 Conclusions and Future Work

This paper studies varies scheduling strategies for the scheduling of EVs at a single charging station. These strategies are evaluated under three performance metrics in order to determine their effectiveness. Through extensive experimental analysis, we find that EDF is the best alternative when one values adherence to deadlines. Applying this strategy results in 83% lower average tardiness in comparison to FCFS in scenarios where the charging station is congested (i.e., under the normal distribution). The preemptive version of this strategy performs even better, resulting in a 9% lower average tardiness than its non-preemptive counterpart. On the other hand, when one intends to minimise average waiting times, we find that SJF is the best option. Compared to the other strategies studied, SJF achieves 29% shorter average waiting times at the charging station when the normal distribution is considered. Interestingly, we found no performance improvement for using preemptive SJF over the non-preemptive version. Furthermore, contrarily to our expectations, we found no differences in performance between EDF(pre) and EDSJF and between LLF and LLSJF.

There are several directions for future study. Firstly, the strategies could be extended with different admission policies. In that case, the charging station would have the option to reject EVs for charging. The advantage of such an extension is twofold. On one hand, EV owners are now guaranteed that their vehicle will be charged before its deadline. On the other hand, the issue of deadline fraud can be combatted where EV owner can consciously present misinformation about their deadline needs to have prioritisation over other owners. Secondly, a monetary dimension can be added to this study by extending the EV model with a willingness to pay attribute. Consequently, we could examine the effects of a pricing policy on the congestion and performance of the strategies. Lastly, for the generation of the arrival times, more realistic data can be used. For instance, using historical arrival times at a certain real-life charging station can possibly lead to more accurate results.

[2] https://gitlab.ewi.tudelft.nl/cse3000/2020-2021/rp-group-16/rp-group-16-aamouzandeh

References

[Centraal Bureau Voor De Statistiek (CBS) / Rijkswaterstaat (RWS), 2012] Centraal Bureau Voor De Statistiek (CBS) / Rijkswaterstaat (RWS). Onderzoek verplaatsingen in nederland 2011 - ovin 2011 versie 1.0, 2012.

[Chen and Tong, 2012] Shiyao Chen and Lang Tong. iems for large scale charging of electric vehicles: Architecture and optimal online scheduling. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 629–634. IEEE, 2012.

[Chen *et al.*, 2011] Shiyao Chen, Lang Tong, and Ting He. Optimal deadline scheduling with commitment. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 111–118. IEEE, 2011.

[De Weerdt *et al.*, 2015] Mathijs M De Weerdt, Sebastian Stein, Enrico H Gerding, Valentin Robu, and Nicholas R Jennings. Intention-aware routing of electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1472–1482, 2015.

[Huang *et al.*, 2012] Jing Huang, Vijay Gupta, and Yih-Fang Huang. Scheduling algorithms for phev charging in shared parking lots. In *2012 American Control Conference (ACC)*, pages 276–281. IEEE, 2012.

[Kong *et al.*, 2016] Fanxin Kong, Qiao Xiang, Linghe Kong, and Xue Liu. On-line event-driven scheduling for electric vehicle charging via park-and-charge. In *2016 IEEE Real-Time Systems Symposium (RTSS)*, pages 69–78. IEEE, 2016.

[Liu and Layland, 1973] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[Mok, 1983] Aloysius Ka-Lau Mok. *Fundamental design problems of distributed systems for the hard-real-time environment*. PhD thesis, Massachusetts Institute of Technology, 1983.

[Muratori *et al.*, 2021] Matteo Muratori, Marcus Alexander, Doug Arent, Morgan Bazilian, Ercan M Dede, John Farrell, Chis Gearhart, David Greene, Alan Jenn, Matthew Keyser, et al. The rise of electric vehicles–2020 status and future expectations. *Progress in Energy*, 3(2):022002, 2021.

[Qin and Zhang, 2011] Hua Qin and Wensheng Zhang. Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. In *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*, pages 51–60, 2011.

[Subramanian *et al.*, 2012] Anand Subramanian, Manuel Garcia, Alejandro Dominguez-Garcia, Duncan Callaway, Kameshwar Poolla, and Pravin Varaiya. Real-time scheduling of deferrable electric loads. In *2012 American Control Conference (ACC)*, pages 3643–3650. IEEE, 2012.