

Scheduling Electric-Vehicle Charging Under Grid Capacity and Minimum Charge Rates

Bruna Cavalcanti Lauro 

Department of Advanced Computing Sciences, Maastricht University, The Netherlands

Matúš Mihalák 

Department of Advanced Computing Sciences, Maastricht University, The Netherlands

Filip Schlembach 

Department of Advanced Computing Sciences, Maastricht University, The Netherlands

Evgueni Smirnov 

Department of Advanced Computing Sciences, Maastricht University, The Netherlands

Abstract

Electric vehicle adoption is growing rapidly, and parking facilities face increasing pressure to expand their charging infrastructure. This paper studies the power allocation scheduling problem that arises when a parking facility with fixed grid-capacity limit must distribute power to connected electric vehicles, under given lower and upper bounds on the charge rate at any given time. We focus on the optimization goal of maximizing the number of vehicles that get charged a given target of power, with the secondary goal of providing as much power as possible. We show that the problem is NP-hard. We formulate the problem as a mixed-integer linear program (MILP) that establishes a theoretical upper bound on performance of any online algorithm, which is the main goal of the real-world problem. We design and compare a few online scheduling strategies against the offline bound, using a real-world instance of vehicles and testing across 6 to 40 chargers. We use MILP as an online strategy, classical earliest-deadline first greedy algorithm, and we also design data-driven approaches using machine learning techniques. Results show that the simplest static load balancing approach cannot compete with the best performing online approach, and that the real-world problem can benefit from dynamic load balancing strategies such as those designed and evaluated in this paper.

2012 ACM Subject Classification Applied computing → Operations research; Computing methodologies → Machine learning

Keywords and phrases Electric-Vehicle Charging, Scheduling, Grid-Capacity Constraints, Charge-Rate Lower-Bounds

Digital Object Identifier 10.4230/OASICS.ATMOS.2026.9

Acknowledgements We thank the anonymous reviewers for very thorough and inspiring comments, which helped to improve the paper.

1 Introduction

The adoption of electric vehicles (EVs) is growing, bringing new challenges and demands on the existing automotive infrastructure, which was tailored and optimized to be used primarily by combustion-engine vehicles. In particular, the relatively long charging times and the limited availability of electrical power provides several scheduling and coordination problems to be solved.



© Bruna Cavalcanti Lauro, Matúš Mihalák, Filip Schlembach, and Evgueni Smirnov; licensed under Creative Commons License CC-BY 4.0

26th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2026).

Editors: Valentina Cacchiani and Stefan Funke; Article No. 9; pp. 9:1–9:17



OpenAccess Series in Informatics

OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One such problem arises in parking facilities, where customers demand more parking spots with charging possibilities of electric vehicles. Yet, most of parking facilities operate under a fixed grid capacity limit that caps the total power available at any moment.¹ As a result, the straightforward charging approach of “split the total available power equally among the connected EVs to the charging facility” is no longer feasible, since the IEC-61851 standard [9] mandates a minimum power threshold of 4.1 kW for full-battery EVs (BEVs) and 1.4 kW for plug-in hybrid EVs (PHEVs): as more vehicles connect simultaneously, the equal share per vehicle decreases; if the equal share falls below a vehicle’s threshold, its internal battery management system stops charging and transitions to an idle state. Consequently, beyond a certain number of concurrent vehicles, the entire facility stalls and delivers zero power until enough vehicles depart. Moreover, this straightforward approach, also called the *static load balancing* (SLB), does not take into account information such as the status of the battery charge, which could be used to further optimize, for example, the total amount of delivered energy or the number of cars charged close to full battery capacity.

In this paper, motivated by our real-world contacts with a parking infrastructure operator, we study the electric-vehicle charging scheduling problem with a power (grid-capacity) constraint and a minimum and maximum charging threshold for vehicles. Our research is guided by the following real-world scenario: a parking garage is currently equipped with 6 parking spots with charging facilities; at any moment, the facility can charge the fleet with a total of at most 50 kW of electric power; the facility uses static load balancing. The main question is whether dynamic load balancing, i.e., a centrally planned scheduling that continuously allocates available power across connected EVs in a, possibly, non-uniform way, can allow to extend the number of parking spots while maintaining a satisfactory level of service. Motivated by these real-world optimization concerns, we study the resulting offline and online scheduling problem defined in the following.

1.1 Terminology, Notation, and Problem Definition

A set of n electric vehicles $[n] := \{1, 2, \dots, n\}$ arrives at the parking facility over a fixed scheduling horizon $[0, T]$ partitioned into one-minute intervals $t \in \{1, 2, \dots, T\} =: [T]$. Each vehicle $i \in [n]$ is characterized by its arrival time $t_{\text{arr},i}$, its departure time $t_{\text{dep},i}$, its total requested energy $E_{\text{req},i}$ (in kW hours), its maximum power limit $P_{\text{max},i}$ (in kW), and its minimum power limit $P_{\text{min},i}$ (in kW). The facility can provide at any time a total power of at most P_{grid} (in kW). All of this information is an input to our problem, including the departure times (which the operator of the parking facility asks from the drivers at the entrance to the parking facility).

A *schedule* is an assignment of power to vehicles over time, given by values $S := \{p_{i,t}\}_{i \in [n]; t \in [T]}$, with the meaning that vehicle i is given power $p_{i,t}$ during time-interval t . A *feasible* schedule is a schedule $S = \{p_{i,t}\}_{i \in [n]; t \in [T]}$ such that (i) for any time interval $t \in T$, the total amount of energy assigned to all vehicles at time t is at most P_{grid} , (ii) no vehicle $i \in [n]$ is assigned a positive power at time $t \notin [t_{\text{arr},i}, t_{\text{dep},i}]$, (iii) every vehicle i is either assigned zero power, or power that is at least its lower-bound $P_{\text{min},i}$ and at most its upper-bound $P_{\text{max},i}$, and (iv) no vehicle is assigned, in total, more than the requested amount of power $E_{\text{req},i}$.

¹ Typical grid capacity of an average parking garage reported by our industrial partner is between 50kW and 150kW.

The goal is to find a feasible schedule that maximizes the following hierarchical objective: first, it maximizes the number of vehicles that get charged at least α -fraction of their requested energy, and second, the total energy delivered to the vehicles. The primary optimization goal is thus the customer satisfaction (number of satisfied customers), and the secondary optimization goal is both the customer satisfaction and the company profit (the total amount of delivered/sold energy). We adopt the jargon of the industry, and call the goal of charging the α -fraction of the requested energy the *service level agreement* target, or, for short, the SLA target. Throughout the paper, we maintain $\alpha = 0.9$ (or, 90%). We chose the value of 90%, because it is the largest value of α for which the charging of a lithium-ion battery is approximately linear in delivered energy [12]. None of the algorithmic approaches presented in this paper depends on the concrete choice of the value of α . The modeling assumes that the charging is approximately linear in delivered energy, and for that, any value of $\alpha \leq 0.9$ is fine.²

We call the interval $[t_{\text{arr},i}, t_{\text{dep},i}]$ the *connection window* of vehicle $i \in [n]$. For a time $t \in [T]$ within the connection window of vehicle $i \in [n]$, i.e., for $t_{\text{arr},i} \leq t \leq t_{\text{dep},i}$, vehicle i is *active* at time t with respect to a feasible schedule S if $p_{i,t} > 0$, and otherwise it is *inactive*.

1.2 Our Contributions

We first consider the offline problem, when the entire input is given to the algorithm at the beginning. We show that the problem is NP-hard, even when every vehicle has the same arrival time equal to zero, and every vehicle has the same (integer) departure time t_{dep} of value $\Theta(n)$ (i.e., linear in n). To address the real-world problem, we formulate the offline scheduling problem as a mixed integer linear program (MILP) and use it to compute optimum schedules to obtain a reference point for comparison to online approaches, and for comparison with the static load balancing approach.

We then consider the online problem, in which the input information about the vehicles is given to the algorithm upon the arrival times of the vehicles, i.e., at any time $t \in [T]$, the algorithm knows the information about all vehicles $i \in [n]$ with arrival time $t_{\text{arr},i} \leq t$. For the online problem, we design and experimentally evaluate several online scheduling algorithms that need to make decisions at every time point $t = 1, 2, \dots, T$ using only the information about vehicles whose arrival time is at or before t . The motivation for knowing the (future) departure time comes from the operation of the parking facility which asks every vehicle, upon arrival, for an (estimated) departure time and also for the requested energy power.

Some of our online strategies use data-driven approaches, in which the online algorithms can use, beyond the current state of the facility, also historical data. We use machine learning techniques to predict future arrival of electric vehicles, their requested power, and their departure times, and evaluate an online algorithm that at any moment computes an offline schedule on the current vehicles plus the predicted vehicles for a short future time window. The motivation for this approach is to find online strategies that possibly approach the offline optimum under realistic operational conditions.

² Albeit, much of the literature takes a more conservative approach and sets the threshold of $\alpha = 0.8$ for approximately-linear charging behavior.

1.3 Related Work

There are several related concepts in the field of machine scheduling. In machine scheduling, in general, there are n jobs having processing times, and possibly release dates, due dates, or weights, and there are m machines that can process the jobs, while, at any time, working on at most one job. A preemptive schedule is a schedule that allows a machine to process a job over several time intervals, whose total length is equal to the job's processing time.

Chen et al. [3] studies an online deadline scheduling problem with an on-arrival commitment requirement on one machine, where a scheduler must immediately accept or reject a job upon arrival. Their optimization goal is to maximize the total load of scheduled processing minus the remaining processing time of uncompleted jobs. Beside the differences in the optimization goal and the number of machines, observe also that in our problem, we cannot accept or reject any vehicles (since vehicles find autonomously any free parking spot with a charging facility). Furthermore, our problem requires allocating a shared continuous power capacity across a simultaneous fleet of connected vehicles at every minute, rather than managing a sequential single-job processor queue.

Alinia et al. [1] study an online scheduling problem for deadline-constrained EV charging, maximizing the total weighted delivered energy. Their problem permits power allocations up to a vehicle-specific maximum and having a facility grid capacity, which mirrors our work. However, their optimization goal is different, and they do not enforce a minimum power threshold as we do.

The PhD thesis of Zaidi [17] provides a classification of *electric-vehicle charging-scheduling problem* with respect to various optimization objectives (from the power-grid operators as well as from the charging-facility point of view), various constraints and considerations related to vehicles (e.g., arrival and departure times, charging limits) or infrastructure (electricity prices or charging capacity), and various solution approaches (exact offline algorithms, heuristics, or online algorithms). Besides the overview, the PhD thesis also provides NP-hardness of several fundamental variants of EV charging scheduling problems: (i) computing a minimum capacity $P_{\max}$ of the grid to satisfy all demands of given electric vehicles; (ii) maximizing the number of fully-charged vehicles where every parking slot can provide a uniform power rate or zero power at any time interval, by seeing the problem as a problem of minimizing the number of late jobs in a setting with parallel machines with release dates and due dates (which is known to be strongly NP-hard due to result of Du and Leung [4]).

Uiterkamp et al. [14] study an EV-charging problem with lower-bound constraints on the provided power (just like our paper does), but their objective is to flatten the combined base-load (given) and charging-load of EVs (to be computed) of the grid.

There are other scheduling papers related to charging of EVs, differing substantially in the constraints or in the objective, see e.g. [8, 16, 6] and references therein for a small subset of such papers.

Existing literature highlights a distinct gap: to the best of our knowledge, no prior work simultaneously incorporates a minimum operating power threshold, passive admission constraints, and a customer-centric hierarchical satisfaction objective.

1.4 Empirical Data and Demand Reconstruction

The experimental evaluation is driven by data collected in a concrete real-world parking facility during Week 51, 2024, one of the highest-demand weeks of the year. Two data sources were combined: (i) gate-access logs recording entry time-stamps, exit time-stamps, and vehicle types, and (ii) charging-session logs recording active charging durations and total energy consumed.

Because historical logs only capture vehicles that successfully accessed a charger, unserved demand had to be reconstructed. Vehicles that arrived when all chargers were occupied were identified by mapping gate arrivals against a minute-by-minute occupancy timeline. Their energy requirements were then generated using Monte Carlo sampling [13] from historical vehicle battery capacities and charging profiles. This reconstruction was developed and validated in previous work [10]. We present it here for completeness as it forms the basis of the dataset used for training and evaluating the scheduling strategies in this paper.

2 The Mixed-Integer Linear Program

The offline problem is given by n vehicles and their arrival times, departure times, requested energy, the lower and upper bounds on admissible power, and by the maximum power capacity of the facility.

To model the scheduling problem as a mixed-integer linear program (MILP), we introduce the variables that define the schedule, i.e., variables $p_{i,t} \in \mathbb{R}_{\geq 0}$, for every $i \in [n]$ and every $t \in [T]$, which express the power (in kW/min) allocated to vehicle i at minute t .

We further define binary indicator variables $s_{i,t} \in \{0, 1\}$, for every $i \in [n]$ and every $t \in [T]$, indicating whether vehicle i is actively charging ($s_{i,t} = 1$) or idle ($s_{i,t} = 0$) at time point t .

To track the number of vehicles that successfully charged 90% of their energy demand, we introduce a binary indicator variable $y_i \in \{0, 1\}$, for every $i \in [n]$, indicating whether vehicle i charged 90% of its requested energy.

Naturally, when solving the resulting MILP, we only instantiate decision variables for the intervals a vehicle is physically present. We define $T_i = \{t_{\text{arr},i}, \dots, t_{\text{dep},i}\}$ as the connection window of vehicle i , and conversely, $N_t = \{i \in [n] \mid t \in T_i\}$ as the set of vehicles present at minute t .

The constraints of the MILP are as follows. First, the total power allocated at any minute cannot exceed the grid limit, i.e.,

$$\sum_{i \in N_t} p_{i,t} \leq P_{\text{grid}} \quad \forall t \in [T].$$

A vehicle can only draw power within its minimum and maximum limits. Using the binary state variable $s_{i,t}$, these bounds are enforced over each vehicle's connection window T_i as follows:

$$p_{i,t} \geq s_{i,t} \cdot P_{\text{min},i} \quad \forall i \in [n], \forall t \in T_i,$$

$$p_{i,t} \leq s_{i,t} \cdot P_{\text{max},i} \quad \forall i \in [n], \forall t \in T_i.$$

This ensures that $p_{i,t}$ is either exactly zero or restricted within a valid operating range. We calculate the total energy delivered to a vehicle, using a variable $E_{\text{total},i}$, as the sum of the power it receives across its connection window, scaled by $1/60$ to convert kW-minutes into kWh. An upper-bound constraint ensures the system never delivers more energy than requested:

$$E_{\text{total},i} = \sum_{t \in T_i} \left(p_{i,t} \cdot \frac{1}{60} \right) \quad \forall i \in [n],$$

$$E_{\text{total},i} \leq E_{\text{req},i} \quad \forall i \in [n].$$

Variable y_i needs to be one if and only if the total energy delivered is at least 90% of the requested energy. The following constraints couples y_i with this goal:

$$E_{\text{total},i} \geq (0.9 \cdot E_{\text{req},i}) \cdot y_i \quad \forall i \in [n].$$

Note that this inequality allows $y_i = 0$ even when the energy target is fulfilled. However, because our objective function maximizes the sum of y_i , the optimization process naturally forces y_i to 1 whenever a schedule achieves the 90% threshold.

Our objective function is to maximize, lexicographically, $\left(\sum_{i \in [n]} y_i, \sum_{i \in [n]} E_{\text{total},i}\right)$.

For our experiments, we solve the MILPs using the Gurobi optimization solver [7].

3 NP-Hardness

We establish that the offline EV charging scheduling problem studied in this paper is NP-hard. We show that the problem is NP-hard even if the only optimization goal is the number of vehicles that get 90% of the requested energy.

► **Theorem 1.** *The offline EV charging scheduling problem is NP-hard.*

Proof. We reduce from *3-Partition*, which is strongly NP-hard [5]. An instance of *3-Partition* consists of $3m$ positive integers $a_1, \dots, a_{3m}$ satisfying $\sum_{i=1}^{3m} a_i = mB$ and $B/4 < a_i < B/2$ for all i . The question is whether these integers can be partitioned into m groups of three, each summing to exactly B .

Construction. Given a *3-Partition* instance, we construct an EV scheduling instance as follows. We set the planning horizon to $T = m$ minutes, the number of vehicles to $n = 3m$, and the grid capacity to $P_{\text{grid}} = B$. All vehicles are present for the full horizon, i.e., $t_{\text{arr},i} = 1$ and $t_{\text{dep},i} = m$ for every $i \in [n]$. For each vehicle $i \in \{1, \dots, 3m\}$, we set

$$P_{\min,i} = P_{\max,i} = a_i, \quad \text{and} \quad E_{\text{req},i} = \frac{a_i}{60} \text{ kWh}. \quad (1)$$

Setting $P_{\min,i} = P_{\max,i} = a_i$ is a special case of the generic constraints of the problem: a vehicle charged for one minute at rate a_i receives $a_i \times \frac{1}{60} = E_{\text{req},i}$ kWh, which equals $100\% \geq 90\%$ of its requested energy.

We now show that a valid 3-partition exists if and only if all $3m$ vehicles get at least 90% of requested energy.

Suppose first that a valid 3-partition exists, with groups $G_1, \dots, G_m$ each summing to B . Schedule the vehicles in group G_t to charge during minute t . At each minute, the total allocated power is $\sum_{i \in G_t} a_i = B = P_{\text{grid}}$, satisfying the grid capacity constraint. Each vehicle i is charged for exactly one minute at its fixed rate a_i , receiving $a_i/60 = E_{\text{req},i}$ kWh. Consequently, all $3m$ vehicles meet the 90% threshold of requested energy.

For the other direction, suppose that all $3m$ vehicles get at least 90% of requested energy. Since $P_{\min,i} = P_{\max,i} = a_i$, vehicle i receives power only at rate a_i or not at all; each active minute therefore contributes exactly $a_i/60$ kWh. To attain the minimum $0.9 \cdot a_i/60$ kWh, each vehicle must be charged for at least one complete minute. Because $B/4 < a_i$ for all i , any four vehicles require combined power exceeding $4 \cdot (B/4) = B = P_{\text{grid}}$, so at most three vehicles can charge simultaneously. With m minutes and at most three vehicles per minute, the total number of available vehicle-charging slots (minutes) is at most $3m$. Since each of the $3m$ vehicles requires at least one slot, each vehicle is charged for *exactly* one minute, and no slot is left unused. The vehicles assigned to minute t form a group G_t that partitions $\{1, \dots, 3m\}$, with $\sum_{i \in G_t} a_i \leq B$ by the grid constraints. Summing over all m groups gives $\sum_{t=1}^m \sum_{i \in G_t} a_i = mB$, so each group must sum to *exactly* B . This constitutes a valid 3-partition. ◀

4 The Online Optimization Problem

While a solution to the offline problem establishes a theoretical upper bound on performance of any dynamic load-balancing algorithm, its reliance on perfect future knowledge (and also the computational cost of solving a large integer program at runtime) makes it not viable for real-time deployment. Practical facility management requires online algorithms capable of making decisions using only current-state knowledge and historical data. We compare the following approaches for the online optimization.

4.1 Static Load Balancing

The simplest approach is static, equal-share load balancing (SLB), where the grid capacity P_{grid} is divided evenly among all active vehicles. While easy to implement, this approach struggles as the facility scales. When congestion causes the equal-share allocation to drop below the hardware minimum threshold $P_{\text{min},i}$, the vehicles put themselves into an idle state (not using the power to charge). In our experiments, we use a slightly improved static load balancing, in which the system recognizes when a new vehicle would put (some) cars to idle state, and the system simply denies power to newly arriving vehicles to maintain a safe minimum charge rate for all ongoing sessions. When a vehicle departs, the previously denied vehicles are re-considered for being charged by the system based on the first-come, first-served principle.

4.2 Earliest Deadline First

To address the inadequacies of static load balancing, we evaluate an earliest deadline first (EDF) algorithm. Originally formalized by Liu and Layland [11], standard EDF is a greedy heuristic that sorts present tasks based on their deadlines, allocating full capacity sequentially to the most urgent job until the resource limit is exhausted.

To ensure a fair comparison against the MILP's hierarchical objective, we design a modified two-tier EDF. At each time step t , the algorithm partitions connected vehicles into two priority queues. The *SLA queue* contains vehicles that have not yet received 90% or more of the requested energy. The *Top-off queue* contains vehicles that received at least 90% of the requested energy, but not yet 100%.

We iterate through the SLA queue in ascending order of departure time. For each considered vehicle i , we allocate power at its maximum rate $P_{\text{max},i}$, or the remaining grid capacity if that is smaller, provided the allocation meets the minimum threshold $P_{\text{min},i}$. We then subtract the allocated power from the remaining grid budget and continue to the next vehicle. Once all vehicles in the SLA queue have been served, we repeat the same procedure for the top-off queue using any remaining capacity.

A limitation of this approach is that it allocates power strictly by urgency, with no regard for whether a vehicle can still reach its target. For instances where not all vehicles can obtain 90% of requested energy, EDF may spend grid capacity on a vehicle for which the 90% target is not reachable anymore, regardless of future power allocations, at the expense of a vehicle that could still have been satisfied. We address this with a feasibility-aware extension, called EDF-FC.

4.3 Earliest Deadline First with Feasibility Check

At each minute t , before sorting the SLA queue, EDF-FC evaluates whether each vehicle below its requested 90% target can still reach the target under the best possible future allocation, i.e., if it were to receive its maximum power for the remainder of its stay. Let

$\tau_i(t) = t_{\text{dep},i} - t$ denote the vehicle's remaining connection time in minutes. The maximum energy it could still receive is

$$E_i^{\text{max}}(t) = \tau_i(t) \cdot P_{\text{max},i} \cdot \frac{1}{60}.$$

A vehicle is *feasible* if $E_{\text{total},i}(t) + E_i^{\text{max}}(t) \geq 0.9 \cdot E_{\text{req},i}$, and *infeasible* otherwise. This reclassifies the original SLA queue into two tiers, evaluated in this priority order. *Feasible-SLA queue* contains vehicles from the SLA queue that are feasible. *Infeasible queue* contains vehicles from SLA that are infeasible. *Top-off queue* is the same queue as defined before.

Within each queue, vehicles are ordered by ascending departure time, exactly as in standard EDF. Vehicles in the infeasible queue are not excluded from charging, they are still served with any capacity left over from the feasible-SLA queue. After that, EDF-FC considers the top-off queue. It is a natural idea to merge the infeasible queue with the top-off queue. In our internal experiments, the variant of EDF-FC where these two queues are merged did not provide any statistically meaningful difference.

4.4 Imitation Learning

To investigate whether a learned policy can approximate MILP-quality decisions at a negligible runtime cost, we developed an imitation learning (IL) approach using a random forest classifier [2]. We treat the offline MILP as an oracle: for each vehicle at each minute of the training dataset, the MILP's binary charging decision ($s_{i,t} \in \{0,1\}$) serves as our ground-truth label. We then train the classifier to reproduce these decisions using features observable in real time, effectively learning the MILP's prioritization logic without invoking the solver at runtime.

We evaluate two architectures that differ in their output structure. The *individual-vehicle architecture* is a single-target binary classification problem: it takes one vehicle's features alongside global facility state variables and outputs a single charge or no-charge decision for that vehicle. The *facility-wide architecture* is a multi-output binary classification problem: it takes all charger slots as fixed simultaneous inputs, padding empty slots with zeros and a binary occupancy flag, and outputs one binary decision per slot in a single inference call.

The training data is imbalanced, approximately 74% of labels are idle (0) and 26% are active charge (1). Macro F1-score is used as the optimization criterion during hyperparameter search, which weights both classes equally and is appropriate given this imbalance.

Training labels were derived from MILP solutions over two weeks of historical demand data. A time-based split was used: one additional week was held out for validation and one for testing, since EV arrival patterns are strongly time-dependent and a random split would allow future data to leak into training. Hyperparameters were tuned via randomized search using a predefined train-validation split, training on the two-week period and evaluating each hyperparameter combination on the held-out validation week. This ensures that hyperparameter selection is based on generalization to future unseen data rather than within-training performance. The hyperparameters searched were *number of trees*, *maximum tree depth*, *minimum samples per split*, *minimum samples per leaf*, and *class weight*. The final model was retrained on the full two weeks of training data using the best hyperparameters found, and evaluated on the test week. The full feature list is reported in Appendix 2.

For both architectures, the input features at each time step include temporal context (hour of day, day of week), facility state (number of connected vehicles, grid utilization ratio, total capacity), and per-vehicle state (time remaining until departure, energy still needed, laxity ratio, vehicle type, and hardware power bounds). The *laxity ratio* is the time remaining divided by the energy still needed, with a lower value indicating higher charging urgency.

To guarantee physical grid compliance, both models output binary decisions rather than power values, preventing the classifier from producing physically impossible allocations. These decisions are converted into actual power values using a greedy post-processing algorithm. It iterates through vehicles the classifier marked as active, sorted by laxity ratio in ascending order. For each vehicle, it allocates power greedily within the bounds $[P_{\min,i}, P_{\max,i}]$, respecting the remaining grid budget. If a vehicle cannot receive at least $P_{\min,i}$, due to insufficient remaining capacity, it is skipped. Any residual capacity is then distributed to vehicles below their 90% energy target, again sorted by laxity ratio.

4.5 Event-Driven Model Predictive Control

The offline MILP from Section 2 assumes all vehicle data is known in advance. We now adapt this formulation for online use through a model predictive control (MPC) framework, an optimization-based control approach that repeatedly solves a planning problem over a future time horizon and executes the first step(s) of the resulting plan, before re-solving when new information arrives.

At any moment during facility operation, the set of currently connected vehicles and their remaining energy needs are fully known. We can therefore treat the remaining planning window as a new offline instance: given the vehicles currently present, their remaining requested energy, and their departure times, we solve the MILP from the current minute forward to produce an optimal schedule for the rest of their stay. The facility then executes this schedule minute-by-minute until something changes.

The key question is when to re-solve. If nothing changes between minutes, the previously computed schedule remains optimal and there is no reason to recompute it. We only re-solve upon arrival of a new vehicle.

To keep each re-solve computationally small, we separate connected vehicles into two groups. Vehicles that have not yet reached their 90% target are passed into the MILP. Vehicles that have already met the 90% target are handled by a greedy post-processing step: after the MILP runs, any remaining grid capacity is distributed to these vehicles in order of earliest departure time, up to their maximum power. Our choice for using greedy heuristic after the MILP is executed is motivated by the fact that at that moment the only goal is to maximize the amount of delivered energy, for which the greedy allocation based on the earliest departure time performed very well in our internal evaluations.

4.6 Stochastic Event-Driven Model Predictive Control

The event-driven MPC only considers vehicles that are physically present at the moment of re-solving. It does not account for vehicles that are likely to arrive soon. Without this knowledge, the optimizer may assign power slowly to vehicles that appear to have sufficient time remaining to meet their 90% charge target. When new vehicles arrive and share the same grid budget, those vehicles may no longer receive enough power to finish in time, missing a target that was achievable under the original schedule. To address this, we extend the event-driven MPC by adding predicted future vehicles to each re-solve instance.

When an event triggers a re-solve, we generate a set of synthetic future vehicles representing expected arrivals over the next 4 hours. This prediction limit was chosen to match the median observed stay duration in the historical data. These synthetic vehicles are added to the MILP alongside the physically present ones, so the optimizer accounts for expected future demand when computing the current schedule. Only the schedule computed for real vehicles is executed: synthetic vehicles influence the allocation but are never physically charged.

To generate realistic synthetic arrivals, we model the arrival process separately for three day groups (Monday through Thursday, Friday, and weekend), identified through a preliminary similarity analysis of average hourly arrival counts across days of the week. Within each day group, we model the number of arrivals per minute using distributions fitted to historical data. We first tested whether a Poisson distribution was appropriate for each hour of the day using chi-squared goodness-of-fit tests on per-minute arrival counts and Kolmogorov-Smirnov tests on inter-arrival times. For off-peak hours, the Poisson assumption held. For peak hours, arrival counts showed over-dispersion, meaning the variance exceeded the mean, which a Poisson distribution cannot capture. We therefore fitted a negative binomial distribution to peak hours, which adds a dispersion parameter and was found to pass the goodness-of-fit tests across all hours that had previously failed. The breakdown of these distribution selections and their parameter sets are detailed in Table 3 and Table 4 of the appendix.

The arrival rate varies smoothly over time within each day group, estimated using kernel density estimation [15] fitted to historical arrival timestamps. For each synthetic vehicle, attributes such as stay duration, energy requested, and vehicle type are sampled from historical profiles for that hour of the day.

5 Experimental Results and Comparative Analysis

We evaluate how much each scheduling strategy improves upon the static load-balancing baseline as the number of chargers grows from 6 to 40 under the fixed 50 kW grid limit. Fig. 1 shows the number of vehicles meeting the SLA target and Fig. 2 shows total energy delivered for each strategy across this range. Table 1 reports detailed performance for 27 chargers, selected simply because it is the last point at which the offline MILP meets the SLA target for at least 95% of all electric vehicles (here, 95% is an arbitrary, close-to-100 threshold). Full numerical results across all capacities are reported in Table 5 of the Appendix.

All results are based on a single demand dataset from Week 51, 2024. The data were not used for training of the imitation-learning approach. Each strategy produces a fixed result on this input. The stochastic MPC involves sampling and the results reflect a single run.

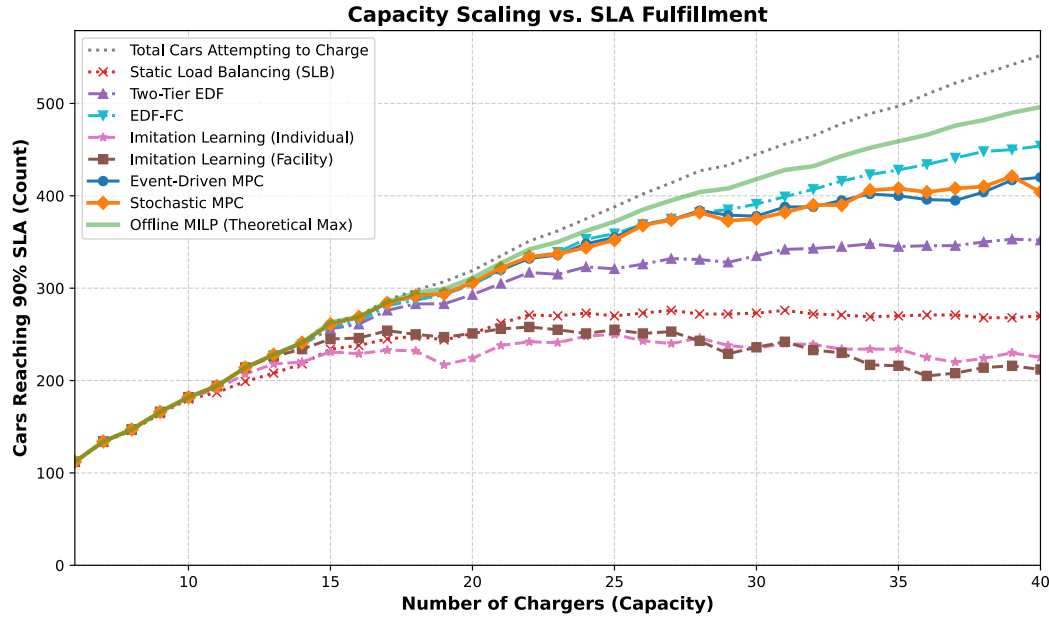
Table 1 Performance at 27 Chargers.

Scheduling Method	90% Charged (Count)	Energy Delivered (kWh)	Total Week Runtime (s)	Mean Step Latency (ms)
Offline MILP	395	5,233.5	-	-
Static Load Balancing (SLB)	276	4,745.9	17.21	0.0941
Two-Tier EDF	332	5,184.2	5.21	0.0090
EDF-FC	375	4,930.9	19.53	0.0356
Imitation Learning (Individual)	240	4,753.8	266.86	19.1075
Imitation Learning (Facility)	253	5,254.4	416.93	30.0551
Event-Driven MPC	374	5,279.8	870.69	0.0344 / 984.3400*
Stochastic Event MPC	374	5,269.3	1,170.25	0.0460 / 1,164.5453*

* Note: Dual values represent *tracking Step* / *Optimization Step* latencies.

5.1 Baseline versus the Offline MILP

We first establish the performance bounds on our dataset. As shown in Fig. 1, SLB satisfies close to 100% of vehicles up to roughly 10 chargers. Beyond this point, total demand frequently exceeds the fixed grid limit. Under SLB, when there is not enough power for



■ **Figure 1** Number of vehicles fulfilling the 90% SLA across number of chargers (6 to 40).

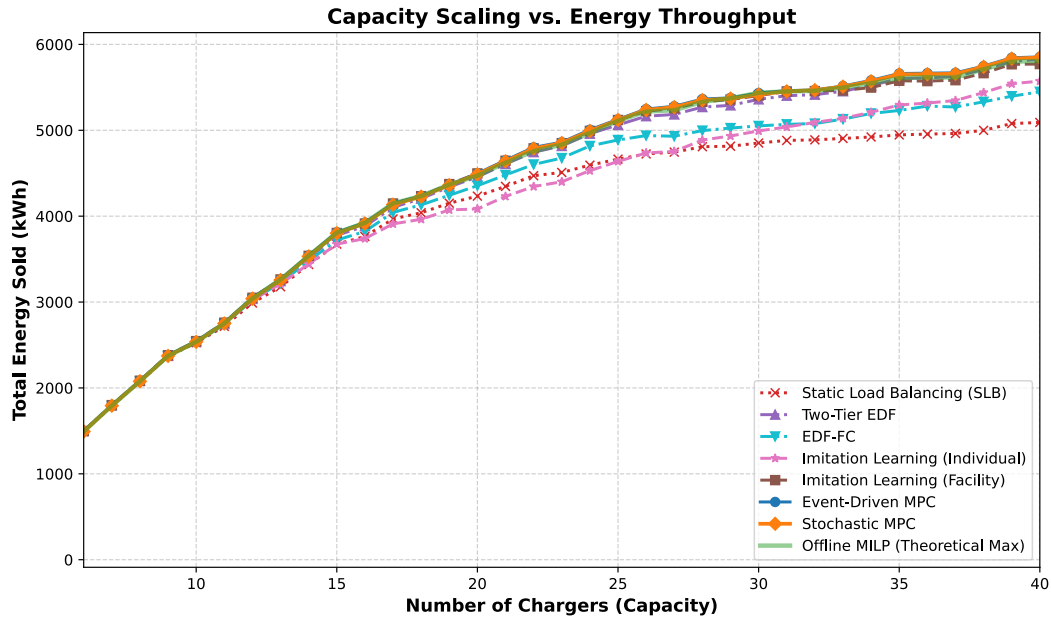
everyone to receive the minimum required charge rate, newly arriving vehicles are denied power so that vehicles already charging can continue to do so. While this protects ongoing sessions, it prevents new arrivals from charging during peak demand, causing SLA fulfillment to plateau after roughly 22 chargers. Interestingly, Fig. 2 shows a matching plateau in energy delivered, but only after 30 chargers. Overall, the results indicate that this baseline strategy cannot effectively scale under congestion on our dataset.

The offline MILP establishes the upper bound, maintaining a 95% of vehicles meeting the SLA target up to 27 chargers and maintaining a 90% of vehicles meeting the SLA target up to 39 chargers on our dataset. At 27 chargers, MILP satisfies 395 vehicles and delivers 5,233.5 kWh, compared to SLB's 276 vehicles and 4,745.9 kWh, a gap of 119 vehicles and 487.6 kWh that represents the maximum recoverable by any online strategy.

5.2 Earliest Deadline First

Despite its simplicity, EDF performs well and far outperforms SLB on our dataset. It satisfies 332 vehicles at 27 chargers, recovering 47% of the gap to the offline MILP in number of vehicles meeting the SLA target (56 vehicles recovered out of the gap of 119 left by the baseline SLB), and closely tracks the offline MILP's energy throughput across the full range tested (Fig. 2). That EDF performs well on total energy-delivered (and, in some cases even outperforms MILP) should not be surprising, since EDF's goal is to allocate as much energy to early-leaving vehicles, thus implicitly aiming to maximize the total-energy delivered as the only optimization goal.

Yet, a visible gap remains in absolute fulfillment of reaching 90% of requested energy (Fig. 1). EDF acts entirely on deadlines and has no way of checking whether a vehicle can still reach its requested energy given its remaining time and energy need. Under heavy congestion it allocates power to vehicles that cannot be satisfied, leaving less capacity for vehicles that could have been.



■ **Figure 2** Total energy sold (kWh) across charger capacities (6 to 40).

5.3 Earliest Deadline First with Feasibility Check

EDF-FC satisfies 375 vehicles at 27 chargers, compared to 332 for standard EDF, closing 43 of the 63-vehicle gap to the offline MILP that standard EDF left unaddressed at this capacity (68%). Energy delivered falls slightly to 4,930.9 kWh, 4.9% below standard EDF's 5,184.2 kWh, despite the higher number of vehicles reaching the SLA target. One of the reasons for this difference is that, since EDF-FC prioritizes charging first the vehicles that can still potentially reach the SLA target, it can happen that a vehicle with early departure does not get charged (because it cannot reach the SLA target) and leaves, and instead another vehicle with later departure gets charged that could have been charged at later time slots as well.

This advantage grows with capacity. As shown in Fig. 1, EDF-FC tracks the MPC strategies closely in terms of the number of vehicles getting 90% of requested energy at lower charger counts, then pulls ahead of both as the facility scales, with the margin widening through the upper end of the tested range. Fig 2 shows the corresponding cost of this advantage: EDF-FC's energy curve sits consistently below the MPC curves across the same range, mirroring the trade-off seen at 27 chargers.

5.4 Imitation Learning Architectures

The imitation learning (IL) strategies failed to match EDF in the number of vehicles meeting the SLA target on our dataset. Both architectures output a binary charge-or-do-not-charge decision per vehicle, and a greedy post-processor converts these into power values without optimizing over the full set of vehicles. The priority ordering learned by the classifier is therefore partly lost, which is reflected in the results: both IL variants satisfy fewer vehicles than EDF despite having access to the same information.

The two architectures diverge on energy delivered. The individual architecture left grid capacity consistently unused, delivering 4,753.8 kWh at 27 chargers. The facility-wide architecture delivered 5,254.4 kWh, actually surpassing the offline MILP's energy delivered,

by learning to activate as many vehicles as possible. This is possible because the MILP’s main objective is to maximize the number of vehicles meeting the SLA target, while IL does not have this direct optimization. However, activating many vehicles without correctly prioritizing them did not translate into SLA gains, as shown in Fig. 1.

5.5 Event Driven and Stochastic MPC

The MPC strategies achieve, together with EDF-FC, the highest number of vehicles meeting the SLA target among all online strategies, matching the offline MILP closely across the full range (Fig. 1). At 27 chargers, the event-driven MPC satisfies 374 vehicles, closing most of the gap to the offline MILP’s 395 vehicles, while delivering 5,279.8 kWh against the offline MILP’s 5,233.5 kWh (Fig. 2), which is by far the most energy delivered among all tested algorithms.

Adding predicted future arrivals in the stochastic MPC yields results almost identical to the reactive version. At 27 chargers both satisfy 374 vehicles, with the stochastic variant delivering 10.5 kWh less. Across the full capacity range the lines overlap almost entirely in Fig. 1, with differences appearing only at a few isolated points. The added predictive complexity offers little advantage, and in some cases even disadvantage, over the reactive event-driven approach on this dataset.

5.6 Computational Feasibility and Trade-offs

Table 1 reports computational performance at 27 chargers. The mean step latency is the average time to compute a power allocation at a single time step.

EDF requires just 0.009 ms per step and completes the full week in 5.21 seconds, making it effectively instantaneous. It satisfies 332 vehicles, recovering most of the gap to the offline MILP without any solver dependency.

EDF-FC adds a small, constant overhead over standard EDF, since the feasibility check is an algorithmic comparison per vehicle per step. Mean step latency rises from 0.0090 ms to 0.0356 ms, and total weekly runtime rises from 5.21 to 19.53 seconds at 27 chargers. Both remain negligible in absolute terms and are cheaper than the IL or MPC strategies, so the feasibility check preserves EDF’s advantage of being solver-free with near-instantaneous computation.

The IL strategies require between 19 and 30 ms per step. The bottleneck is not classifier inference but data preparation: at every time step the controller reconstructs all feature vectors in memory for every active vehicle. Given that their SLA fulfillment actually underperforms the SLB baseline under congestion, this cost is difficult to justify.

The MPC strategies show a split: on minutes where no event has occurred, both variants follow their pre-calculated schedule at 0.034 ms and 0.046 ms per step. At event moments, the event-driven MPC averages 984 ms per optimization step and the stochastic MPC averages 1,164 ms. Because re-solves happen only when the facility state changes, the vast majority of steps run at the tracking rate. Total weekly runtime is 870 seconds for the event-driven MPC and 1,170 seconds for the stochastic variant. The stochastic MPC takes 18% longer per optimization step and adds nearly 300 seconds of total runtime in exchange for satisfying the same number of vehicles and delivering 10.5 kWh less energy.

6 Discussion and Conclusion

The designed heuristic and MPC algorithms outperform, not surprisingly, the SLB system, which confirms that the scheduling algorithms, not the grid limit itself, is the main barrier to expanding the parking facilities.

EDF achieves very good results on real-world data at almost no computational cost, but its main weakness is that it allocates power purely by deadline urgency, with no regard for whether a vehicle can still reach the 90% target given its remaining time and energy need. The improved EDF-FC tackles this issue, quite successfully. At 27 chargers, EDF-FC closes 43 of the 63 vehicles standard EDF left unclosed against the offline MILP, a 68% recovery of that gap, at essentially no added computational cost. Most of EDF's shortfall at this capacity was therefore attributable to misallocated capacity rather than to its lack of joint optimization across vehicles.

A more unexpected result, visible in Fig. 1, is that EDF-FC does not only close ground on the MPC strategies, it overtakes them as the facility scales, with the margin widening toward the upper end of the tested range. This is worth treating with some caution: it is a single-dataset result, and the event-driven MPC theoretically solves an exact lexicographic optimization over all present vehicles at every re-solve, which cannot be worse than a greedy heuristic if no new cars arrive. A plausible explanation for the better performance of EDF-FC is that the MPC's advantage is only local; it optimizes exactly for the vehicles present at the moment of each re-solve, but has no visibility into vehicles that will arrive before the plan is next revisited, so a schedule optimal at the moment it is computed can become suboptimal once new vehicles arrive.

The stochastic MPC adds predicted future arrivals to the event-driven MPC but produces nearly identical results. The arrival distribution was validated and the prediction model is statistically well-grounded, so the issue shall not be the prediction quality. The event-driven MPC already re-solves the moment a new vehicle arrives, so the window in which a prediction could change the schedule is very short. A more accurate forecasting model might improve this slightly, but is unlikely to change the finding that non-predictive event-driven optimization is sufficient under current demand patterns.

The IL models underperformed in all areas. The core issue is that a binary classifier cannot directly control power, so the optimization intelligence in the learned policy gets lost when the post-processor converts decisions into actual kilowatts. A natural extension and a possible future work are machine learning models that output power values directly rather than binary decisions, though any machine learning approach on physical hardware must guarantee that hardware constraints are never violated, which adds further engineering complexity.

All experimental results assume a fixed target of 90% of requested energy. Experiments with other targets is a natural future work, albeit our preliminary experiments show little changes in how the scheduling algorithms mutually compare.

References

- 1 Bahram Alinia, Mohammad Sadegh Talebi, Mohammad H. Hajiesmaili, Ali Yekkehkhany, and Noel Crespi. Competitive online scheduling algorithms with applications in deadline-constrained EV charging. In *Proceedings of the 26th IEEE/ACM International Workshop on Quality of Service (IWQoS)*, pages 1–10, 2018. doi:10.1109/IWQoS.2018.8624184.
- 2 Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:10.1023/A:1010933404324.

- 3 Shiyao Chen, Lang Tong, and Ting He. Optimal deadline scheduling with commitment. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 111–118, 2011. doi:10.1109/Allerton.2011.6120157.
- 4 Jianzhong Du and Joseph Y.-T Leung. Minimizing the number of late jobs on unrelated machines. *Operations Research Letters*, 10(3):153–158, 1991. doi:10.1016/0167-6377(91)90032-K.
- 5 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, CA, USA, 1979.
- 6 Mahmoud Golabi, Abdenmour Azerine, Ammar Oulamara, and Lhassane Idoumghar. Advanced models and a hybrid method for electric vehicle charging scheduling with diverse charger characteristics. *RAIRO-Oper. Res.*, 59(3):1681–1701, 2025. doi:10.1051/ro/2025056.
- 7 Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2024. URL: <https://www.gurobi.com>.
- 8 Alejandro Hernández-Arauzo, Jorge Puente, Ramiro Varela, and Javier Sedano. Electric vehicle charging under power and balance constraints as dynamic scheduling. *Computers & Industrial Engineering*, 85:306–315, 2015. doi:10.1016/j.cie.2015.04.002.
- 9 International Electrotechnical Commission. IEC 61851-1: Electric vehicle conductive charging system—part 1: General requirements. IEC Standard 61851-1, International Electrotechnical Commission, 2017.
- 10 Bruna Cavalcanti Lauro. Semester report of the ke-at-work honour program. Technical report, Department of Advanced Computing Sciences, Maastricht University, Maastricht, The Netherlands, 2025.
- 11 Chang L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, 1973. doi:10.1145/321738.321743.
- 12 Francesco Marra, Guang Ya Yang, Chresten Træholt, Esben Larsen, Claus Nygaard Rasmussen, and Shi You. Demand profile study of battery electric vehicle under different charging options. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–7, 2012. doi:10.1109/PESGM.2012.6345063.
- 13 Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, NY, USA, 2 edition, 2004. doi:10.1007/978-1-4757-4145-2.
- 14 Martijn H.H. Schoot Uiterkamp, Thijs van der Klauw, Marco E.T. Gerards, and Johann L. Hurink. Offline and online scheduling of electric vehicle charging with a minimum charging threshold. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, 2018. doi:10.1109/SmartGridComm.2018.8587556.
- 15 Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.
- 16 I. Zaidi, A. Oulamara, L. Idoumghar, and M. Basset. Maximizing the number of satisfied charging demands of electric vehicles on identical chargers. *Omega*, 127, 2024. doi:10.1016/j.omega.2024.103106.
- 17 Imene Zaidi. *Electric Vehicle Charging Scheduling Problems: Complexity and Solution Approaches*. Phd thesis, L’université de Haute Alsace, Mulhouse, France, 2022.

A Appendix

A.1 Imitation Learning Model Inputs and Features

Table 2 describes the input features used by both machine learning models to make scheduling decisions.

■ **Table 2** Features Used by the Machine Learning Models.

Feature Group	Feature Name	Description
Temporal Context	hour_of_day	Maps the 24-hour daily demand cycle ($0 \dots 23$).
	day_of_week	The days of the week ($0 \dots 6$).
Facility State	connected_ev_count	The total number of connected vehicles ($ N_t $).
	grid_utilization	The total power demand relative to the limit P_{grid} .
	available_capacity	The remaining power headroom at the transformer (kW).
Per-Vehicle State	time_to_departure	Minutes until the vehicle's departure $t_{\text{dep},i}$.
	energy_remaining	Unfulfilled energy required to hit the user target ($E_{\text{req},i}$).
	laxity_ratio	Remaining parking time over remaining energy needed.
	vehicle_type	Fleet category identifier ($\text{BEV} = 1, \text{PHEV} = 0$).
	power_bounds	Bounds on the charge rates ($P_{\text{min},i}, P_{\text{max},i}$).

A.2 Arrival Process Modeling and Fitted Parameters

This section describes the probability distributions used to model future vehicle arrivals in the stochastic MPC framework. We split the week into three day-groups: Early Week (Monday-Thursday), Friday, and Weekend (Saturday-Sunday). Out of the 72 hourly blocks in a week, 44 low-demand hours are modeled using a Poisson distribution (λ), and 28 busy hours are modeled using a Negative Binomial distribution (r, p). Table 3 shows how many hours use each distribution type. Table 4 lists the exact calculated parameters for selected hours across the week.

■ **Table 3** Distribution Choices for the 72 Hourly Blocks.

Day-Group Segment	Poisson Hours	Negative Binomial Hours	Total
Early Week (Mon–Thu)	15 Hours	9 Hours	24 Hours
Friday	14 Hours	10 Hours	24 Hours
Weekend (Sat–Sun)	15 Hours	9 Hours	24 Hours
Total	44 Hours	28 Hours	72 Hours

A.3 Complete Capacity Sensitivity Data Matrix

Table 5 provides the comprehensive numerical results for all evaluated scheduling strategies across the full sensitivity spectrum from 6 to 40 physical chargers under the fixed 50 kW transformer constraint.

■ **Table 4** Fitted Parameters for Selected Hours.

Day-Group	Hour	Time Description	Distribution	Parameters
Early Week	0	Night	Poisson	$\lambda = 0.0002$
	5	Morning Transition	Negative Binomial	$r = 0.9890, p = 0.9343$
	7	Morning Rush	Negative Binomial	$r = 2.5900, p = 0.9389$
	11	Mid-Day	Negative Binomial	$r = 0.5310, p = 0.8294$
	16	Afternoon	Poisson	$\lambda = 0.0658$
	18	Evening	Negative Binomial	$r = 0.8530, p = 0.9540$
Friday	0	Night	Poisson	$\lambda = 0.0003$
	7	Morning Rush	Poisson	$\lambda = 0.1535$
	10	Morning Peak	Negative Binomial	$r = 3.2560, p = 0.9358$
	13	Mid-Day	Poisson	$\lambda = 0.1288$
	19	Evening Peak	Negative Binomial	$r = 0.7010, p = 0.9362$
Weekend	1	Night	Poisson	$\lambda = 0.0000$
	5	Morning Transition	Negative Binomial	$r = 0.0450, p = 0.8351$
	9	Morning Peak	Negative Binomial	$r = 2.2790, p = 0.9108$
	12	Noon Peak	Negative Binomial	$r = 4.9760, p = 0.9194$
	16	Afternoon	Poisson	$\lambda = 0.0792$

■ **Table 5** SLA Fulfillment and Energy Delivered per Strategy Across Changer Capacities (6 to 40).

Cap.	Total Cars	Offline MILP		SLB Baseline		Two-Tier EDF		EDF-FC		IL Individual		IL Facility		Event MPC		Stochastic MPC	
		SLA	Energy	SLA	Energy	SLA	Energy	SLA	Energy	SLA	Energy	SLA	Energy	SLA	Energy	SLA	Energy
6	112	112	1492.8	112	1492.8	112	1498.4	112	1493.3	112	1498.4	112	1498.6	112	1498.4	112	1492.8
7	134	134	1793.3	133	1792.1	134	1799.0	134	1792.6	134	1799.7	134	1799.9	134	1799.7	134	1793.3
8	147	147	2077.2	146	2075.0	147	2081.3	147	2075.3	147	2083.1	147	2084.5	147	2084.1	147	2077.2
9	166	166	2373.2	164	2365.9	166	2377.1	166	2369.5	165	2376.8	166	2381.5	166	2380.5	166	2372.8
10	182	182	2536.4	180	2526.0	182	2540.8	182	2532.5	181	2538.3	182	2545.1	182	2545.1	182	2536.2
11	194	194	2750.6	187	2717.5	194	2749.2	194	2739.3	191	2742.5	194	2759.9	194	2760.2	194	2750.5
12	214	214	3041.0	199	2989.4	214	3036.7	214	3025.6	207	3022.4	214	3049.8	214	3050.8	214	3039.8
13	228	228	3261.2	208	3178.3	227	3243.7	227	3223.9	218	3211.3	226	3263.3	228	3266.1	228	3256.4
14	241	241	3539.1	218	3437.4	239	3510.7	239	3479.4	220	3442.9	234	3538.4	241	3542.3	241	3531.0
15	263	263	3807.4	234	3674.6	256	3776.5	258	3721.7	231	3674.8	245	3804.1	261	3813.0	261	3801.4
16	270	269	3921.7	238	3763.7	261	3883.7	264	3821.8	229	3739.0	246	3913.8	268	3925.6	269	3914.4
17	287	285	4148.9	245	3967.5	276	4102.8	281	4044.6	233	3910.4	254	4147.6	284	4146.7	284	4134.5
18	298	295	4235.1	248	4040.1	283	4204.1	287	4131.1	232	3963.3	250	4232.9	291	4235.8	292	4227.6
19	307	299	4368.3	244	4151.2	283	4340.6	293	4243.3	217	4073.0	247	4373.4	294	4365.3	294	4361.2
20	319	311	4470.5	251	4233.4	293	4456.7	304	4354.6	224	4083.9	251	4497.7	306	4497.5	306	4487.5
21	335	327	4621.9	262	4346.9	305	4607.9	319	4479.5	238	4230.3	256	4645.4	320	4651.1	322	4639.3
22	351	342	4756.3	271	4471.0	317	4745.5	333	4602.2	242	4344.7	258	4789.6	332	4801.1	334	4787.2
23	362	350	4832.1	270	4509.1	315	4817.0	339	4676.6	241	4400.8	255	4849.8	336	4861.7	337	4851.4
24	375	362	4968.1	273	4594.2	323	4960.3	353	4817.7	248	4530.0	251	4993.1	348	5006.2	344	4992.1
25	388	372	5110.1	270	4664.3	321	5062.1	359	4887.3	250	4637.2	255	5119.7	355	5128.1	352	5120.1
26	402	385	5218.0	273	4724.6	326	5163.8	369	4938.1	243	4735.9	251	5222.7	369	5247.4	368	5236.7
27	414	395	5233.5	276	4745.9	332	5184.2	375	4930.9	240	4753.8	253	5254.4	374	5279.8	374	5269.3
28	427	404	5330.8	272	4808.7	331	5270.3	382	4996.3	246	4886.1	243	5331.6	384	5364.4	382	5350.6
29	433	408	5373.1	272	4814.5	328	5290.2	385	5025.7	238	4933.7	229	5357.6	379	5377.2	373	5368.6
30	445	418	5436.8	273	4852.1	335	5360.2	391	5050.8	235	4992.4	236	5403.9	378	5434.0	375	5416.8
31	456	428	5452.5	276	4881.9	342	5402.6	399	5069.5	240	5038.8	242	5452.0	388	5461.7	382	5453.3
32	465	432	5460.0	272	4888.7	343	5414.8	407	5078.2	239	5091.4	233	5456.6	388	5469.4	390	5467.3
33	478	443	5496.5	271	4905.3	345	5454.3	416	5130.2	234	5139.0	230	5464.4	395	5515.6	390	5508.7
34	489	452	5546.6	269	4921.9	348	5521.2	423	5196.1	234	5209.9	217	5497.2	402	5581.7	406	5570.6
35	497	459	5606.7	270	4946.3	345	5602.5	428	5230.8	234	5294.5	216	5576.0	400	5660.9	408	5651.0
36	510	466	5617.7	271	4955.5	346	5610.3	434	5282.0	225	5317.2	205	5572.7	396	5665.2	404	5652.8
37	522	476	5614.5	271	4962.9	346	5626.6	441	5271.1	220	5346.5	208	5583.3	395	5669.1	408	5657.9
38	532	482	5710.8	268	4999.3	350	5708.7	448	5331.7	224	5439.1	214	5665.1	404	5747.4	410	5739.5
39	542	490	5798.8	268	5078.4	353	5800.9	450	5396.9	230	5542.0	216	5768.3	417	5844.8	421	5834.2
40	552	496	5807.1	270	5091.1	352	5828.2	454	5448.2	225	5572.6	212	5771.1	420	5854.3	404	5847.0