Informatics

Scheduling the Charging of Electric Vehicles with SOC-Dependent Maximum Charging Power

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Benjamin Schaden, BSc
Matrikelnummer 01527237

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Günther Raidl
Mitwirkung: Dipl.-Ing. Thomas Jatschka, BSc

Wien, 30. April 2021

Benjamin Schaden | Günther Raidl

Informatics

Scheduling the Charging of Electric Vehicles with SOC-Dependent Maximum Charging Power

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Benjamin Schaden, BSc
Registration Number 01527237

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Günther Raidl
Assistance: Dipl.-Ing. Thomas Jatschka, BSc

Vienna, 30th April, 2021

Benjamin Schaden

Günther Raidl

Erklärung zur Verfassung der Arbeit

Benjamin Schaden, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. April 2021

Benjamin Schaden

Danksagung

Ohne Unterstützung von einigen Personen wäre es nicht möglich gewesen, diese Arbeit zu verfassen.

Ich möchte meinen Betreuern Dipl.-Ing. Thomas Jatscka und Prof. Günther Raidl für die wertvollen Ratschläge danken, die sie mir im Laufe des Projektes gegeben haben. Unsere regelmäßigen Meetings waren nicht nur hilfreich um diese Arbeit thematisch in die richtige Richtung zu lenken, sie gaben mir auch Motivation beim Verfassen dieser Arbeit.

An dieser Stelle will ich auch Honda Research Institute Europe für die Finanzierung dieses Projektes danken. Ein großes Dankeschön an Dr. Steffen Limmer und Dr. Tobias Rodemann, die ihre praktische Erfahrung aus dem Bereich des Ladens von Elektrofahrzeugen einbrachten.

Zu guter Letzt möchte ich auch noch meine Dankbarkeit gegenüber meinen Eltern ausdrücken, die mir das Studium ermöglicht haben. Danke, dass ihr an mich geglaubt habt und für meine Anliegen immer ein offenes Ohr hattet.

Acknowledgements

It would not have been possible for me to write this thesis without the support I received from several people.

I would like to thank my supervisors Dipl.-Ing. Thomas Jatscka and Prof. Günther Raidl, who guided me when working on this project. They gave me valuable advice and provided their professional expertise whenever I got stuck on a problem. Our regular meetings were not only helpful to direct the work into the right research direction, they also gave me motivation boosts.

At this point, I would also like to thank Honda Research Institute Europe for funding this project. In particular, big thanks to Dr. Steffen Limmer and Dr. Tobias Rodemann who brought in their practical experience from the domain of electric vehicle charging.

Finally, I want to express my gratitude towards my parents who made my studies possible. Thank you for believing in me and giving me a sympathetic ear from time to time.

Kurzfassung

In dieser Arbeit betrachten wir die Aufgabe der Erstellung eines Ladeplans für eine Fahrzeugflotte aus der Perspektive einer Elektroladestation. Der Ladeplan soll, unter der Annahme von zeitabhängigen Strompreisen, die gesamten Ladekosten minimieren. Dabei soll auch die zeitliche Verfügbarkeit jedes Fahrzeugs, der Ladestand jedes Fahrzeugs und die maximale Ladeleistung der Ladestation berücksichtigt werden. Ein besonderer Schwerpunkt liegt dabei auf der maximalen Ladeleistung eines Fahrzeugs, die durch eine ladestandsabhängige Funktion limitiert wird. Dieser Aspekt des Problems ist besonders für das Schnellladen von Fahrzeugen relevant.

Da die präsentierten Modelle auf einem diskretisierten Zeithorizont basieren, ist es möglich, dass die vorgeschriebene maximale Ladeleistung eines Fahrzeugs zu Beginn eines Zeitschritts nicht während des gesamten Zeitschritts gehalten werden kann. Wir werden uns dieser Problematik widmen, indem wir die exakte maximale Energie berechnen, die innerhalb eines Zeitschritts geladen werden kann. Außerdem werden wir auch untere und obere Schranken für diese maximale Energie herleiten.

Wir werden verschiedene (gemischt-ganzzahlig) lineare Programme einführen, um das Scheduling-Problem zu lösen. Für eine der Formulierungen zeigen wir, wie man mit Hilfe eines Schnittebenenverfahrens herausragende Laufzeiten erreichen kann. Weiters sehen wir uns den Fall an, bei dem die Ladeleistungsfunktion nicht konkav ist und führen eine Formulierung ein, die stückweise lineare, nicht-konkave Ladeleistungskurven annimmt. Wir verbessern die Laufzeit dieser Formulierung auf einigen Instanzen, indem wir einen Branch-and-Cut-Ansatz anwenden. Außerdem sehen wir uns zwei gemischt-ganzzahlig lineare Formulierungen an, die das Laden ausschließlich in diskreten Energiewerten erlauben.

Anhand selbst erstellter Probleminstanzen werden alle vorgestellten Ansätze experimentell untersucht. Zusätzlich führen wir Experimente in einem *model based predictive control scenario* durch, das üblicherweise in der Praxis zum Einsatz kommt. Es stellt sich heraus, dass das Problem effizient durch lineare Programmierung gelöst werden kann, wenn die Ladeleistungsfunktion der Fahrzeuge konkav und stückweise linear ist. Die Situation ist wesentlich schwieriger für allgemeine stückweise lineare Funktionen. Hier werden deutlich höhere Laufzeiten erwartet, um einen optimalen Ladeplan zu erstellen, da ganzzahlige Variablen in den entsprechenden Modellen notwendig sind. Durch die Approximation der tatsächlichen Ladeleistungsfunktion mit einer konkaven Funktion können wir die

besseren Laufzeiten des speziellen linearen Programms ausnutzen. Jedoch können dabei Ladepläne erstellt werden, die in der Praxis nicht durchführbar sind, da Fahrzeuge ihren gewünschten Zielladestand nicht erreichen. Wir werden diesen Fehler quantifizieren und feststellen, dass dieser in der Praxis vernachlässigbar ist.

Schlussendlich werden wir den Leser bei der Auswahl des passenden Problemlösungsmodells unterstützen und Ratschläge für die Wahl von Problemlösungsparametern geben.

Abstract

We consider the task of finding a charging schedule for a vehicle fleet from the perspective of an electric vehicle charging station. The schedule must minimize the overall charging costs under time-dependent electricity costs while respecting each vehicle's temporal availability, its state of charge, as well as the charging station's maximum charging power. A special focus is put on the aspect that each vehicle's maximum charging power is limited by a function that depends on the vehicle's state of charge, which is particularly important for fast-charging.

Since our presented models are based on a discretized time horizon, a vehicle's maximum charging power may decrease within a single time step. We will show how to deal with this issue by providing an exact derivation for the maximum charging energy of a single time step, as well as lower and upper bounds.

We introduce different (mixed-integer) linear programming formulations to solve the scheduling problem. For one of the formulations we show how to achieve outstanding runtime performance with a cutting plane technique. Also, we consider the case that the maximum charging power function is non-concave by proposing a formulation that can handle piecewise linear, non-concave charging power functions. We enhance its runtime on some instances by applying a branch-and-cut technique. Furthermore two formulations that allow charging in discrete energy units only are considered and their respective linear programming relaxations are compared.

All introduced techniques are experimentally evaluated on benchmark instances, which are partly based on real-world data. We conduct experiments for a model based predictive control scenario, which is typically deployed in practice. It turns out that the problem can be efficiently solved by means of linear programming in case the vehicles' maximum charging power functions are concave and piecewise linear. The situation is remarkably more difficult for general piecewise linear functions where one can expect much higher runtimes for finding an optimal charging schedule, as integral variables are needed in the respective models. By approximation of the maximum charging power with a concave function, we can utilize the performance benefits of a specialized linear programming formulation. By doing so, the charging schedule may be practically infeasible, since vehicles might not reach their specified target state of charge. We will quantify this error and see that it is negligible in practice.

Finally, we will guide the reader on the selection of an appropriate model and give advice how to choose certain problem solving parameters.

Contents

CHAPTER 1

Introduction

The electric vehicle (EV) market is growing like never before in Europe [1]. The rising number of deployed EVs imposes a need for additional charging infrastructure, as well as clever algorithms that schedule the charging of a large amount of EVs. Vehicle charging scheduling is not only limited to industry use-cases anymore, it will sooner or later become an issue which we will face in our daily lives. Publicly accessible charging stations are already pervasive in many countries, an exemplary fast-charging station is shown in Figure 1.1.

A typical scenario that requires such an algorithm, is scheduling the charging of EVs at a company's vehicle charging station. Assume EVs are connected at a charging station and it is known at which times these vehicles will be needed with which state of charges (SOCs). It is required that the vehicles' batteries are charged within the given time frame. Since electricity grid capacities are usually limited, the total amount of power that can be drawn from the charging station at any time is assumed to be bounded.

We assume a time-of-use electricity tariff, i.e., electricity costs per unit of consumed energy vary over time and are assumed to be known prior to scheduling. The goal of the algorithm is to find a charging schedule that respects the aforementioned requirements, while minimizing the overall charging costs.

An aspect that is practically relevant, in particular in case of fast-charging, is the physical limitation of a vehicle's battery. The maximum charging rate of a vehicle is usually not constant throughout the charging process. Instead, it substantially depends on the battery's SOC and tends to be distributed as shown in Figure 1.2. Each vehicle type has its own maximum charging power curve, which is assumed to be known at the time of scheduling. The particular focus of this work will be put on scheduling techniques which take into account such SOC-dependent non-linear maximum charging power functions.

[1] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mckinsey-electric-vehicle-index-europe-cushions-a-global-plunge-in-ev-sales

Figure 1.1: SMATRICS fast-charging station located in front of the Schönbrunn Palace, Vienna.

We deal with a heterogeneous EV fleet, i.e., each vehicle type may have its own maximum charging power curve.

In our proposed problem-solving approaches we assume a discretized time horizon and can therefore only limit the maximum charging power of a single time step with a constant value. The maximum charging power depends on the SOC however, therefore it is not constant within a single time step. In practice, the charging controller might regulate the maximum charging power within a single time step. In order to avoid creating erroneous charging schedules, we consider the maximum charging *energy* instead.

We deduce one *exact* and two *approximate* maximum charging energy functions from the given maximum charging power function. The exact maximum charging energy function specifies the precise energy that can be charged in a single time step. However, using this charging curve with our proposed models can be problematic, since the maximum grid power can be exceeded within a single time step, as we will explain in Section 3.1. Therefore we consider a lower bound maximum energy function that gives us the maximum energy when constantly charging with the smallest maximum power function value within a time step. Analogously, we deduce an upper bound maximum energy function, for which we assume that it is constantly charged with maximum charging power of a time step.

Schedules created with the lower bound function underestimate the maximum charging energy of a vehicle and can therefore be realized in practice flawlessly. However such schedules are possibly suboptimal regarding the total charging costs. On the contrary,

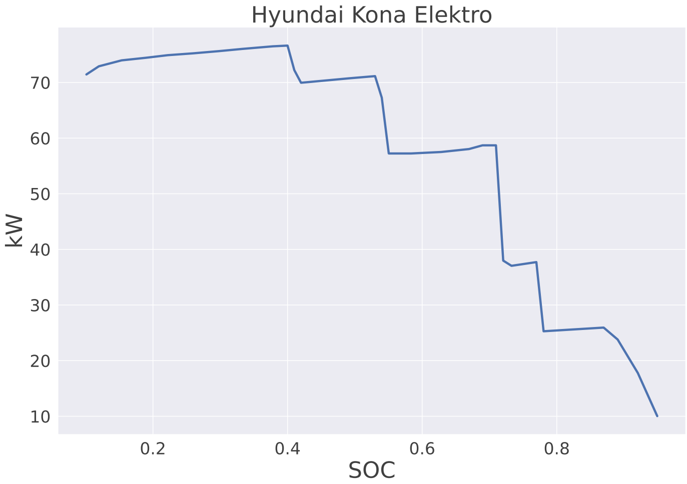


Figure 1.2: Typical maximum charging power of an EV depending on the state of charge. Charging data obtained from Fastned [Fas20].

the upper bound function overestimates the maximum charging energy which can yield practically infeasible schedules. Nevertheless, we are also interested in schedules created with the upper bound function, since they indicate the approximation's impact on the total charging costs.

Additionally, we preprocess the maximum charging *power* function in different ways. On the one hand, we approximate it with a piecewise linear function using different numbers of linear segments. The size of some introduced models depend on this problem parameter and its impact will be analyzed in the experimental results. On the other hand, the maximum charging power function is approximated with a convex hull which results in a concave maximum charging power function. This specific property is exploited with a specialized linear programming formulation that relies on concave maximum power functions.

The effect of the maximum charging power/energy approximation on the solution quality will be studied by means of computational experiments. Through these approximations, infeasible schedules might be created, i.e., vehicles might not reach the desired target SOC. We will quantify this error and also consider the outcome on the total charging costs.

To accommodate incoming EVs and possible changes in the electricity costs or planned departure times of EVs, we assume that a *model based predictive control* (MPC) strategy [CA13] is applied. This means that the scheduling problem is iteratively (re-)solved in an online setting whenever any input data changes or after a certain time has passed, and the actually applied charging plan is continuously adapted accordingly. A rolling horizon is considered for the model based predictive control strategy, which means that the scheduling time horizon is temporally moved forward with every solving iteration. Within the rolling horizon context, vehicles may arrive at different time steps. However, from the perspective of a single schedule, all vehicles are already connected to the charging station and can be charged immediately.

1.1 Aim of the Work

The goal of this work is the development and comparison of different problem-solving approaches for the previously introduced scheduling problem. We first provide a formal problem specification in order to distinguish our problem from other related problems. We clarify why maximum charging energy approximations are necessary to generate sound charging schedules. Then we present different (mixed-integer) linear programming formulations that deal with the maximum charging power constraints in distinct ways. After reading this work, the reader should be aware of how the formulations differ and be able to point out their advantages and drawbacks.

1.2 Outline

The next chapter describes terminology from the field of mathematical programming which is used in the context of our scheduling problem. Basic concepts like *linear programming*, but also more advanced methods, such as the *branch-and-cut algorithm*, are explained.

Chapter 3 formalizes our EV charging scheduling problem. The aforementioned issue that the maximum charging power might be regulated within a single time step is pointed out. We provide an exact derivation of the maximum charging energy from the maximum charging power, as well as simplifying lower and upper bounds. It is also explained how concavity and piecewise linearity relate between the maximum *power* function and the maximum *energy* function.

In Chapter 4, related works are discussed and their key differences are highlighted. We extract the core ideas from these works and relate them to our problem.

Next, Chapter 5 presents different (mixed-integer) linear programming (MILP) techniques, in which we specifically focus on the variable maximum charging power constraints. As we will see, if the maximum charging power function is concave and piecewise linear, we are able to solve the problem efficiently by means of a linear program. For this specific circumstance, we consider an effective cutting plane approach that can also be applied for more general concave maximum power functions which are not necessarily piecewise linear.

Non-concave, piecewise linear charging power functions are more difficult to deal with. We introduce a mixed-integer linear program that uses integral variables in order to deal with such functions. For this model, we will apply a branch-and-cut technique to improve its runtime on some instances.

Furthermore, we will investigate two MILP formulations which allow charging in discrete energy units only. We will compare both formulations to each other by inspecting their

LP relaxations and by showing that one linear programming relaxation is stronger than the other. Practical experiments will confirm our theoretical insights. One of the energy discretized formulations is based on a flow network model. We will argue that its flow variables must be kept integral in general, but may be relaxed to a continuous domain under special circumstances.

Chapter 6 explains how we generate problem instances for our experiments. We consider individual instances and whole model based predictive control scenarios with a rolling horizon. For individual instances, only a single charging schedule has to computed, whereas we iteratively solve multiple schedules in the MPC scenario. One could say that a single MPC instance consists of multiple individual instances. The benchmark instances, both individual and MPC, are partly randomly generated and partly based on real-world data.

In Chapter 7 we present experimental results of benchmarks that were conducted on a high performance cluster. It turns out that the linear programming formulation, which assumes the maximum power function to be concave, shows superior runtime compared to all other shown formulations. Its cutting plane variant is even faster than the static approach and scales better with the number of vehicles.

Compared to the linear program with concave maximum power function, the mixed-integer linear program that can handle non-concave, piecewise linear functions is significantly slower. Its branch-and-cut approach shows performance benefits on some instances, however it is still inferior to the linear program in terms of runtime.

Concerning the energy discretized formulations, we will see that its size and runtime strongly depends on energy discretization factor. Similar to the results in [HPL17], we can also confirm that the network formulation is faster than the other energy discretized formulation.

In our experiments, we additionally take a look at the impact of the of different charging curve approximations regarding charging costs and charging error. We will see that approximating the maximum power function with few piecewise linear segments is hardly noticeable in terms of charging costs and charging error. Moreover, approximating the maximum power function with its convex hull leads to rather small charging errors and charging costs differences, but may greatly improve solving performance. When it comes to the upper and lower bound maximum energy approximation, we will see that both bounds converge to the exact maximum energy function for shorter time interval length.

Finally, Chapter 8 concludes this work and outlines promising future research directions.

CHAPTER 2

Methodological Approach

Before we proceed, let us briefly describe some basic terminology from the domain of mathematical programming, similarly as in [BT97, Wol98].

Linear Programming (LP) deals with the problem of minimizing (or maximizing) a linear cost function subject to linear equality and inequality constraints. For a *general* linear programming problem, we are given a cost vector $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{Q}^n$ and we aim to minimize the linear *objective function* $\boldsymbol{c}'\boldsymbol{x} = \sum_{i=1}^{n} c_i x_i$ over all n-dimensional real-valued vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$, subject to a set of linear equality and inequality constraints $\boldsymbol{Ax} \leq \boldsymbol{b}$ with $\boldsymbol{A} \in \mathbb{Q}^{m \times n}, \boldsymbol{b} \in \mathbb{Q}^m$. Variables $x_1, \ldots, x_n$ are referred to as *decision variables* and the set $\{\boldsymbol{x} \mid \boldsymbol{Ax} \leq \boldsymbol{b}\}$ denotes the *feasible region*. If a solution $\boldsymbol{x}^*$ minimizes the objective function w.r.t. all feasible solutions, it is an *optimal (feasible) solution* and $\boldsymbol{c}'\boldsymbol{x}^*$ are the *optimal costs*. A general linear program is stated as

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}'\boldsymbol{x} \\ \text{subject to} & \boldsymbol{Ax} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

A linear program of the form

$$\begin{array}{rl} \text{minimize} & \boldsymbol{c}'\boldsymbol{x} \\ \text{subject to} & \boldsymbol{Ax} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

is said to be in *standard form*. Any standard form linear program is a special case of a general linear program. Also, each general form linear program can be turned into a standard form problem.

Linear programming problems can be solved in pseudo-polynomial time w.r.t. the size of an input instance, by using the ellipsoid method for example. When speaking of the size of an instance, we refer to the number of bits used to represent a linear programming instance, assuming an appropriate instance format.

Mixed-Integer Linear Programming (MILP) extends linear programming by allowing decision variables to be restricted to integral values. More formally, a mixed-integer program is specified as

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{d}'\boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} \leq \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0} \\ & \boldsymbol{x} \text{ integer} \end{aligned}$$

with $\boldsymbol{A} \in \mathbb{Q}^{m \times n}, \boldsymbol{B} \in \mathbb{Q}^{m \times p}, \boldsymbol{b} \in \mathbb{Q}^m, \boldsymbol{c} \in \mathbb{Q}^n, \boldsymbol{d} \in \mathbb{Q}^p$.

If all decision variables must be integral, we speak of *integer programming*. Moreover, if there are no continuous variables and all elements of $\boldsymbol{x}$ are restricted to $\{0, 1\}$, we speak of *zero-one integer programming*.

Mixed-integer linear programming is more expressive than linear programming due to the additional integrality constraints, however it comes with the drawback that MILP formulations are in general much more difficult to solve. Zero-one integer programming, which is a special case of MILP, belongs to Karp's 21 NP-complete problems [Kar72]. Therefore, mixed-integer linear programming is, in general, computationally intractable.

The Linear Programming Relaxation of a MILP is obtained by allowing all its integral decision variables to be continuous. More formally, the linear programming relaxation of the aforementioned mixed-integer programming problem is defined as

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{d}'\boldsymbol{y} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} \leq \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

Linear programming relaxations play an important role for the comparison of the strength of different MILP formulations. Let A and B be two formulations of the same (mixed-)integer programming problem. We define P_A and P_B to be the feasible regions of the linear programming relaxations of A and B respectively. We say that formulation A is *stronger* than formulation B, if $P_A \subset P_B$.

Most importantly, stronger linear programming formulations may lead to better performance when solving a (mixed-)integer linear program.

The Cutting Plane Method can be used to solve LP problems with a large number of constraints. In the following we will shortly explain how the cutting plane method works.

Given a linear program P, we initially relax P to contain only a restricted subset of constraints (P^{rel}). The relaxed LP problem is solved, which yields a solution $\boldsymbol{x}^*$. Now, we distinguish between two possibilities:

- $\boldsymbol{x}^*$ is also a solution for P. By the fact that any other feasible solution of P is also a feasible solution of P^{rel} and $\boldsymbol{x}^*$ is optimal for P^{rel}, it follows that $\boldsymbol{x}^*$ must be an *optimal* solution for P. The algorithm terminates and returns $\boldsymbol{x}^*$.

- $\boldsymbol{x}^*$ is not a solution for P, i.e., there exists a constraint in P that is violated by $\boldsymbol{x}^*$. We add the violated inequality, a so-called cutting plane, to P^{rel} and re-solve the linear program. The task of determining such a cutting plane is known as the *separation problem*.

The overall procedure is repeated until no more violated constraints are found. In practice it might be beneficial to add more than one cut in a single iteration.

The cutting plane method can also be used to solve a integer linear program P^{int}. In this setting, P^{rel} is initialized with the LP relaxation of P^{int} and it is checked whether $\boldsymbol{x}^*$ is integral in every cutting plane iteration. If not, we add an inequality that all integer solutions of P^{int} satisfy, but $\boldsymbol{x}^*$ does not.

Branch-and-Bound (B&B) is a common framework to solve MILP problems. The idea of a general branch-and-bound approach is the usage of a divide and conquer method, that splits the problem into smaller subproblems. The decomposition can be represented as a search tree, in which we use *lower* and *upper bounds* to prune certain branches of the search space.

A globally best *incumbent* solution with its objective value (global upper bound for a minimization problem) ub is maintained. We update the incumbent solution and its corresponding objective if a better feasible solution has been encountered while implicitly traversing the search tree. Let us give a high level description of a generic branch-and-bound algorithm, similar to [Wol98].

- Select a subproblem S from the list of open subproblems.
- If S is infeasible, delete it. Otherwise compute a lower bound $lb(S)$ for subproblem S.
- If $lb(S) \geq ub$, delete S.
- If $lb(S) < ub$, solve S to optimality. Alternatively, further decompose S into smaller subproblems and add them to the list of open subproblems.

When talking about *LP-based branch-and-bound*, the LP relaxation of the original mixed-integer linear program is used to obtain lower bounds bounds, i.e., $lb(S)$ is given by the objective of the LP relaxation in the current node.

Branch-and-Cut (B&C) embeds the cutting plane method into the branch-and-bound search tree. It is an LP-based branch-and-bound algorithm that solves the LP relaxation at each B&B node with a cutting plane method.

A typical scenario in which it is reasonable to apply the branch-and-cut method is the following. Assume we want to consider a strong (mixed-)integer programming formulation of a problem, which has a tight LP relaxation but a large number of constraints. We want to exploit the tight LP relaxation in order to obtain better bounds, however considering the large set of inequalities from the very beginning would be inefficient. Instead of initially adding all inequalities to the formulation, the problem is relaxed to contain only a small set of constraints. Violated cuts are then augmented in different branch-and-bound nodes, similar to the cutting plane method.

When applying B&C, one needs to think about further questions, e.g. whether cuts should be added only in the current B&B node or even for multiple nodes. On a related note, a *cut-and-branch algorithm* is a branch-and-bound algorithm in which cuts are only generated at the top node of the search tree. In practice, one clearly faces a trade-off between quality of the bounds and time spent with solving the LP relaxation.

A Piecewise Linear Function is a real-valued function that is composed of multiple linear segments. In this work we assume piecewise linear functions to be continuous.

Concave Function. A real-valued function is said to be *concave* on an interval $[a, b]$, if for any $x_1, x_2 \in [a, b]$ and any $\alpha \in [0, 1]$ it holds that

$$f(\alpha x_1 + (1-\alpha)x_2) \geq \alpha f(x_1) + (1-\alpha) f(x_2)$$

Function f is concave on interval $[a, b]$ if and only if $-f$ is convex on interval $[a, b]$.

Convex Set and Convex Hull. Let S be a vector space and $C \subseteq S$. Set C is said to be convex, if for any $x_1, x_2 \in C$ and $\alpha \in [0, 1]$ it holds that $x_1\alpha + x_2(1-\alpha) \in C$.

The *convex hull* of a (possibly non-convex) set C' is defined as the minimal convex set containing C'.

CHAPTER 3

Formal Problem Modeling

The EV charging scheduling problem with SOC-dependent maximum charging power (EVS-SOC) formalizes the task of scheduling the charging of a number of EVs such that the total charging costs are minimized. The charging schedule is preemptive, which means that the charging process of an EV may be interrupted an arbitrary number of times. It is assumed that electricity costs change over time and that they are known in advance. Discrete finite time steps $T = \{0, \ldots, t_{\max}\}$ are used to model the considered time horizon. Each of these represents a time interval of constant duration Δt.

The scheduling is controlled by a single central entity, the so-called aggregator. The total power that can be used from the grid at any time is limited by $P^{\text{gridmax}} > 0$. Electricity costs per unit of consumed energy are given by $c_t > 0$ individually for each time step $t \in T$.

Moreover, the EVS-SOC takes as input a set of n EVs $V = \{1, \ldots, n\}$ that are currently connected to the charging station. For each EV $v \in V$ we are given

- its (planned) departure times $t_v^{\text{dep}} \in T$,
- the initial state of charge $s_{v,0} \in [0, 1]$, i.e., the SOC at the beginning of time step zero,
- the minimum required state of charge when departing $s_v^{\text{dep}} \in [0, 1]$,
- the battery's energy capacity $C_v > 0$, and
- a function $P_v^{\max} : [0, 1] \mapsto \mathbb{R}^+$ indicating the battery's maximum charging power given its SOC; $P_v^{\max}$ must be positive for any SOC less than one and is zero for SOC one; it might or might not be concave.

The goal of EVS-SOC is to find a feasible charging schedule that minimizes the total charging costs while charging each vehicle v from SOC $s_{v,0}$ to (at least) SOC s_v^{dep} by time step t_v^{dep}.

3.1 Maximum Charging Energy Function

Since the maximum charging power function $P_v^{\max}$ depends on the SOC, it is in general not constant within a single time step. This may lead to the problem that a charging power set for a time step is not allowed throughout the whole charging interval. The vehicle's charging controller will then dynamically adjust (reduce) the actually used power to never exceed the SOC-dependent maximum power.

In order to take care of this aspect, we turn from considering the charging power to considering the energy by which an EV can be charged in a time step. We propose alternative approaches how to deduce an (approximate) maximum energy function $E_v^{\max}(s) : [0,1] \mapsto \mathbb{R}^+$ from $P_v^{\max}$ that states the maximum energy by which EV v with SOC s can be charged within duration Δt.

Exact maximum energy. We determine the maximum charging energy $E_v^{\max}$ that is achieved when applying the dynamic charging power $P_v^{\max}$ throughout a whole time step. Considering an EV $v \in V$ with initial SOC $s_{v,t} \in [0,1]$ at some time step $t \in \{0, \ldots, t_v^{\text{dep}} - 1\}$, the time needed to charge the EV to some SOC $s' \in [s_{v,t}, 1]$ using this dynamic maximum charging power is

$$T_v^{\text{min-ex}}(s_{v,t}, s') = C_v \cdot \int_{s_{v,t}}^{s'} \frac{1}{P_v^{\max}(s)}\, ds. \tag{3.1}$$

The maximum energy by which the EV can be charged during a time step of duration Δt is then

$$E_v^{\max}(s_{v,t}) = C_v \cdot (s' - s_{v,t}) \quad \text{s.t.} \begin{cases} T_v^{\text{min-ex}}(s_{v,t}, s') = \Delta t & \text{for} \quad T_v^{\min}(s_{v,t}, 1) > \Delta t \\ s' = 1 & \text{else.} \end{cases} \tag{3.2}$$

Hereby we consider in the else case that charging always stops when SOC value one is reached. While calculating $E_v^{\max}(s_{v,t})$ is non-trivial for general $P_v^{\max}$, it is not difficult to efficiently determine approximate values computationally in a discretized fashion.

To distinguish this calculation of $E_v^{\max}$ from the variants that come in the next paragraphs, we denote it by $E_v^{\text{max-ex}}$.

The problem with this approach is primarily that it is hard to express the maximum grid power constraint since within a time step the actually used power may vary for each EV substantially. Instead, we will only be able to express that the maximum grid power is not exceeded *on average* within a time step. This, however, may be a too weak condition in certain applications. Therefore, we consider the following simpler alternatives modeling lower and upper bounds.

Lower bound. Here we take the largest power that can be constantly applied throughout a whole time step of duration Δt without the charging controller reducing the power. The time needed to charge the EV to some SOC $s' \in [s_{v,t}, 1]$ using the maximum power that can be constantly applied is

$$T_v^{\text{min-lb}}(s_{v,t}, s') = \frac{C_v \cdot (s' - s_{v,t})}{\min_{s \in [s_{v,t}, s']} P_v^{\max}(s)}. \tag{3.3}$$

The maximum energy by which the EV can be charged during a time step of duration Δt is then again obtained by Eq. (3.2) but in conjunction with the above $T_v^{\text{min-lb}}$ (3.3) instead of $T_v^{\text{min-ex}}$ (3.1). We refer to this variant by $E_v^{\text{max-lb}}$.

By avoiding to set for a time step a power that will have to be reduced by the charging controller at some point of time, the obtained maximum energy is a lower bound for the actually obtainable energy $E_v^{\text{max-ex}}$. Using this maximum energy in our whole problem setting means that an obtained solution will guarantee that indeed all EVs are charged to the desired departure SOCs. As we may occasionally use a more restricted charging power than could actually be applied, the schedule might not be optimal in the original sense, and a solution's objective value will be an upper bound for the real optimum.

Upper bound. In contrast to the previous approach, we now take the largest charging power that can be achieved within a time step of duration Δt when not restricting the power otherwise. We make the simplifying assumption that this power can be constantly applied throughout the whole time step, although in reality the charging controller may reduce the charging power for part of the time step.

In this case, the time needed to charge the EV to some SOC $s' \in [s_{v,t}, 1]$ is

$$T_v^{\text{min-ub}}(s_{v,t}, s') = \frac{C_v \cdot (s' - s_{v,t})}{\max_{s \in [s_{v,t}, s']} P_v^{\max}(s)}. \tag{3.4}$$

The maximum energy by which we assume that the EV can be constantly charged during a time step of duration Δt is again obtained by Eq. (3.2), now in conjunction with $T^{\text{min-ub}}$ defined according to (3.4). We refer to this variant by $E_v^{\text{max-ub}}$.

When applying this strategy, we will never underestimate the real maximum energy by which charging can take place, we therefore have an upper bound for $E_v^{\text{max-ex}}$. For a solution to the problem, this means that the departure SOCs might not be reached, and the solution value is a lower bound for an actually optimal solution.

In the lower and upper bound approaches, the obtained errors depend on the choice of Δt. One may choose Δt so that the actual maximum charging power does never decrease within such a time period by more than some ε. The error in energy with which an EV is then charged under assumed maximum power is then bounded by $\varepsilon \cdot \Delta t$.

We want to point out the following relationships between $P_v^{\max}$ and its corresponding maximum energy functions.

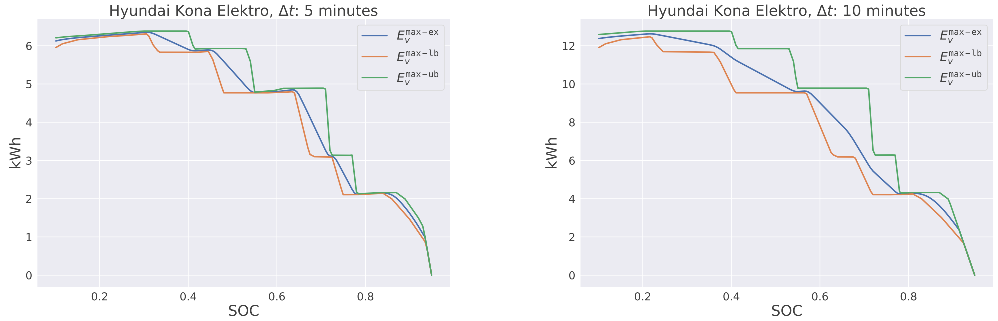


Figure 3.1: $E_v^{\max}$ functions for a Hyundai Kona Elektro for $\Delta t \in \{5, 10\}$ minutes.

- If $P_v^{\max}$ is a piecewise linear function, then $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ are piecewise linear functions as well. On the contrary, $E_v^{\text{max-ex}}$ might not be a piecewise linear function, even if $P_v^{\max}$ is piecewise linear.

- If $P_v^{\max}$ is a concave function, so is $E_v^{\text{max-lb}}$, $E_v^{\text{max-ub}}$, and $E_v^{\text{max-ex}}$.

To give the reader an impression how $E_v^{\text{max-lb}}$, $E_v^{\text{max-ex}}$, and $E_v^{\text{max-ub}}$ relate to each other, Figure 3.1 shows these functions for different Δt values for a Hyundai Kona Elektro. Note that the area between $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ decreases with smaller Δt values.

In the following chapters we will pursue all three energy functions and investigate the pros and cons of each in comparison. We will use the notation $E_v^{\max}$ as a placeholder for any specific energy function from $\{E_v^{\text{max-ex}}, E_v^{\text{max-lb}}, E_v^{\text{max-ub}}\}$.

Converting Energy back to Power. In practice, the charging aggregator usually regulates the maximum charging *power* instead of the maximum charging *energy*. Consequently when scheduling with energy values we have to convert back energy values to power values. For schedules created with $E_v^{\text{max-lb}}$, the computed energy values of a schedule can be divided by Δt to obtain charging power values that can be constantly applied throughout a single time step. Due to the definition of $E_v^{\text{max-lb}}$ it is ensured that $P_v^{\max}$ is not exceeded. On the other hand, for schedules created with $E_v^{\text{max-ex}}$ or $E_v^{\text{max-ub}}$, we can not apply the same conversion since $P_v^{\max}$ might be exceeded then.

3.2 Non-Linear Model

We now formally define EVS-SOC by the following non-linear program, where variables $x_{v,t}$ represent the energy by which EV $v \in V$ is charged in time step $t = 0, \ldots, t_v^{\text{dep}} - 1$. Variables $s_{v,t}$ indicate the SOC of each EV $v \in V$ at the beginning of each time step

$t = 0, \dots, t_v^{\text{dep}}$.

$$\min \quad \sum_{v \in V} \sum_{t=0}^{t_v^{\text{dep}}-1} c_t \cdot x_{v,t} \tag{3.5}$$

$$x_{v,t} \leq E_v^{\max}(s_{v,t}) \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1 \tag{3.6}$$

$$\sum_{v \in V | 0 \leq t < t_v^{\text{dep}}} x_{v,t} \leq \Delta t \cdot P^{\text{gridmax}} \qquad t \in T \tag{3.7}$$

$$s_v^{\text{dep}} \leq s_{v,t_v^{\text{dep}}} \qquad v \in V \tag{3.8}$$

$$s_{v,t} = s_{v,t-1} + x_{v,t-1}/C_v \qquad v \in V,\ t = 1, \dots, t_v^{\text{dep}} \tag{3.9}$$

$$0 \leq x_{v,t} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1 \tag{3.10}$$

$$0 \leq s_{v,t} \leq 1 \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{3.11}$$

The objective function (3.5) minimizes the sum of the costs for the total consumed energy over all time steps. Inequalities (3.6) ensure that the energy by which each EV is charged during each time step does not exceed the SOC-dependent maximum energy. Note that this inequality is in general non-linear. Constraints (3.7) limit the total energy consumed from the grid during each time step to $\Delta t \cdot P^{\text{gridmax}}$. The departure SOCs are enforced by Inequalities (3.8). Equalities (3.9) determine the SOC at the beginning of each other time step $t = 1, \dots, t_v^{\text{dep}}$ for each EV v. Thereunto the previous state of charge $s_{v,t-1}$ is considered together with the charging rate of the previous time slot $x_{v,t-1}$, the time interval length Δt and the total battery capacity C_v. Variable domains are defined in (3.10) and (3.11). Due to the domain of variable $x_{v,t}$, an EV may not discharge.

We remark that in practice, the domain of $P_v^{\max}$ is often not defined on the entire SOC interval $[0, 1]$ but just for some restricted $[s_v^{\min}, s_v^{\max}]$, $0 < s_v^{\min} < s_v^{\max} < 1$. In the following, we will regard this issue as an implementation detail and assume the domain of $P_v^{\max}$ to be $[0, 1]$.

CHAPTER 4

Related Work

Substantial research has been done in recent years on the electric vehicle charging scheduling problem. Although it might seem that most contributions deal with the same problem, there exist significant differences in the problem formulations. Authors usually introduce their own unique problem setting, which makes it difficult to compare results and ideas of different works on this topic. To make the reader aware of the assumed problem characteristics, we briefly point out the differences in the following.

Concerning the electricity pricing model, most existing work deals with a time-of-use (TOU) tariff, in which electricity prices depend on the current time of the day [CTL$^+$12, HPL17, KCE18]. These prices are usually determined by the electricity provider one day in advance. Other works, instead, assume uncertain bounded electricity prices and handle uncertainty with robust optimization for instance [KS17].

When it comes to the scheduling task itself, there are two common approaches how to distribute the task. In one of them, scheduling is performed by a central entity, a so-called aggregator, which is given full information needed for the scheduling task to create a global schedule for all EVs [HPL17, KS17]. On the other hand there is distributed scheduling, where each vehicle makes scheduling decisions on its own [MN19].

In several works it is additionally assumed that the charging station only has limited charging capacities [HPL17, KS17]. Some of these works describe a scenario in which it is allowed to exceed these capacities by paying additional costs [HPL17]. In other works, a certain amount of electricity can be taken for free from a local power supply at the charging station, e.g. a photovoltaic system. Additional required power can be obtained from the electricity grid at a TOU tariff [JZ19, KS20].

Another problem characteristic, at which the particular focus of this work lies, is the battery's maximum charging rate. Most existing works assume a maximum charging rate that remains constant for the whole time a battery is charged, see, e.g. [HHS10, SHTT18]. In practice, however, the typical lithium-ion-battery charging profile has variable maximum charging power that strongly depends on the state of charge, as we already pointed out in the introduction. Considering this SOC-dependent maximum charging power more carefully is particularly important in applications where fast-charging is applied and the maximum charging power is not so much limited by the provided energy or charging station. Literature on scheduling the charging under consideration of variable maximum charging power is scarce yet.

Some authors consider the fact that lithium-ion batteries have three charging phases [PCdA14]. In the first phase, the battery is totally depleted and charged slowly. During the second „constant-current“ phase, the battery is filled up to approximately 70% of its total capacity with constant current. The remaining energy is delivered in the third „constant-voltage“ phase, in which the charging current decreases and the charging voltage stays constant. To incorporate the three charging phases, the authors of [PCdA14] include voltage and current in their model.

In other works [MCDM20], a maximum charging power function is deduced from various parameters of electrical engineering, like battery voltage, output impedance and battery capacity. A Second-Order Cone Program is then proposed to solve a scheduling problem with „battery voltage awareness“. In our work we abstract from these specifics and only take variable charging power in dependence of the SOC into account.

Similarly, the authors of [SB10] describe an approach to generate electric vehicle charging schedules with the goal of minimizing charging costs and respecting the battery's charging profile. They introduce a nonlinear equation that represents the maximum charging power in dependence of the battery's internal resistance and its SOC. Moreover, they claim that the nonlinear charging profile can be approximated with linear equations. A linear and quadratic formulation is proposed and it is experimentally evaluated whether charging with the linear formulation exceeds the actual maximum charging power. The authors conclude the work with the insight that the violation of the power constraints is relatively small and suggest the usage of the linear formulation due to its better runtime.

Some authors distinguish on another aspect of charging schemes, specifically on the property whether the charging process of an EV can be interrupted (is preemptive) or not (is non-preemptive). Both modeling approaches can be found in [HPL17].

As we have seen, many similar problem variants exist in the literature. The authors of [MG15] provide a broad overview about related works and deployed problem solving methods. In the following, we will shortly recap most important works that are closely related to our problem.

Cao et al. [CTL$^+$12] consider a similar problem setting in which also a time-of-use electricity tariff is assumed. They deal with a maximum charging power function of a single vehicle. The authors propose a heuristic that aims to minimize the total charging costs while filling valleys in the energy load demand and respecting the SOC-dependent maximum charging power.

El-Bayeh et al. [EBMS$^+$18] study the impact of considering of a maximum charging power function on the total charging costs. They don't only respect maximum charging profiles, but also take discharging profiles into account. In their problem setting, a nonlinear SOC-dependent maximum charging power function (called battery power profile in their work) is assumed to be given. They approximate it with a piecewise linear function and introduce an algorithm that fits the original battery power profile to the piecewise linear function. Subsequently, they draw a comparison between the charging costs when charging with a constant maximum charging power and the charging costs when charging with a vehicle specific SOC-dependent piecewise linear function. Thereunto they utilize a mixed-integer nonlinear program, which distinguishes their approach from our problem solving techniques. The authors adopt three different pricing mechanisms (fixed pricing, time-of-use pricing, dynamic pricing) and conclude their work with experimental results. They point out the annual charging cost differences that can be saved by respecting the SOC-dependent maximum charging power function.

Isihara and Limmer [IL20] give a basic formulation of an EV charging scheduling problem without consideration of variable maximum charging power. With the exception of this aspect, EVS-SOC essentially corresponds to their C-C variant, where discharging is not allowed and the charging power can be chosen continuously.

Limmer [Lim20] provides a linear programming formulation that considers multiple objectives. Minimizing the peak load, maximizing the satisfaction of EV charging demands under load limit constraints, or minimizing the electricity costs with help of trading on the electricity market are typical used objectives, that might conflict with each other. Based on preferences defined by the charging station operator, a trade-off has to be made.

A common approach to handle multiple objectives in a linear program is to combine them via a weighted sum. However, it might be difficult to determine these weights in practice. Therefore a lexicographical approach is presented, where different objectives are arranged in a hierarchy. The program can be seen as an alternative to the weighted sum approach, but may also be useful for determining weights for it. Limmer mentions advantages and drawbacks for both approaches.

Han, Park, and Lee [HPL17] consider a similar problem setting as EVS-SOC. The authors assume that the charging station has limited grid capacity, which may be exceeded at the price of paying penalty costs.

Moreover, this work distinguishes charging schemes on the property whether or not the charging process can be interrupted (i.e., is preemptive) or not (i.e., is non-preemptive). For both charging schemes a mixed-integer program is presented. Since the formulation for the preemptive scheme is quite large and impractical on large instances, the authors additionally introduce an extended formulation which makes use of a transition network. The idea behind the extended formulation is to let the mixed-integer linear programming solver exploit the intrinsic network structure of the formulation. As experiments show, the extended formulation has in practice a better run-time performance.

In general, preemptive charging schemes can interrupt the charging process arbitrarily often. Frequent charging interruptions, however, may introduce deterioration for batteries or threaten the stability of the charging network. Therefore, the authors address this issue by limiting the number of charging interruptions. This is accomplished by modifying the MILP formulation such that frequent interruptions impose additional costs in the objective function.

Korolko and Sahinoglu [KS17] also address optimal electric vehicle charging in an unregulated electricity market. The problem setting and inputs are similar to the ones in [HPL17]. Again, the authors deal with variable electricity prices that are assumed to be known at the time of scheduling. A discretized time horizon is assumed as well.

Contrary to EVS-SOC, the problem is initially considered from the perspective of a single EV and a non-linear formulation is developed for it. Subsequently, it is shown how to solve the non-linear program using a cutting plane technique together with a linear programming (LP) solver. More concretely, the main idea of the method is to solve the non-linear program by solving a sequence of linear programs. The algorithm starts with a relaxation of the non-linear program, which is obtained by omitting all non-linear constraints from the original non-linear formulation. The resulting relaxation is a linear program. Based on the solution of the relaxation (which can be easily obtained using an LP solver), one tries to identify violated non-linear constraints. It is shown how to construct a corresponding hyperplane, which is in turn added to the linear program to cut off the current solution. This procedure is iteratively repeated until a certain solution quality is reached.

The work continues by extending the formulation to multiple EVs. By doing so, one introduces binary decision variables, which determine the assignment of EVs to time shifts. The resulting non-linear program can still be solved using the previously explained cutting plane procedure, however, one needs a MILP solver now.

Last but not least, the authors also come up with a robust optimization approach that takes into account the uncertainty of electricity prices.

Finally, we want to remark that none of these previous works considered the issue that the variable maximum charging power in general varies also within a time step of the discretized time horizon. We are the first considering this aspect in more detail.

CHAPTER 5

Problem Solving Approaches

In the following we study different ways to deal with the non-linear maximum charging energy constraints (3.6). We first consider the simpler case that the maximum power function is concave, where we essentially can solve the problem with an LP formulation or a cutting plane approach.

5.1 Concave Maximum Energy Functions

As already mentioned before, if $P_v^{\max}$ is continuous and concave, it follows that also $E_v^{\max} \in \{E_v^{\text{max-ex}}, E_v^{\text{max-lb}}, E_v^{\text{max-ub}}\}$ is concave as well.

In the following, we will further assume that $E_v^{\max}$ is differentiable. We are aware that, depending on $P_v^{\max}$, this assumption might not be completely valid. Actually, $E_v^{\max}$ might have breakpoints, in which the left-sided and right-sided limits of the differential do not coincide. Nevertheless, we will treat $E_v^{\max}$ as if it were differentiable at any SOC of its domain, since differing left-sided and right-sided limits will not affect the results of the following modeling approach.

Due to the assumed properties of $E_v^{\max}$, we can replace the non-linear Inequality (3.6) from EVS-SOC with the combination of the infinite set of linear inequalities

$$x_{v,t} \leq E_v^{\max\prime}(\hat{s}) \cdot (s_{v,t} - \hat{s}) + E_v^{\max}(\hat{s}) \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1,\ \hat{s} \in [s_{v,0}, s_v^{\text{dep}}] \quad (5.1)$$

where $E_v^{\max\prime}$ is the first derivative of $E_v^{\max}$. We call the resulting linear programming model EVS-SOC-LIN.

Note that if $P_v^{\max}$ is a piecewise linear function, then also $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ are piecewise linear functions. The set of inequalities reduces then to a finite one where we have one inequality corresponding to each linear function segment.

In the spirit of [KS17], who essentially consider a similar kind of inequalities, we can solve EVS-SOC-LIN by a cutting plane approach. Thereby the relaxation of EVS-SOC-LIN without Inequalities (5.1) is first solved. Then, Inequalities (5.1) that are violated by the current LP solution are iteratively determined, added, and the LP problem is re-solved. The process is repeated until no more Inequalities (5.1) are violated.

The separation of a violated inequality for a current solution $(x^{\mathrm{LP}}, s^{\mathrm{LP}})$ to the relaxed EVS-SOC-LIN works as follows. For all $v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 1$, we check if $x_{v,t}^{\mathrm{LP}} > E^{\max}(s_{v,t}^{\mathrm{LP}})$. In this case we add the violated Inequality (5.1) for vehicle v, time step t, and $\hat{s} = s_{v,t}^{\mathrm{LP}}$. Note that for one vehicle, multiple inequalities for different time steps can be added within a single cutting plane iteration. This separation procedure is performed for all vehicles $v \in V$ and as long as any violated inequalities are found, the augmented LP problem is re-solved.

An alternative to the above is the following. Whenever $x_{v,t}^{\mathrm{LP}} > E^{\max}(s_{v,t}^{\mathrm{LP}})$ for some EV v and time step t, one can add the violated Inequality (5.1) not only for time step t but for all time steps $t' = 0, \ldots, t_v^{\mathrm{dep}} - 1$. The intention here is to possibly reduce the number of needed resolving iterations, but clearly the size of the LP formulation increases more rapidly. Preliminary experiments indicated that indeed this variant performs better in practice in most cases. Therefore we apply it in all our experiments documented in the remainder of this work.

We also compared this variant with the approach presented in [KS17]. In our approaches, we check for all time steps $t = 0, \ldots, t_v^{\mathrm{dep}} - 1$ whether Inequality (5.1) is violated, whereas in [KS17] only the smallest time step that violates Inequality (5.1) is augmented in each cutting plane iteration. We found that our variant performs slightly better for our problem instances.

5.2 General Piecewise Linear Maximum Energy Functions

In the following model, we require for each EV $v \in V$ that the maximum charging energy function $E_v^{\max}$ is a piecewise linear function. Additionally, it is assumed that each $E_v^{\max}$ function is non-concave, regardless of whether $E_v^{\max}$ is actually concave or not. We assume that we are given a finite set of SOC values $\{S_{v,k} \mid k = 1, \ldots, k_v^{\max}\}$ in sorted order, with $S_{v,1} = 0$ and $S_{v,k_v^{\max}} = 1$ and the values in between representing the breakpoints of the piecewise linear function. These values are pairwise distinct and can be unevenly distributed among the SOC interval $[0, 1]$. For each $S_{v,k}$ we know the value of the maximum charging energy $E_v^{\max}(S_{v,k})$.

If we assume again the maximum charging energy function $E_v^{\max}$ to be concave, we can replace (3.6) in formulation (3.5–3.11) with

$$x_{v,t} \le a_{v,k} s_{v,t} + b_{v,k} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 1,\ k = 2, \ldots, k_v^{\max} \tag{5.2}$$

such that

$$a_{v,k} = \frac{E_v^{\max}(S_{v,k}) - E_v^{\max}(S_{v,k-1})}{S_{v,k} - S_{v,k-1}} \quad \text{and} \quad b_{v,k} = E_v^{\max}(S_{v,k-1}) - a_{v,k} S_{v,k-1}. \tag{5.3}$$

However, if $E_v^{\max}$ is not concave, we need additional variables and constraints, which ensure that only one piecewise linear function is active in each SOC interval. We model the piecewise linear function as suggested in chapter 10.1 of [BT97].

Thereunto we use continuous variables $\alpha_{v,t,k}$ to express the SOC $s_{v,t}$ as a convex combination of $S_{v,k}$ and $\alpha_{v,t,k}$. The variables $\alpha_{v,t,k}$ are also used to represent the maximum charging energy function as a convex combination of $E_v^{\max}(S_{v,k})$ and $\alpha_{v,t,k}$.

Furthermore we introduce additional binary variables $\beta_{v,t,k}$, which are used to ensure that at most two consecutive $\alpha_{v,t,k}$ and $\alpha_{v,t,k+1}$ variables are non-zero. By replacing Constraints (3.6) in formulation (3.5–3.11) with the following Constraints (5.4–5.12), we obtain a MILP problem, which we refer to as EVS-SOC-GLIN.

$$s_{v,t} = \sum_{k=1}^{k_v^{\max}} S_{v,k} \cdot \alpha_{v,t,k} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{5.4}$$

$$x_{v,t} \leq \sum_{k=1}^{k_v^{\max}} E_v^{\max}(S_{v,k}) \cdot \alpha_{v,t,k} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1 \tag{5.5}$$

$$\sum_{k=1}^{k_v^{\max}} \alpha_{v,t,k} = 1 \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{5.6}$$

$$\sum_{k=1}^{k_v^{\max}-1} \beta_{v,t,k} = 1 \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{5.7}$$

$$\alpha_{v,t,0} \leq \beta_{v,t,0} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{5.8}$$

$$\alpha_{v,t,k} \leq \beta_{v,t,k-1} + \beta_{v,t,k} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}},\ k = 2, \dots, k_v^{\max} - 1 \tag{5.9}$$

$$\alpha_{v,t,k_v^{\max}} \leq \beta_{v,t,k_v^{\max}-1} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} \tag{5.10}$$

$$0 \leq \alpha_{v,t,k} \leq 1 \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}},\ k = 1, \dots, k_v^{\max} \tag{5.11}$$

$$\beta_{v,t,k} \in \{0,1\} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}},\ k = 1, \dots, k_v^{\max} - 1 \tag{5.12}$$

Equations (5.4) link the SOC values $s_{v,t}$ with the continuous weight variables $\alpha_{v,t,k}$. The charging energy $x_{v,t}$ of EV v at time slot t is limited by Inequalities (5.5) to the maximum charging energy. Constraints (5.6) set the sum of the continuous weights $\alpha_{v,t,k}$ over all discrete SOC levels $k = 1, \dots, k_v^{\max}$ to one. Equations (5.7) ensure that exactly one $\beta_{v,t,k}$ variable is active for each EV v and time slot t. The $\alpha_{v,t,k}$ variables are linked with the $\beta_{v,t,k}$ variables by Inequalities (5.8–5.10). Altogether, (5.7–5.10) are the so-called adjacency constraints, which ensure that at most two consecutive $\alpha_{v,t,k}, \alpha_{v,t,k+1}$ variables are non-zero. Constraints (5.11–5.12) define the domain of $\alpha_{v,t,k}$ and $\beta_{v,t,k}$ respectively.

5.2.1 Branch-and-Cut Approach

As we will see in Chapter 7, the EVS-SOC-LIN formulation, which requires $E_v^{\max}$ to be concave, performs remarkably well. The idea of the following approach is to exploit the fast runtimes of EVS-SOC-LIN, while extending it to deal with general, possibly non-concave $E_v^{\max}$ functions.

We introduce a branch-and-cut approach, in which we initially solve a relaxation of EVS-SOC-GLIN. Instead of restricting $x_{v,t}$ to the piecewise linear $E_v^{\max}$ function, $x_{v,t}$ now may take any value from the convex hull of $\{(S_{v,k}, E_v^{\max}(S_{v,k})) \mid k = 1, \ldots, k_v^{\max}\} \cup \{(S_{v,0}, 0), (S_{v,k}, 0)\}$.

To obtain the relaxation, we consider the original EVS-SOC-GLIN formulation with all its variables and constraints except the linking constraints (5.8–5.10). Observe that contrary to the original formulation, two non-consecutive $\alpha_{v,t,k}$ variables may be non-zero now.

Whenever a solution candidate is found, we check for all $v \in V,\ t = 0, \ldots, t_v^{\text{dep}} - 1$ whether $x_{v,t}$ exceeds the actual $E_v^{\max}$ value at SOC $s_{v,t}$, i.e., we check if $x_{v,t} > E_v^{\max}(s_{v,t})$. If this is the case, a cut is added which links all non-zero $\alpha_{v,t,k}$ variables with their respective $\beta_{v,t,k}$ variables, as we did in Constraints (5.8–5.10). Such cuts are separated and added until for all $v \in V,\ t = 0, \ldots, t_v^{\text{dep}} - 1$ it holds that $x_{v,t} \leq E_v^{\max}(s_{v,t})$.

5.3 Discretization of Energy

The so-called PCP MILP model in [HPL17] is based on the idea of allowing only a discrete set of SOC levels and using binary indicator variables for these levels for each vehicle and each time step. A significant difference of the problem considered there and our formulation, however, is that in [HPL17] the charging power always has to be either zero or the maximum allowed value. In this way, discrete charging levels appear naturally, while we have to do a more explicit discretization to follow similar principles.

We therefore define by EVS-SOC-ΔE the variant of our original EVS-SOC problem in which energy is considered in discrete units of ΔE only, i.e., only multiples of ΔE can be charged in each time step Δt. Instead of continuous SOC values, we are now given for each vehicle $v \in V$ a rounded initial energy level at arrival $y_{v,0}$ and the required departure energy level $y_v^{\text{dep}} \in \mathbb{N}$, both as integral multiples of our unit ΔE. Coming from our original EVS-SOC problem definition, corresponding approximate values for the initial and departure energies can be obtained by rounding, i.e.,

$$y_{v,0} = \lfloor s_{v,0} \cdot C_v / \Delta E \rfloor \quad \text{and} \quad y_v^{\text{dep}} = \lceil s_v^{\text{dep}} \cdot C_v / \Delta E \rceil \tag{5.13}$$

are the numbers of energy units ΔE at the beginning and at departure, respectively. Then, it is ensured that $y_{v,0} < y_v^{\text{dep}}$ and vehicle v can have $y_v^{\text{dep}} - y_{v,0} + 1$ SOC values $\{k \cdot \Delta E / C_v \mid k = y_{v,0}, \ldots, y_v^{\text{dep}}\}$.

We now adapt EVS-SOC (3.5–3.11) to EVS-SOC-ΔE by replacing variables $s_{v,t}$ with integer variables $y_{v,t}$ that indicate the number of energy units in each vehicle v stored at the beginning of each time step t. Variable $x_{v,t}$ now represents the number of energy units ΔE charged into vehicle v at time step t.

$$\min \quad \Delta E \cdot \sum_{v \in V} \sum_{t=0}^{t_v^{\mathrm{dep}}-1} c_t \cdot x_{v,t} \tag{5.14}$$

$$x_{v,t} \leq \left\lfloor \frac{1}{\Delta E} \cdot E_v^{\max}\left(y_{v,t} \cdot \frac{\Delta E}{C_v} \right) \right\rfloor \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1 \tag{5.15}$$

$$\sum_{v \in V \mid 0 \leq t < t_v^{\mathrm{dep}}} x_{v,t} \leq \lfloor \Delta t \cdot P^{\mathrm{gridmax}} / \Delta E \rfloor \qquad t \in T \tag{5.16}$$

$$y_{v,t_v^{\mathrm{dep}}} = y_v^{\mathrm{dep}} \qquad v \in V \tag{5.17}$$

$$y_{v,t} = y_{v,t-1} + x_{v,t-1} \qquad v \in V,\ t = 1, \dots, t_v^{\mathrm{dep}} \tag{5.18}$$

$$x_{v,t} \in \{0, \dots, x_v^{\max}\} \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1 \tag{5.19}$$

$$y_{v,t} \in \{y_{v,0}, \dots, y_v^{\mathrm{dep}}\} \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} \tag{5.20}$$

The upper limit of the domains of the $x_{v,t}$ variables is $x_v^{\max} = \max_{k=y_{v,0},\dots,y_v^{\mathrm{dep}}} \lfloor E_v^{\max}(k \cdot \Delta E / C_v) / \Delta E \rfloor$.

We now introduce variables $\lambda_{v,t,k} \in \{0,1\}$ which indicate with value one that $y_{v,t} = k$, i.e., vehicle v has SOC $k \cdot \Delta E / C_v$ at the beginning of time step t.

This allows to replace the non-linear Constraints (5.15) in formulation (5.14–5.20) by the following in order to obtain a MILP problem, which we refer to as EVS-SOC-λ.

$$y_{v,t} = \sum_{k=y_{v,0}}^{y_v^{\mathrm{dep}}} k \cdot \lambda_{v,t,k} \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1 \tag{5.21}$$

$$\sum_{k=y_{v,0}}^{y_v^{\mathrm{dep}}} \lambda_{v,t,k} = 1 \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1 \tag{5.22}$$

$$x_{v,t} \leq \sum_{k=y_{v,0}}^{y_v^{\mathrm{dep}}} \left\lfloor \frac{1}{\Delta E} \cdot E_v^{\max}\left(k \cdot \frac{\Delta E}{C_v} \right) \right\rfloor \cdot \lambda_{v,t,k} \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1 \tag{5.23}$$

$$\lambda_{v,t,k} \in \{0,1\} \qquad v \in V,\ t = 0, \dots, t_v^{\mathrm{dep}} - 1,\ k = y_{v,0}, \dots, y_v^{\mathrm{dep}} \tag{5.24}$$

Equations (5.21) link variables $y_{v,t}$ with the binary variables $\lambda_{v,t,k}$, and Equations (5.22) ensure that for each vehicle exactly one $\lambda_{v,t,k}$ variable is one at each time step. Inequalities (5.23) actually limit the maximum energy charged in dependence of the current SOC level.

5.3.1 Network Flow Approach

With PCP-E, a second MILP formulation is proposed in [HPL17] that uses a time-expanded network point-of-view. Again, however, the charging power is restricted to be either zero or the maximum allowed value. To adopt the basic idea from this approach, we again discretize the energy values and SOC as above.

We now consider the following layered flow network for each vehicle $v \in V$. The nodes are partitioned into layers $N_v = N_{v,0} \cup \dots \cup N_{v,t_v^{\text{dep}}}$, where $N_{v,0} = \{u_{v,0,y_{v,0}}\}$, $N_{v,t_v^{\text{dep}}} = \{u_{v,t_v^{\text{dep}},y_v^{\text{dep}}}\}$, and $N_{v,t} = \{u_{v,t,k} \mid k = y_{v,0}, \dots, y_v^{\text{dep}}\}$ for $t = 0, \dots, t_v^{\text{dep}} - 1$.

A flow through some node $u_{v,t,k}$ represents the case that vehicle v has SOC $k \cdot \Delta E / C_v$ at time t. Therefore, arcs $(u_{v,t,k}, u_{v,t+1,k'})$ exist for all feasible transitions from one time step to the next, i.e., for nodes $u_{v,t,k} \in N_{v,t}$ and $u_{v,t+1,k'} \in N_{v,t+1}$ if $k' \geq k \wedge \Delta E \cdot (k' - k) \leq E_v^{\max}(k \cdot \Delta E / C_v)$, for $0, \dots, t_v^{\text{dep}} - 1$. Let $A_{v,t}$ be the set of all these arcs connecting nodes from $N_{v,t}$ with nodes from $N_{v,t+1}$, $t = 0, \dots, t^{\text{dep}} - 1$. We associate flow variables $f_{v,t,k,k'} \in \{0,1\}$ with the respective arcs.

Using this network and the flow variables, we can reformulate (5.14–5.20) as the following MILP formulation, which we refer to as EVS-SOC-NET.

$$\min \quad \Delta E \cdot \sum_{v \in V} \sum_{t=0}^{t_v^{\text{dep}}-1} c_t \cdot \sum_{(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \tag{5.25}$$

$$\sum_{v \in V \mid 0 \leq t < t_v^{\text{dep}}} \sum_{(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \leq \left\lfloor \frac{\Delta t \cdot P^{\text{gridmax}}}{\Delta E} \cdot \right\rfloor \qquad t \in T \tag{5.26}$$

$$\sum_{(u_{v,t-1,k'}, u_{v,t,k}) \in A_{v,t-1}} f_{v,t-1,k',k} = \sum_{(u_{v,t,k}, u_{v,t+1,k''}) \in A_{v,t}} f_{v,t,k,k''} \qquad v \in V,\ k = y_{v,0}, \dots, y_v^{\text{dep}},\ t = 1, \dots, t_v^{\text{dep}} - 1 \tag{5.27}$$

$$\sum_{(u_{v,0,y_{v,0}}, u_{v,1,k}) \in A_{v,0}} f_{v,0,y_{v,0},k} = 1 \qquad v \in V \tag{5.28}$$

$$\sum_{(u_{v,t_v^{\text{dep}}-1,k}, u_{v,t_v^{\text{dep}},y_v^{\text{dep}}}) \in A_{v,t_v^{\text{dep}}-1}} f_{v,t_v^{\text{dep}}-1,k,y_v^{\text{dep}}} = 1 \qquad v \in V \tag{5.29}$$

$$f_{v,t,k,k'} \in \{0,1\} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1,\ (u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t} \tag{5.30}$$

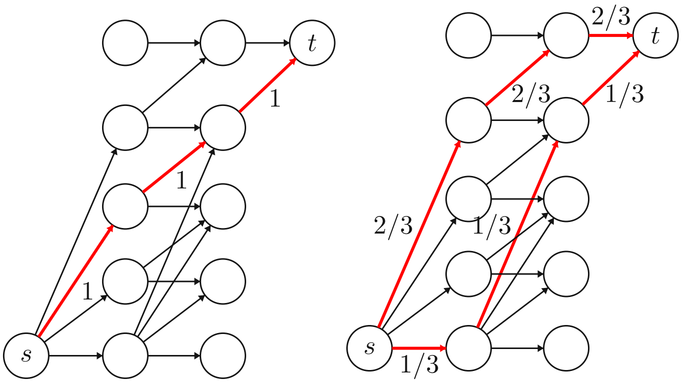
The objective function (5.25) minimizes the sum of the costs for the total consumed energy over all time steps. Inequalities (5.26) limit the total consumed energy units at each time step to the grids energy capacities $\lfloor \Delta t \cdot P^{\text{gridmax}} / \Delta E \rfloor$. Rounding is applied to strengthen the formulation. Equalities (5.27–5.29) are flow conservation constraints, which ensure that the flow traverses on a path from node $u_{v,0,y_{v,0}}$ to node $u_{v,t_v^{\text{dep}},y_v^{\text{dep}}}$

through the network. The domains of the integral flow variables are finally specified in (5.30).

Integrality of Flow Variables. Note that the flow variables must indeed be integral here, since in an LP relaxation they may assume fractional values due to Constraint (5.26) in combination with different costs for different time slots, even for just a single vehicle.

Additionally, keeping the flow variables fractional allows the charging energy to exceed $E_v^{\max}$, which will be explained with help of the following example. Assume a flow network of a single vehicle as shown in Figure 5.1 and let $\Delta t = 1$ hour, $\Delta E = 1$ kWh, $P^{\text{gridmax}} = 2$ kW, $C_1 = 4$ kWh, $s_{1,0} = 0$ and $s_1^{\text{dep}} = 1$. Furthermore, assume $E_1^{\max}(0/4) = 3$ kWh, $E_1^{\max}(1/4) = 1$ kWh, $E_1^{\max}(2/4) = 1$ kWh, $E_1^{\max}(3/4) = 1$ kWh and $E_1^{\max}(4/4) = 0$ kWh.

In Figure 5.1a, we can see the only feasible integral flow in the network that does not exceed the grid Constraints (5.26). However, if continuous flows are allowed, the flow might split on multiple paths while still respecting Constraints (5.26). Observe that in the solution indicated in Figure 5.1b, the vehicle's SOC is 0.5 at time step 1, however it is charged with $1/3 \cdot 3 + 2/3 \cdot 1$ at the same time step, exceeding $E_1^{\max}(0.5)$. This exemplary instance should clarify why the network flow variables $f_{v,t,k,k'}$ must be kept integral in EVS-SOC-NET.



(a) Integral Flow (b) Continuous Flow

Figure 5.1: Different solutions for integral and continuous flow variables. Used arcs are indicated in red and annotated with its corresponding flow value. We abbreviate $s = u_{v,0,y_{v,0}}$, $t = u_{v,t_v^{\text{dep}},y_v^{\text{dep}}}$.

Decomposition of EVS-SOC-NET. If Constraint (5.26) is omitted, EVS-SOC-NET decomposes into n subproblems, which can be solved independently. More concretely, each individual subproblem can be represented as minimum-cost flow problem, in which all arcs have flow capacity 1 and the costs for an arc $(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}$ are set to $(k' - k) \cdot \Delta E \cdot c_t$. When solving the minimum-cost flow problem, the integrality theorem guarantees integral flow due to integral flow demand and arc capacities [1]. Therefore, the obtained flow yields a feasible charging schedule for a single vehicle, which can be determined efficiently by linear programming for example.

Comparing LP relaxations. Let P_λ and P_{NET} be the polytopes corresponding to the LP relaxations of the EVS-SOC-λ and EVS-SOC-NET formulations. As we have seen before, there exists an instance and a solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in P_\lambda$ but its corresponding solution $\boldsymbol{f} \notin P_{\text{NET}}$, i.e., $P_\lambda \not\subset P_{\text{NET}}$.

For example, consider an instance with $n = 1$, $\Delta t = 1$ hour, $\Delta E = 1$ kWh, $P^{\text{gridmax}} = 10$ kW, $c_t = (2, 1)$ cent/kWh, $t_1^{\text{dep}} = 2$, $s_{1,0} = 0$, $s_1^{\text{dep}} = 1$, $C_1 = 2$ kWh and $E_1^{\max}(s) = 2s + 1$ kWh. An optimal solution for this instance maximizes the charged energy in time step 1 due to its cheaper costs of 1 cent/kWh. The relaxed EVS-SOC-λ formulation finds an optimal solution with $x_{1,0} = 0.5$, $x_{1,1} = 1.5$, $\lambda_{1,0,0} = 1$, $\lambda_{1,1,0} = 0.75$, $\lambda_{1,1,2} = 0.25$ and $\lambda_{v,t,k} = 0$ for all remaining λ variables. Clearly, this solution is feasible regarding Constraints (5.23) since

$$x_{1,0} \leq \lfloor E_1^{\max}(0) \rfloor \cdot \lambda_{1,0,0} \tag{5.31}$$

$$x_{1,1} \leq \lfloor E_1^{\max}(0) \rfloor \cdot \lambda_{1,1,0} + \lfloor E_1^{\max}(1) \rfloor \cdot \lambda_{1,1,2} \tag{5.32}$$

Considering the same instance with the LP relaxation of EVS-SOC-NET, we observe that there is only one path from the start to the end node in the flow network, which is $(u_{1,0,0}, u_{1,1,1}, u_{1,2,2})$. Even though $E_1^{\max}(0.5) = 2$ kWh, the charged energy is limited to 1 kWh at SOC 0.5 due to the structure of the network. Therefore there is only one solution in which we charge 1 kWh at each time step $t = 0, 1$.

Polytope P_{NET} is contained in polytope P_λ. In the following, we show $P_{\text{NET}} \subseteq P_\lambda$. Thereunto we consider an arbitrary instance and show for an arbitrary feasible solution $\boldsymbol{f} \in P_{\text{NET}}$ that there exists a corresponding feasible solution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \in P_\lambda$. We define how $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ can be obtained from $\boldsymbol{f}$.

$$x_{v,t} = \sum_{(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1 \tag{5.33}$$

$$y_{v,t} = \sum_{(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}} k \cdot f_{v,t,k,k'} \qquad v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1 \tag{5.34}$$

$$y_{v,t_v^{\text{dep}}} = \sum_{(u_{v,t_v^{\text{dep}}-1,k}, u_{v,t_v^{\text{dep}},k'}) \in A_{v,t_v^{\text{dep}}-1}} k' \cdot f_{v,t_v^{\text{dep}}-1,k,k'} \qquad v \in V \tag{5.35}$$

$$\lambda_{v,t,k} = \sum_{(u_{v,t,k}, u_{v,t+1,k'}) \in A_{v,t}} f_{v,t,k,k'} \qquad \begin{array}{r} v \in V,\ t = 0, \dots, t_v^{\text{dep}} - 1, \\ k = y_{v,0}, \dots, y_v^{\text{dep}} \end{array} \tag{5.36}$$

[1] https://www2.cs.duke.edu/courses/fall12/compsci590.1/network_flow.pdf

Now we show that $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda})$ satisfies Constraints (5.16–5.24). We transform each constraint of EVS-SOC-λ into a constraint of EVS-SOC-NET by applying the Substitutions (5.33–5.36).

- Inequalities (5.16): Satisfied by applying Substitutions (5.33) to Constraints (5.26).
- Equalities (5.17): Applying Substitution (5.35) yields equation

  $$\sum_{(u_{v,t_v^{\mathrm{dep}}-1,k},\, u_{v,t_v^{\mathrm{dep}},k'}) \in A_{v,t_v^{\mathrm{dep}}-1}} k' \cdot f_{v,t_v^{\mathrm{dep}}-1,k,k'} = y_v^{\mathrm{dep}} \qquad v \in V \tag{5.37}$$

  which is satisfied by the structure of the flow network ($k' = y_v^{\mathrm{dep}}$) and Constraints (5.29).

- Equalities (5.18): We start with showing the equality for $t = 1, \ldots, t_v^{\mathrm{dep}} - 1$. Shifting the index t yields

  $$y_{v,t} = y_{v,t-1} + x_{v,t-1} \qquad v \in V,\ t = 1, \ldots, t_v^{\mathrm{dep}} - 1 \tag{5.38}$$

  $$y_{v,t+1} = y_{v,t} + x_{v,t} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 2 \tag{5.39}$$

  Applying Substitutions (5.33) and (5.34) yields

  $$\begin{aligned} &\sum_{(u_{v,t+1,k},\, u_{v,t+2,k'}) \in A_{v,t+1}} k \cdot f_{v,t+1,k,k'} = \\ &\sum_{(u_{v,t,k},\, u_{v,t+1,k'}) \in A_{v,t}} k \cdot f_{v,t,k,k'} + \\ &\sum_{(u_{v,t,k},\, u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \end{aligned} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 2 \tag{5.40}$$

  and can be simplified to

  $$\begin{aligned} &\sum_{(u_{v,t+1,k},\, u_{v,t+2,k'}) \in A_{v,t+1}} k \cdot f_{v,t+1,k,k'} = \\ &\sum_{(u_{v,t,k},\, u_{v,t+1,k'}) \in A_{v,t}} k' \cdot f_{v,t,k,k'} \end{aligned} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 2 \tag{5.41}$$

  The latter inequality is satisfied by flow conservation Constraints (5.27). Analogously, Equality (5.18) can be shown for time step t_v^{dep}.

- Inequalities (5.19):

  $$0 \leq x_{v,t} \leq x_v^{\max} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 1 \tag{5.42}$$

  $$0 \leq \sum_{(u_{v,t,k},\, u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \leq x_v^{\max} \qquad v \in V,\ t = 0, \ldots, t_v^{\mathrm{dep}} - 1 \tag{5.43}$$

with $x_v^{\max} = \max_{k=y_{v,0},\ldots,y_v^{\text{dep}}} \lfloor E_v^{\max}(k \cdot \Delta E/C_v)/\Delta E \rfloor$.

Clearly, it holds that $0 \leq \sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'}$ for $v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1$, since $k' \geq k$ and flow variables are non-negative. Furthermore $\sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} (k'-k) \cdot f_{v,t,k,k'} \leq x_v^{\max}$, since $(k'-k) \leq \lfloor E_v^{\max}(k \cdot \Delta E/C_v)/\Delta E \rfloor$ and $\sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} f_{v,t,k,k'} = 1$ for $v \in V, t = 0,\ldots,t_v^{\text{dep}} - 1$.

- Inequalities (5.20): We start with showing the equality for $t = 0,\ldots,t_v^{\text{dep}} - 1$.

$$y_{v,0} \leq y_{v,t} \leq y_v^{\text{dep}} \qquad v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1 \tag{5.44}$$

$$y_{v,0} \leq \sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} k \cdot f_{v,t,k,k'} \leq y_v^{\text{dep}} \qquad v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1 \tag{5.45}$$

Due to the structure of the network it holds that $y_{v,0} \leq k \leq y_v^{\text{dep}}$. Together with $\sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} f_{v,t,k,k'} = 1$ for $v \in V, t = 0,\ldots,t_v^{\text{dep}} - 1$, Inequalities (5.45) are satisfied. We also consider Inequalities (5.20) for time step t_v^{dep}:

$$y_{v,0} \leq y_{v,t_v^{\text{dep}}} \leq y_v^{\text{dep}} \qquad v \in V \tag{5.46}$$

$$y_{v,0} \leq \sum_{(u_{v,t_v^{\text{dep}}-1,k},u_{v,t_v^{\text{dep}},k'}) \in A_{v,t_v^{\text{dep}}-1}} k' \cdot f_{v,t_v^{\text{dep}}-1,k,k'} \leq y_v^{\text{dep}} \qquad v \in V \tag{5.47}$$

Similar to before, we argue $k' = y_v^{\text{dep}}$ and $\sum_{(u_{v,t_v^{\text{dep}}-1,k},u_{v,t_v^{\text{dep}},k'}) \in A_{v,t}} f_{v,t_v^{\text{dep}}-1,k,k'} = 1$ for $v \in V$. Hence Inequalities (5.47) and further Inequalities (5.20) are satisfied.

- Equalities (5.21): Satisfied by applying Substitutions (5.34) and (5.36) to Equalities (5.21).

- Equalities (5.22): Applying Substitution (5.36) yields

$$\sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} f_{v,t,k,k'} = 1 \qquad v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1 \tag{5.48}$$

By the network flow conservation constraints, these equalities are satisfied.

- Inequalities (5.23):

$$x_{v,t} \leq \sum_{k=y_{v,0}}^{y_v^{\text{dep}}} \left\lfloor \frac{1}{\Delta E} \cdot E_v^{\max}\left(k \cdot \frac{\Delta E}{C_v}\right) \right\rfloor \cdot \lambda_{v,t,k} \qquad v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1 \tag{5.49}$$

$$\sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} (k' - k) \cdot f_{v,t,k,k'} \leq \sum_{k=y_{v,0}}^{y_v^{\text{dep}}} \left\lfloor \frac{1}{\Delta E} \cdot E_v^{\max}\left(k \cdot \frac{\Delta E}{C_v}\right) \right\rfloor \cdot \sum_{(u_{v,t,k},u_{v,t+1,k'}) \in A_{v,t}} f_{v,t,k,k'} \qquad v \in V,\ t = 0,\ldots,t_v^{\text{dep}} - 1 \tag{5.50}$$

The latter inequalities are satisfied since $(k' - k) \leq \left\lfloor \frac{1}{\Delta E} \cdot E_v^{\max}\left(k \cdot \frac{\Delta E}{C_v}\right) \right\rfloor$ holds by construction of the flow network.

- Inequalities (5.24): Satisfied when applying Substitution (5.36) to Inequalities (5.30).

Overall, we conclude that the LP relaxation of EVS-SOC-NET is stronger than the LP relaxation of EVS-SOC-λ. However this comes at a price of introducing a larger number of variables and constraints to the EVS-SOC-NET model.

CHAPTER 6

Benchmark Instances

Due to the lack of pure real-world problem instances we randomly generate benchmark instances and use real-world data as far as possible. We first consider individual EVS-SOC instances that represent snapshot scenarios at certain times with a specific number of vehicles that are assumed to have arrived at the charging station following a homogenous Poisson process. Afterwards, in Chapter 6.2, we will consider whole model based predictive control scenarios with a rolling horizon.

6.1 Individual EVS-SOC Instances

We distinguish between three types of problem parameters, depending on whether the parameter is set by the user, randomly generated, or based on real-world data. To the input data set by the user, we count the number of EVs n, the length of a time interval Δt, as well as the grid's power capacity P^{gridmax}. We generate 30 instances for each combination of $n \in \{10, 20, 50, 100\}$, $\Delta t \in \{1, 5, 10\}$ minutes, and $P^{\text{gridmax}} \in \{10n, 25n, 40n\}$.

We consider eight different types of EVs shown in Table 6.1. The EV's battery capacities were taken from the EV Database `https://www.ev-database.de`. The respective maximum power functions $P_v^{\max}$ were manually extracted from plots found on the website of a Dutch EV charging station operator `https://fastnedcharging.com`. More specifically, 25 up to 70 points of a plot were manually determined in dependence of notable changes of the gradient, and linear interpolation was applied in between. All extracted $P_v^{\max}$ functions are shown in Figure 6.1. Observe that the maximum power function's available domain of definition $[s_v^{\min}, s_v^{\max}]$ varies among the EVs. If a vehicle type supports speed charging, the respective most powerful charging curve is used.

Since the $P_v^{\max}$ data extracted from the original plots is quite fine-grained, we additionally derive simplified piecewise linear approximations with five and ten linear pieces, respectively. For this task, we utilized the Python package `pwlf` [JV19], which deter-

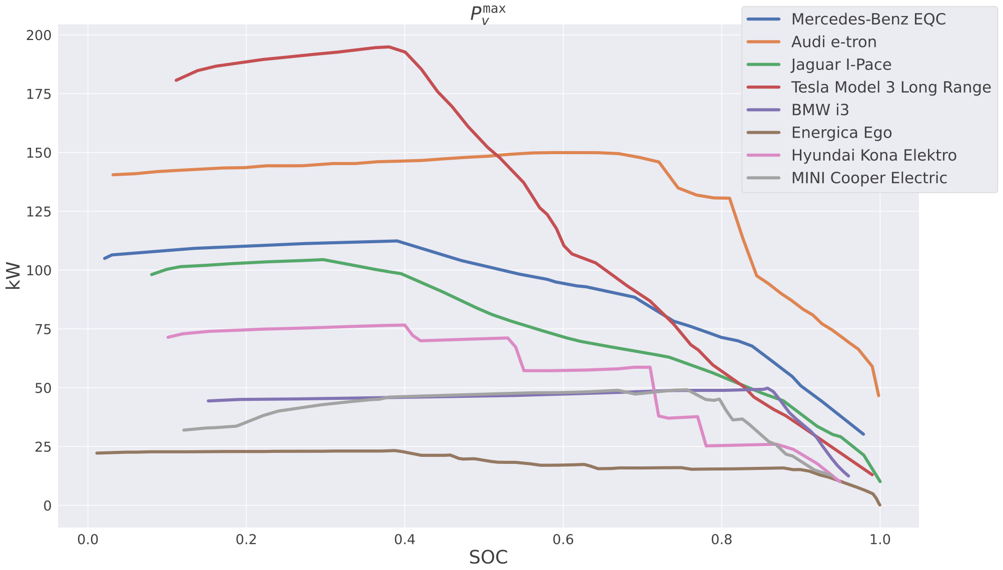


Figure 6.1: Maximum charging power functions $P_v^{\max}$ for all considered vehicle types.

mines approximately optimal breakpoints automatically. A comparison between the original $P_v^{\max}$ and its piecewise approximations is shown in Figure 6.2. Observe that the approximation of the original $P_v^{\max}$ function with 10 segments is quite good for most vehicles.

Remember that the EVS-SOC-LIN formulation only works on concave maximum energy functions. Whenever we benchmark this formulation, we use maximum charging power functions that are derived from the original ones by determining the convex hull of the set of points $\{(s, P_v^{\max}(s)) \mid s \in [s_v^{\min}, s_v^{\max}]\} \cup \{(s_v^{\min}, 0), (s_v^{\max}, 0)\}$. From the concavity

EV Name	C_v (kWh)	$s_v^{\min}$	$s_v^{\max}$	$\#P_v^{\max}$-lin. pieces
Energica Ego	21.5	1.1	99.9	53
MINI Cooper Electric	32.6	12.1	93.8	34
BMW i3	42.2	15.1	96.0	26
Hyundai Kona Elektro	67.5	10.1	94.9	28
Tesla Model 3 Long Range	82.0	11.1	99.0	35
Mercedes-Benz EQC	85.0	2.1	97.8	24
Jaguar I-Pace	90.0	8.0	100.0	29
Audi e-tron	95.0	3.1	99.8	44

Table 6.1: Used EV types with battery capacity C_v, $P_v^{\max}$ domain $[s_v^{\min}, s_v^{\max}]$ and the number of linear pieces of $P_v^{\max}$.

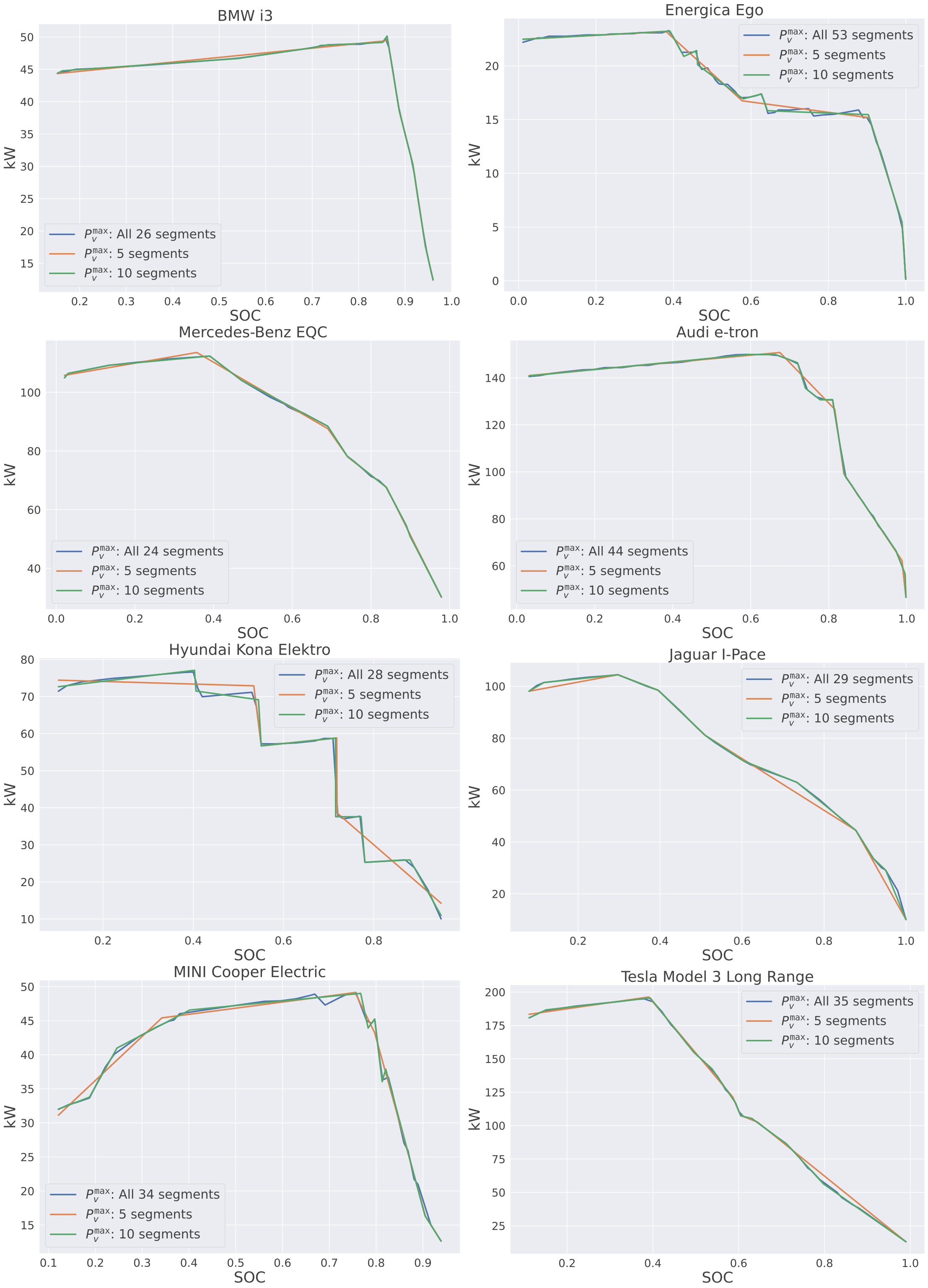


Figure 6.2: Comparison of $P_v^{\max}$ curves with different number of segments for each vehicle type.

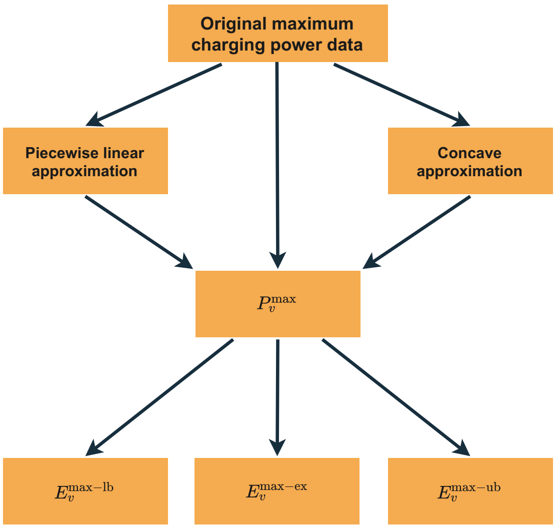


Figure 6.3: Schematic preprocessing flow chart of the charging data.

of $P_v^{\max}$ follows concavity of $E_v^{\max}$, as mentioned in Chapter 3. To give the reader a better understanding of how the charging curves relate to each other, we outline the preprocessing steps in Figure 6.3. With „original maximum charging power data“ we refer to the data taken from the charging station operator. Notice that this data is either approximated with a piecewise linear or concave function (convex hull). With $P_v^{\max}$ we refer to any, possibly preprocessed, maximum charging power function. Based on this curve, the maximum charging energy functions $E_v^{\text{max-lb}}$, $E_v^{\text{max-ex}}$ and $E_v^{\text{max-ub}}$ are deduced.

For each EV $v \in V$ in a benchmark instance, one of the above EV types is chosen uniformly at random. Moreover, we choose an availability duration at the charging station d_v^{avail} randomly according to a normal distribution with a mean value of 6 hours and a standard deviation of 1.5 hours.

Next, we select an arrival time t_v^{arr} uniformly at random from the interval $(-d_v^{\text{avail}}/\Delta t, 0)$ and obtain a respective departure time $t_v^{\text{dep}} = \lceil t_v^{\text{arr}} + d_v^{\text{avail}}/\Delta t \rceil$. Considering the available domains of definition of the maximum power functions, we generally assume that each vehicle shall be charged from a SOC of 20% at arrival to a SOC of 90% at departure. In our benchmark instances, we therefore choose the initial SOC proportional to the already bygone availability time, i.e., for all $v \in V$,

$$s_{v,0} = \frac{-t_v^{\text{arr}}}{d_v^{\text{avail}}/\Delta t} \cdot 0.73 + 0.2 \tag{6.1}$$

The departure SOC s_v^{dep} is set to 90% for all EVs.

The end of the time horizon is obtained from the last EV's departure time, i.e., $t_{\max} = \max_{v \in V} t_v^{\text{dep}}$. Electricity costs per unit of consumed energy c_t are independently chosen for each time step $t \in T$ uniformly at random from $[1.9, 3.5)$ cent/kWh.

6.2 Rolling Horizon Benchmark Scenarios

In addition to the individual benchmark instances, we consider rolling horizon simulations over whole days starting at time 0:00 and ending at 24:00. The time is again discretized into equally long time steps of $\Delta t \in \{5, 10\}$ minutes. Electricity costs per unit of consumed energy are chosen as in Chapter 6.1 and it is assumed that they are known in advance for the whole charging period. For the number of vehicles we use $n \in \{10, 20, 50, 100\}$. Again, we pick each vehicle type uniformly at random from the set of available vehicle types.

It is assumed that most vehicles arrive around two peak times at 6:00 and 14:00. For picking the arrival time t_v^{arr} for a vehicle $v \in V$, we therefore first randomly select with equal probability one of these two peak times and then sample t_v^{arr} from a normal distribution with the chosen peak time as mean value and a standard deviation of two hours. Times outside of the considered horizon of 24 hours are re-sampled.

The charging duration d_v^{avail} is chosen as in Chapter 6.1 and t_v^{dep} is derived correspondingly. Also, s_v^{dep} and P^{gridmax} are set as before. At time 0:00 we set $s_{v,0} = 0.2$ and with each rescheduling we determine $s_{v,0}$ based on the charging schedule of the previous iteration.

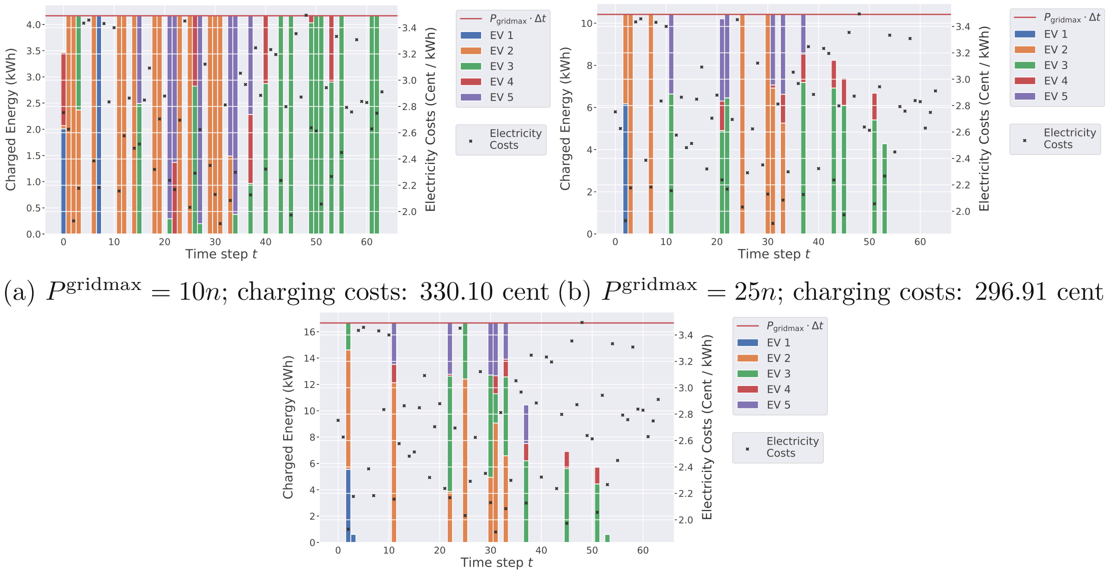
The schedule is (re-)optimized at time 0:00 and then every $\tau = 10$ minutes, always considering only EVs that are currently available at the charging station. The found charging schedule is then assumed to be applied for the next τ minutes until a new schedule is determined.

Thirty independent whole-day scenarios were constructed and are considered in the experimental evaluation.

Inspecting a single instance. In Figure 6.4, we visualize an optimal solution for a single individual instance with $n = 5$, $\Delta t = 5$ minutes using EVS-SOC-GLIN for all $P^{\text{gridmax}} \in \{10n, 25n, 40n\}$. As maximum energy function we chose $E_v^{\text{max-lb}}$ based on P_v^{max} with 5 piecewise linear segments. A single sub-figure represents an optimal charging schedule of a vehicle fleet. Bars specify the energy a vehicle is charged with at each time step. The corresponding scale is located on the left. The grid's maximum energy supply $P^{\text{gridmax}} \cdot \Delta t$ is indicated as horizontal line in each plot. Crosses reveal the electricity costs for each time step and the appropriate scale is located on the right.

It can be seen that for smaller P^{gridmax} values, there are more time steps in which the total charged energy reaches the grid's energy capacities $P^{\text{gridmax}} \cdot \Delta t$. For higher P^{gridmax} values, more vehicles are charged in parallel within a single time step and cheap electricity costs can be exploited more effectively.

We also visualize an optimal rolling horizon solution with $n = 10$, $\Delta t = 5$ minutes, $\tau = 10$ minutes using EVS-SOC-GLIN for $P^{\text{gridmax}} \in \{10n, 25n\}$ in Figure 6.5. The same E^{max} function as for the individual instance above was used. For this instance, the temporal availability of the vehicles is indicated in Figure 6.5c.

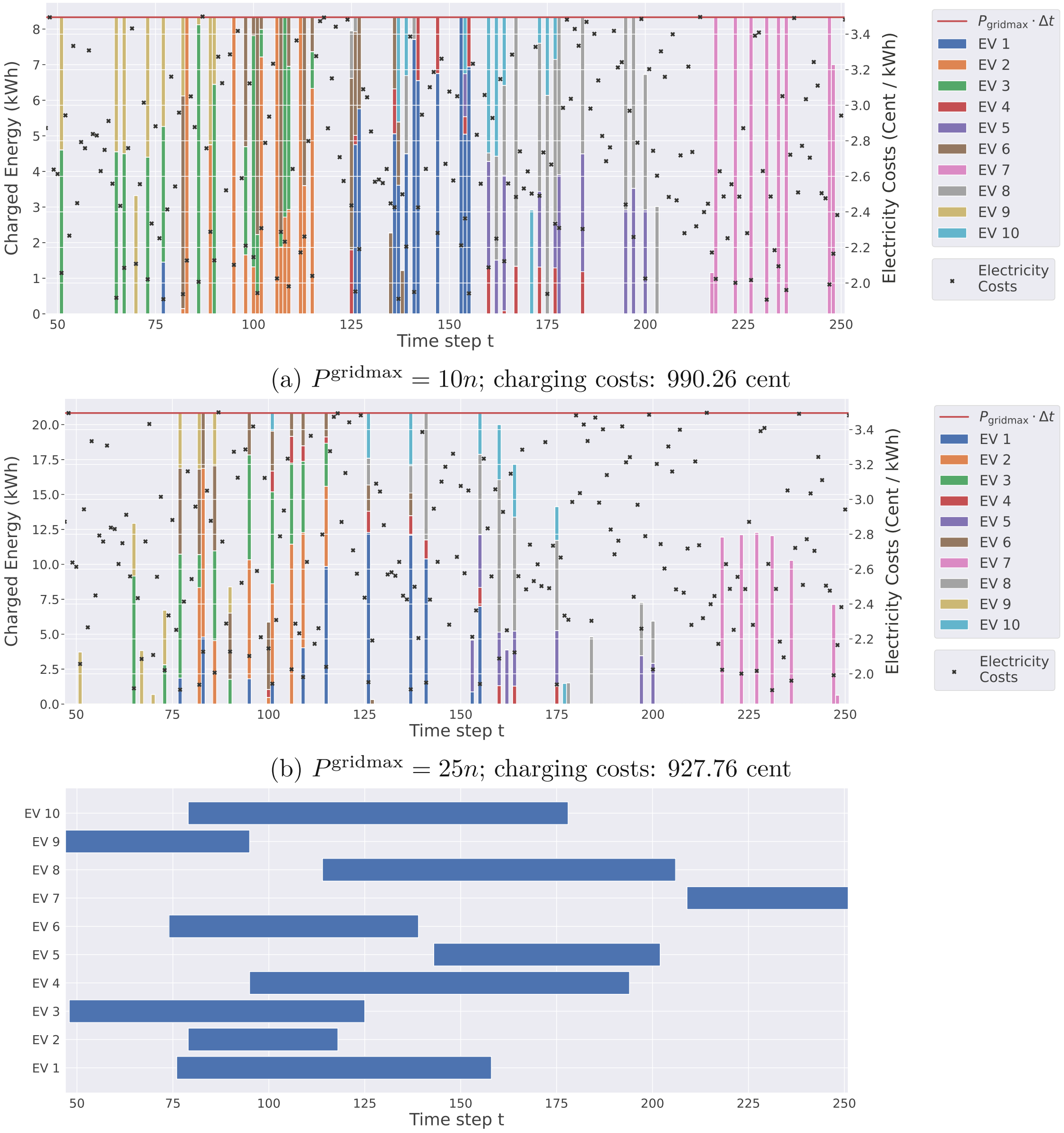


(a) $P^{\text{gridmax}} = 10n$; charging costs: 330.10 cent (b) $P^{\text{gridmax}} = 25n$; charging costs: 296.91 cent

(c) $P^{\text{gridmax}} = 40n$; charging costs: 290.42 cent

Figure 6.4: Optimal solution for an instance with $n = 5$, $\Delta t = 5$ minutes, $P^{\text{gridmax}} \in \{10n, 25n, 40n\}$ using EVS-SOC-GLIN.

Arrival times are distributed around both peak times 6:00 and 14:00 (time step 72 and 168), as can be seen in Figure 6.5c. EV 7 arrives late and has a relatively short period of stay. Again, a smaller P^{gridmax} value implies that more time steps have to be fully exploited, i.e., less charging breaks are planned.



(a) $P^{\text{gridmax}} = 10n$; charging costs: 990.26 cent

(b) $P^{\text{gridmax}} = 25n$; charging costs: 927.76 cent

(c) Temporal availability of each vehicle.

Figure 6.5: Optimal solution for a rolling horizon instance with $n = 10$, $\Delta t = 5$ minutes, $P^{\text{gridmax}} \in \{10n, 25n\}$ using EVS-SOC-GLIN.

CHAPTER 7

Experiments

All formulations were implemented in Julia 1.6.0-rc1[1] using the the optimization modeling package JuMP v0.21.5 and Gurobi 9.1.0[2] as LP/MILP solver. Gurobi was configured to run in single-threaded mode with a time limit of 30 minutes per instance. All remaining Gurobi parameters were kept at their default values. The experiments were conducted on an Intel Xeon E5-2640 v4 with 2.40GHz and 16GB memory limit, running Ubuntu 18.04.5 LTS. If not stated otherwise we report in the following mean or median results on the 30 problem instances per instance parameter combination $(n, \Delta t, P^{\text{gridmax}}, E_v^{\max})$.

7.1 Runtimes

We first focus on the runtimes of the individual approaches and how many instances could be solved to proven optimality in the given time limit of 30 minutes.

EVS-SOC-LIN. We compare two variants of EVS-SOC-LIN. In one of them, all maximum charging energy constraints (5.1) are statically added to the LP formulation, whereas in the other variant these constraints are dynamically separated as cuts via the cutting plane approach as described in Chapter 5.1. As maximum energy function we use $E_v^{\text{max-ex}}$ based on the convex $P_v^{\max}$ functions (without any further piecewise linear approximation). We set $P^{\text{gridmax}} = 25n$. The results of the comparison are reported in Table 7.1 and are illustrated in Figure 7.1. Column n_{seg} denotes the total number of piecewise linear segments of the $E_v^{\max}$ functions over all vehicles. All reported instances were solved to optimality in all runs. For the cutting plane approach, the total number of added cuts per instance is denoted by n_{cuts}.

[1] https://julialang.org

[2] https://www.gurobi.com

n	Δt (min)	n_{seg}	Runtime (s)				n_{cuts}	
			Static		Cutting Plane		Cutting Plane	
		Mean	Median	StdDev	Median	StdDev	Mean	StdDev
5	1	901	1.19	1.02	1.31	0.38	12800	5986
5	5	901	0.23	0.11	0.90	0.25	1102	542
5	10	901	0.08	0.07	0.97	0.26	322	205
10	1	1802	4.98	3.27	1.65	0.52	25271	9541
10	5	1802	0.59	0.22	1.06	0.24	2341	950
10	10	1802	0.22	0.10	1.01	0.20	757	387
20	1	3605	14.33	8.48	3.29	0.83	60778	18725
20	5	3605	1.21	0.45	1.16	0.27	5117	1547
20	10	3605	0.68	0.20	1.07	0.21	1585	516
50	1	9041	70.69	31.89	9.11	2.66	175979	28195
50	5	9041	4.17	1.58	1.57	0.33	13737	2329
50	10	9041	1.57	0.54	1.15	0.21	3989	858
100	1	18086	280.22	100.87	25.45	9.66	390873	44162
100	5	18086	13.11	4.73	2.11	0.51	27920	3515
100	10	18086	3.80	1.35	1.32	0.34	8126	1419

Table 7.1: EVS-SOC-LIN runtime comparison for concave maximum power functions and $P^{\text{gridmax}} = 25n$: Solving the static MILP problem versus the cutting plane approach.

Observe that for a fixed Δt the cutting plane approach shows its performance advantages with growing n. The improvement is also noticeable if we fix n and consider decreasing Δt values. Similarly, for a fixed Δt the number of cuts increases with larger n values, whereas for a fixed n the number of cuts increases with smaller Δt values. The results indicate that the cutting plane technique shows performance benefits when a larger number of cuts has been separated, i.e., the maximum charging power condition was not easily fulfilled. Overall, it can be said that the cutting plane variant outperforms the static model on larger instances and when n_{seg} is larger. We additionally conducted the experiments for $P^{\text{gridmax}} = 10n$ and $40n$ and observed the same trends.

EVS-SOC-GLIN. Similar to before, we compare two variants of EVS-SOC-GLIN for the general non-concave maximum charging power functions. In the first variant we directly solve the static MILP problem in which all linking constraints (5.8–5.10) are included from the beginning, whereas the second approach is the branch-and-cut variant (B&C) in which these linking constraints are dynamically added as needed, cf. Chapter 5.2.1. As maximum energy function we use $E_v^{\text{max-ex}}$ and $E_v^{\text{max-lb}}$, both based on the original full resolution $P_v^{\max}$ functions. For $P^{\text{gridmax}} \in \{10n, 25n, 40n\}$ we report detailed results in Tables 7.2, 7.3, and 7.4, respectively. Columns, $n_{\text{seg}}[E_v^{\max}]$ denote the total number of piecewise linear segments functions $E_v^{\max}$ consist of, summed over all n vehicles of an instance. Again, n_{cuts} denotes for the total number of cuts added within B&C. The last columns indicate the finally remaining optimality gaps between lower and

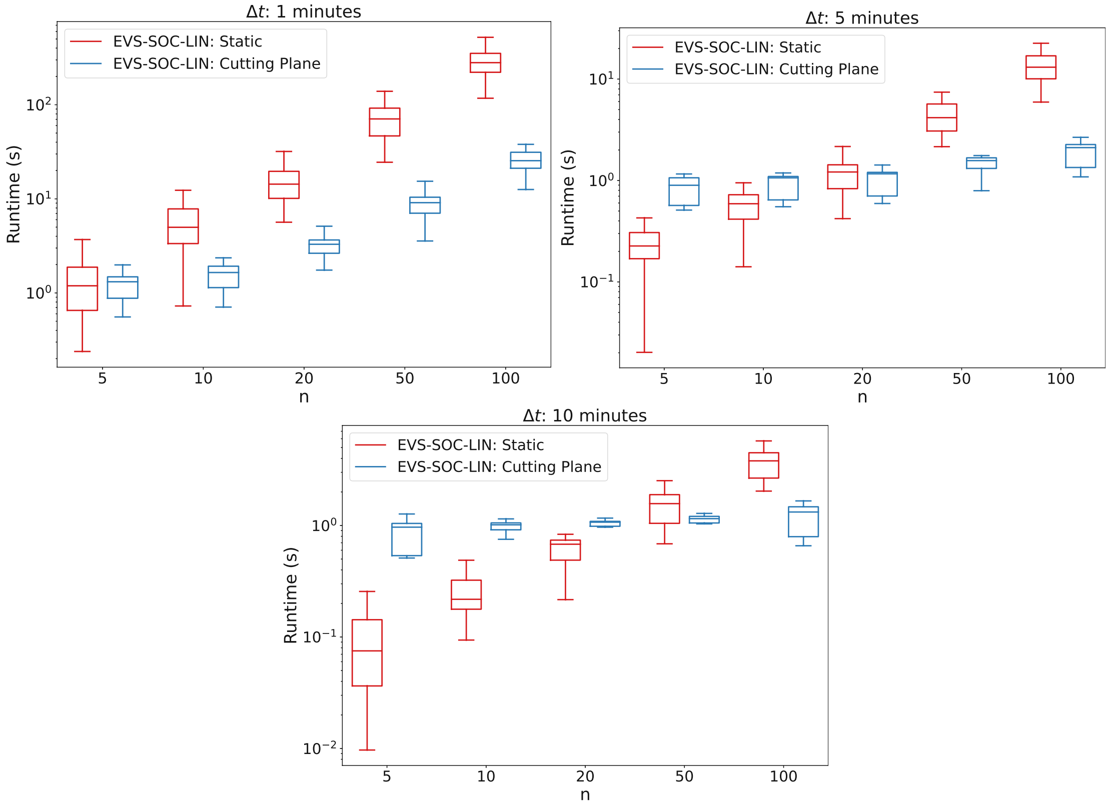


Figure 7.1: EVS-SOC-LIN runtime comparison for directly solving the LP problem versus the cutting plane approach, corresponding to results of Table 7.1.

upper bounds. Only relative gaps of instances with a feasible solution are considered. For parameter combinations without gaps (marked with “-”), no feasible solution has been found for any instance. For parameter combinations where no runtime is reported, all corresponding runs terminated due to an out-of-memory error.

Opposed to EVS-SOC-LIN, not all instances could be solved to optimality by the EVS-SOC-GLIN variants. Hence, in Tables 7.2, 7.3 and 7.4 we also report the number of instances n_{solved} where a feasible solution was found, the number of instances n_{opt} solved to optimality, as well as the gaps for each group.

Considering the results with $P^{\text{gridmax}} = 10n$, one can notice that the B&C approach shows performance benefits for a small number $n \leq 20$ of vehicles. It is better than the static variant in terms of solved instances and median runtime, however solves less instances to optimality. On larger instances, the static variant seems to outperform the B&C variant in terms of optimally solved instances. For the results with $P^{\text{gridmax}} = 25n$, the runtime performance benefit of B&C for small n values is still noticeable, however it is not as strong as for $P^{\text{gridmax}} = 10n$. Lastly, for $P^{\text{gridmax}} = 40n$ the static variant solves more instances to optimality and has better runtime on almost all parameter

n	Δt (min)	$n_{\text{seg}}[E_v^{\max}]$	n_{opt}		n_{feas}		Runtime (s)				n_{cuts}		Relative Gap (%)	
							Median		StdDev		Median	StdDev	Median	
		Mean	Static	B&C	Static	B&C	Static	B&C	Static	B&C	B&C		Static	B&C
							$E_v^{\text{max-lb}}$							
5	1	155	24	24	30	30	391.75	43.39	662.53	720.28	1038	1944	0.01	0.01
5	5	139	30	30	30	30	6.58	1.43	17.30	64.96	144	307	0.00	0.01
5	10	119	30	30	30	30	1.67	0.83	6.66	1.77	56	120	0.00	0.00
10	1	311	5	12	21	29	1800.00	1800.00	389.82	764.87	4068	2726	0.03	0.03
10	5	279	29	27	30	30	79.94	8.84	353.48	544.47	498	581	0.01	0.01
10	10	242	30	30	30	30	7.04	2.06	35.58	4.31	194	167	0.00	0.01
20	1	612	0	1	2	11	1800.00	1800.00	0.00	181.68	8974	1820	0.08	0.19
20	5	553	23	19	30	30	500.49	684.63	640.95	781.48	1846	1081	0.01	0.01
20	10	475	30	29	30	30	40.18	13.35	71.71	425.71	505	274	0.01	0.01
50	1	1544	0	0	0	0	1800.00	1800.00	0.00	0.00	15910	2518	-	-
50	5	1393	2	2	26	30	1800.00	1800.00	174.76	351.60	6106	1249	0.05	0.05
50	10	1192	29	18	30	30	307.62	827.59	458.49	779.32	1930	594	0.01	0.01
100	1	3095	0	0	0	0	1800.00	1800.00	0.00	0.00	11886	3940	-	-
100	5	2796	0	0	9	9	1800.00	1800.00	0.00	0.00	9961	1319	0.08	0.12
100	10	2399	11	4	30	30	1800.00	1800.00	418.13	452.22	4434	861	0.01	0.03
							$E_v^{\text{max-ex}}$							
5	1	901	3	12	8	27	1800.00	1800.00	414.46	834.31	5304	6069	0.03	0.01
5	5	901	30	26	30	30	143.42	9.59	312.39	628.62	820	1553	0.00	0.00
5	10	901	30	30	30	30	34.53	2.60	106.85	48.02	319	623	0.00	0.00
10	1	1802	0	3	1	21	1800.00	1800.00	0.00	401.91	13982	7431	0.04	0.08
10	5	1802	13	16	29	30	1800.00	725.37	568.29	872.54	2858	2600	0.01	0.01
10	10	1802	30	28	30	30	201.32	10.29	327.78	451.48	680	845	0.00	0.01
20	1	3605	0	0	0	10	1800.00	1800.00	0.00	0.00	23449	6856	-	0.14
20	5	3605	2	6	14	30	1800.00	1800.00	85.05	629.36	6479	4009	0.07	0.05
20	10	3605	22	18	30	30	1038.91	116.59	569.76	862.58	1507	1708	0.01	0.01
50	1	9041	0	0	0	0	1800.00	1800.00	0.00	0.00	6856	4528	-	-
50	5	9041	0	1	0	23	1800.00	1800.00	0.00	308.85	15048	3971	-	0.11
50	10	9041	1	7	4	30	1800.00	1800.00	123.65	585.67	6160	3202	0.18	0.03
100	1	18078	0	0	0	0	-	1800.00	-	0.00	0	5698	-	-
100	5	18086	0	0	0	10	1800.00	1800.00	0.00	0.00	18944	5630	-	0.08
100	10	18086	0	2	0	25	1800.00	1800.00	0.00	393.05	10750	3536	-	0.06

Table 7.2: EVS-SOC-GLIN results for solving the static model versus B&C with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$ based on the original $P_v^{\max}$ functions and $P^{\text{gridmax}} = 10n$.

n	Δt (min)	$n_{\text{seg}}[E_v^{\max}]$	n_{opt}		n_{feas}		Runtime (s)				n_{cuts}		Relative Gap (%)	
							Median		StdDev		Median	StdDev	Median	
		Mean	Static	B&C	Static	B&C	Static	B&C	Static	B&C	B&C		Static	B&C
$E_v^{\text{max-lb}}$														
5	1	155	12	5	29	30	1800.00	1800.00	690.82	636.49	3184	3153	0.02	0.06
5	5	139	28	26	30	30	25.52	6.68	450.62	616.54	422	461	0.01	0.01
5	10	119	30	29	30	30	1.27	1.62	10.37	329.09	153	154	0.01	0.01
10	1	312	1	0	20	23	1800.00	1800.00	29.78	0.00	7298	2829	0.10	0.12
10	5	279	29	19	30	30	183.39	770.59	445.94	815.87	1132	771	0.01	0.01
10	10	242	30	29	30	30	17.87	11.88	163.75	437.28	452	223	0.01	0.01
20	1	612	0	0	4	3	1800.00	1800.00	0.00	0.00	11938	2372	0.26	0.28
20	5	553	14	6	30	30	1800.00	1800.00	608.45	637.22	2702	936	0.01	0.05
20	10	475	29	22	30	30	60.59	201.06	422.28	755.21	967	359	0.01	0.01
50	1	1544	0	0	0	0	1800.00	1800.00	0.00	0.00	22034	4220	-	-
50	5	1393	2	1	29	30	1800.00	1800.00	160.23	280.55	6997	1257	0.08	0.11
50	10	1192	20	5	30	30	902.21	1800.00	698.40	604.91	2575	653	0.01	0.03
100	1	3095	0	0	0	0	1800.00	1800.00	0.00	0.00	29193	6236	-	-
100	5	2796	0	0	14	7	1800.00	1800.00	0.00	0.00	11737	1494	0.12	0.18
100	10	2399	6	0	30	30	1800.00	1800.00	482.70	0.00	5340	1077	0.03	0.06
$E_v^{\text{max-ex}}$														
5	1	901	1	1	9	25	1800.00	1800.00	274.48	138.10	15258	9382	0.21	0.20
5	5	901	26	17	30	30	448.47	761.59	644.73	831.42	2153	2330	0.01	0.01
5	10	901	29	26	30	30	56.12	16.43	321.42	610.48	866	990	0.00	0.01
10	1	1802	0	0	1	18	1800.00	1800.00	0.00	0.00	23328	9977	0.23	0.33
10	5	1802	12	7	26	30	1800.00	1800.00	580.44	699.52	5220	3467	0.04	0.06
10	10	1802	29	22	30	30	204.26	233.60	417.05	757.12	2063	1389	0.01	0.01
20	1	3605	0	0	0	2	1800.00	1800.00	0.00	0.00	17970	9466	-	0.32
20	5	3605	1	1	15	29	1800.00	1800.00	113.34	318.65	10784	4058	0.08	0.12
20	10	3605	20	10	29	30	1097.26	1800.00	573.95	709.09	4647	2500	0.01	0.03
50	1	9041	0	0	0	0	1800.00	1800.00	0.00	0.00	23986	9245	-	-
50	5	9041	0	0	0	17	1800.00	1800.00	0.00	0.00	23708	5721	-	0.18
50	10	9041	0	3	16	28	1800.00	1800.00	0.00	439.63	12160	4186	0.04	0.08
100	1	18086	0	0	0	0	1800.00	1800.00	0.00	0.00	0	4697	-	-
100	5	18086	0	0	0	0	1800.00	1800.00	0.00	0.00	25754	8585	-	-
100	10	18086	0	0	0	19	1800.00	1800.00	0.00	0.00	19752	5121	-	0.09

Table 7.3: EVS-SOC-GLIN results for solving the static model versus B&C with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$ based on the original $P_v^{\max}$ functions and $P^{\text{gridmax}} = 25n$.

n	Δt (min)	$n_{\text{seg}}[E_v^{\max}]$	n_{opt}		n_{feas}		Runtime (s)				n_{cuts}		Relative Gap (%)	
							Median		StdDev		Median	StdDev	Median	
		Mean	Static	B&C	Static	B&C	Static	B&C	Static	B&C	B&C		Static	B&C
							$E_v^{\text{max-lb}}$							
5	1	155	11	2	29	29	1800.00	1800.00	542.57	57.30	4476	2923	0.04	0.15
5	5	139	28	24	30	30	31.04	55.93	513.40	715.78	619	492	0.01	0.01
5	10	119	30	29	30	30	2.49	4.05	53.44	371.17	247	153	0.01	0.01
10	1	311	0	0	20	20	1800.00	1800.00	0.00	0.00	8161	3130	0.21	0.17
10	5	279	21	8	30	30	301.14	1800.00	745.75	677.68	1410	676	0.01	0.03
10	10	242	28	26	30	30	27.80	36.06	450.92	660.00	456	201	0.01	0.01
20	1	612	0	0	2	1	1800.00	1800.00	0.00	0.00	13361	2440	0.27	0.48
20	5	553	5	0	30	30	1800.00	1800.00	365.20	0.00	2863	884	0.04	0.10
20	10	475	28	19	30	30	69.51	571.16	479.77	745.04	1078	327	0.01	0.01
50	1	1544	0	0	0	0	1800.00	1800.00	0.00	0.00	25908	3569	-	-
50	5	1393	0	0	28	28	1800.00	1800.00	0.00	0.00	7110	1096	0.12	0.21
50	10	1192	18	1	30	30	1097.80	1800.00	640.13	183.90	2748	520	0.01	0.05
100	1	3095	0	0	0	0	1800.00	1800.00	0.00	0.00	29066	6072	-	-
100	5	2796	0	0	7	2	1800.00	1800.00	0.00	0.00	11782	1239	0.22	0.21
100	10	2399	1	0	29	30	1800.00	1800.00	121.93	0.00	5650	808	0.06	0.10
							$E_v^{\text{max-ex}}$							
5	1	901	2	0	9	24	1800.00	1800.00	261.72	0.00	20190	9588	0.23	0.44
5	5	901	25	9	30	30	582.18	1800.00	651.87	643.80	3180	2231	0.01	0.07
5	10	901	30	23	30	30	80.12	34.07	160.32	753.56	1228	955	0.00	0.01
10	1	1802	0	0	1	13	1800.00	1800.00	0.00	0.00	24450	8643	0.49	0.77
10	5	1802	12	0	26	30	1800.00	1800.00	598.34	0.00	6026	3161	0.02	0.17
10	10	1802	29	17	30	30	245.17	1147.26	375.49	837.79	2161	1553	0.01	0.01
20	1	3605	0	0	0	0	1800.00	1800.00	0.00	0.00	17460	9716	-	-
20	5	3605	0	0	15	29	1800.00	1800.00	0.00	0.00	13276	3457	0.14	0.22
20	10	3605	19	3	29	30	1437.18	1800.00	550.74	447.72	5692	2190	0.01	0.08
50	1	9041	0	0	0	0	1800.00	1800.00	0.00	0.00	12253	7961	-	-
50	5	9041	0	0	0	11	1800.00	1800.00	0.00	0.00	27617	4805	-	0.21
50	10	9041	0	0	14	27	1800.00	1800.00	0.00	0.00	13538	2670	0.10	0.12
100	1	18083	0	0	0	0	-	1800.00	-	0.00	0	9122	-	-
100	5	18086	0	0	0	0	1800.00	1800.00	0.00	0.00	31692	13113	-	-
100	10	18086	0	0	0	11	1800.00	1800.00	0.00	0.00	23081	4035	-	0.14

Table 7.4: EVS-SOC-GLIN results for solving the static model versus B&C with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$ based on the original $P_v^{\max}$ functions and $P^{\text{gridmax}} = 40n$.

configurations.

A possible explanation for this observation seems to be that for $P^{\text{gridmax}} = 10n$ the charging energy of a vehicle v is rather limited by P^{gridmax} than by E_v^{max}. Initial solutions of B&C will then violate Constraints (5.5) less often, which implies spending less time for the separation of cuts. This presumption should be supported if we consider the number of added cuts. Fixing n and Δt, one can observe that with growing P^{gridmax} values clearly more cuts are added.

When comparing $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$ for any fixed P^{gridmax}, n and Δt value, $E_v^{\text{max-ex}}$ has more segments than $E_v^{\text{max-lb}}$ due to the nature of its computation. Also, for $E_v^{\text{max-lb}}$ smaller Δt values imply a higher number of $E_v^{\text{max-lb}}$ segments. For a fixed n, Δt the larger number of $E_v^{\text{max-ex}}$ segments comes with fewer (optimally) solved instances and higher runtimes for the static and B&C approach.

In order to see how both solution approaches to EVS-SOC-GLIN perform on instances with fewer piecewise linear segments in E_v^{max}, we conduct similar experiments using the approximations of P_v^{max} with five segments. There we consider $E_v^{\text{max-ub}}$ instead of $E_v^{\text{max-ex}}$, since the number of $E_v^{\text{max-ex}}$ segments does not depend on the number of P_v^{max} segments. We only conduct experiments for $P^{\text{gridmax}} = 25n$ to keep the results to a manageable size. The results can be seen in Table 7.5.

With only a few exceptions, the B&C approach almost always finds more feasible solutions than the static variant for each parameter group. However, if we compare the number of optimally solved instances, the static variant seems to be better for most groups. Also, if the static and the B&C variant find the same number of feasible solutions, the gap of the static variant is smaller or equal for most parameter combinations.

Due to the fewer number of segments in the P_v^{max} functions and consequently also simpler $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ functions, more instances could be solved to optimality and to more instances feasible solutions could generally be found, when comparing Tables 7.5 and 7.3. Moreover, the impact of fewer P_v^{max} segments is also observable when we consider the median runtimes and the number of added cuts. For almost all parameter combinations of n and Δt, fewer P_v^{max} segments lead to lower median runtimes and fewer cuts.

EVS-SOC-λ and EVS-SOC-NET. For the energy-discretized formulations EVS-SOC-λ and EVS-SOC-NET, we conduct experiments with $\Delta E \in \{5, 10\} \cdot \Delta t$, $P^{\text{gridmax}} = 25n$, and $E_v^{\text{max-lb}}$ based on the P_v^{max} with the full number of piecewise linear segments. We choose ΔE in dependence of Δt as all E_v^{max} functions scale with Δt. Selecting a too large ΔE value together with an E_v^{max} function, which is based on a small Δt value, could even lead to infeasible instances. This is another good reason to link these parameters.

Results of the experiment are reported in Table 7.6. Similarly as before, parameter combinations where no runtime is reported (marked with "-") reflect runs terminated due to an out-of-memory error. For parameter combinations without gaps, a feasible solution has not been found for any instance.

n	Δt (min)	$n_{\text{seg}}[E_v^{\max}]$	n_{opt}		n_{feas}		Runtime (s)				n_{cuts}		Relative Gap (%)	
							Median		StdDev		Median	StdDev	Median	
		Mean	Static	B&C	Static	B&C	Static	B&C	Static	B&C	B&C		Static	B&C
$E_v^{\text{max-lb}}$														
5	1	40	29	22	30	30	60.14	19.63	394.28	791.01	387	485	0.01	0.01
5	5	46	30	30	30	30	2.40	1.98	5.97	263.17	88	102	0.01	0.01
5	10	43	30	30	30	30	0.64	1.13	1.37	1.21	42	50	0.00	0.01
10	1	80	27	13	30	30	509.28	1800.00	582.27	830.23	1162	639	0.01	0.02
10	5	92	30	30	30	30	11.01	8.34	28.13	224.13	232	136	0.01	0.01
10	10	87	30	30	30	30	1.49	2.68	1.78	8.36	118	62	0.01	0.01
20	1	160	5	2	12	30	1800.00	1800.00	193.77	407.09	2488	722	0.03	0.06
20	5	185	30	25	30	30	54.58	61.09	199.06	659.96	516	192	0.01	0.01
20	10	174	30	30	30	30	5.03	7.45	13.02	37.35	217	79	0.01	0.01
50	1	398	0	0	0	12	1800.00	1800.00	0.00	0.00	5598	796	-	0.24
50	5	459	28	10	30	30	640.74	1800.00	516.17	754.54	1556	363	0.01	0.02
50	10	433	30	29	30	30	37.23	36.95	54.09	379.05	624	160	0.01	0.01
100	1	798	0	0	0	0	1800.00	1800.00	0.00	0.00	9312	1458	-	-
100	5	921	12	3	30	30	1800.00	1800.00	466.38	464.39	3237	568	0.01	0.06
100	10	871	30	25	30	30	112.16	84.83	156.15	652.92	1360	259	0.01	0.01
$E_v^{\text{max-ub}}$														
5	1	43	29	22	30	30	97.67	21.09	413.22	788.86	450	451	0.01	0.01
5	5	47	30	30	30	30	1.60	1.51	4.97	6.91	61	83	0.00	0.01
5	10	44	30	30	30	30	0.42	1.00	0.51	0.35	23	25	0.00	0.00
10	1	88	22	11	29	30	664.20	1800.00	649.23	798.12	1034	621	0.01	0.02
10	5	94	30	30	30	30	5.81	3.90	24.30	30.60	150	112	0.01	0.01
10	10	86	30	30	30	30	1.23	1.28	0.81	1.10	54	35	0.00	0.01
20	1	175	2	2	6	30	1800.00	1800.00	243.49	407.14	2413	747	0.03	0.06
20	5	191	30	28	30	30	38.23	22.91	104.45	458.57	342	144	0.01	0.01
20	10	176	30	30	30	30	3.21	2.05	1.92	2.67	114	57	0.01	0.01
50	1	437	0	0	0	18	1800.00	1800.00	0.00	0.00	5772	910	-	0.32
50	5	475	28	18	30	30	457.69	577.00	547.39	825.31	974	304	0.01	0.01
50	10	438	30	30	30	30	13.27	9.13	15.42	7.01	298	99	0.01	0.01
100	1	875	0	0	0	0	1800.00	1800.00	0.00	0.00	9514	1445	-	-
100	5	953	16	12	30	30	1711.40	1800.00	551.41	720.21	2114	435	0.01	0.02
100	10	877	30	30	30	30	44.80	11.77	22.66	8.55	625	183	0.01	0.01

Table 7.5: EVS-SOC-GLIN results for solving the static model versus B&C with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ based on five-segment piecewise linear approximations of the original $P_v^{\max}$ functions, $P^{\text{gridmax}} = 25n$.

n	Δt (min)	n_{opt}		n_{feas}		Runtime (s)				Relative Gap (%)	
						Median		StdDev		Median	
		λ	NET	λ	NET	λ	NET	λ	NET	λ	NET
$\Delta E = 5 \cdot \Delta t$											
5	1	1	4	4	9	1800.00	1800.00	228.63	630.91	0.08	100.07
5	5	30	30	30	30	31.21	15.92	78.30	60.71	0	0
5	10	30	30	30	30	1.14	1.38	3.66	2.64	0	0
10	1	0	1	0	2	1800.00	1744.32	0.00	78.74	-	0.01
10	5	30	30	30	30	150.52	59.69	137.35	70.45	0.01	0
10	10	30	30	30	30	3.92	3.29	9.59	3.52	0	0
20	1	0	0	0	0	1800.00	-	0.00	-	-	-
20	5	22	30	30	30	866.66	174.27	602.53	147.37	0.01	0.01
20	10	29	29	29	29	22.73	8.21	19.20	5.94	0.01	0
50	1	0	0	0	0	1800.00	-	0.00	-	-	-
50	5	0	29	2	30	1800.00	663.61	0.00	417.12	0.07	0.01
50	10	29	29	29	29	94.09	29.21	164.54	24.19	0.01	0
100	1	0	0	0	0	1800.00	-	0.00	-	-	-
100	5	0	15	0	18	1800.00	1553.97	0.00	485.46	-	0
100	10	28	29	29	29	416.53	74.34	457.50	47.82	0.01	0
$\Delta E = 10 \cdot \Delta t$											
5	1	5	9	9	25	1800.00	1800.00	527.35	746.94	0.01	13.82
5	5	28	28	28	28	9.30	3.42	27.27	11.1	0	0
5	10	30	30	30	30	0.58	0.37	0.72	0.75	0	0
10	1	0	4	0	12	1800.00	1800.00	328.57	708.27	-	104.24
10	5	27	27	27	27	76.56	9.24	81.55	11.25	0.01	0
10	10	30	30	30	30	1.43	0.95	2.91	0.98	0	0
20	1	0	0	0	2	1800.00	1800.00	328.48	1037.07	-	110.85
20	5	24	26	26	26	275.91	22.48	444.30	29.46	0.01	0
20	10	29	29	29	29	3.88	1.55	9.52	3.12	0.01	0
50	1	0	0	0	0	1800.00	-	548.41	-	-	-
50	5	6	23	19	23	1800.00	70.72	794.35	96.26	0.03	0
50	10	27	27	27	27	16.16	5.24	29.53	3.57	0.01	0
100	1	0	0	0	0	1800.00	-	547.58	-	-	-
100	5	0	22	2	22	1800.00	129.58	809.32	200.7	0.04	0
100	10	26	26	26	26	79.43	11.57	130.83	7.4	0.01	0

Table 7.6: EVS-SOC-λ (λ) and EVS-SOC-NET (NET) with $E_v^{\text{max-lb}}$ based on the original $P_v^{\max}$, $P^{\text{gridmax}} = 25n$, on the restricted set of the first ten instances per n and Δt.

Comparing both formulations, the network model finds more feasible solutions and also solves more solutions to optimality for any n, Δt, and ΔE. Also, the network model has better median runtime for almost all parameter combinations. For some parameter groups, the relative gap of EVS-SOC-NET is larger than the relative gap of EVS-SOC-λ, which can be explained by the fact that the network model finds more feasible solutions.

For a fixed n, Δt, and model, it can be seen that smaller ΔE values have a negative impact on the runtime. Interestingly, smaller Δt values seem to have a stronger impact on the runtime than for the EVS-SOC-GLIN approach, whose corresponding results are depicted in Table 7.3. This comes at no surprise, since a smaller Δt also implies a smaller

ΔE. Lastly, it should be mentioned that for fixed Δt and any of the energy-discretized models, scaling the number of vehicles seems to have less impact on the runtime than for EVS-SOC-GLIN.

7.2 Charging Cost Differences & Charging Errors

While the simpler approximations of the original $P_v^{\max}$ functions lead to shorter runtimes, there is clearly a tradeoff concerning the precision of the model, introduced errors, and final solution qualities. We have a closer look on these aspects in the following.

First, we evaluate EVS-SOC-GLIN on six different $E_v^{\max}$ functions: $E_v^{\text{max-ex}}$, $E_v^{\text{max-lb}}$, and $E_v^{\text{max-ub}}$, each based on the five-segment $P_v^{\max}$ approximation and the original $P_v^{\max}$. Since we want to measure the impact of the different charging curves on the charging costs, we select a high P^{gridmax} value of $40n$ as in this case the variable maximum charging power constraints have higher impact. Only results on instances solved to optimality are reported. Also, we only consider instances where a solution for all six $E_v^{\max}$ functions was found. Parameter combinations where no such instances exist were omitted. The median charging costs can be found in Table 7.7. Values in the brackets next to $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ state the charging cost gaps to $E_v^{\text{max-ex}}$ in percent, i.e., $100\% \cdot (|E_v^{\text{max-ex}} - E_v^{\max}|)/E_v^{\text{max-ex}}$ for $E_v^{\max} \in \{E_v^{\text{max-lb}}, E_v^{\text{max-ub}}\}$.

n	Δt (min)	Median Charging Costs				
		$E_v^{\text{max-lb}}$	(%-gap)	$E_v^{\text{max-ex}}$	$E_v^{\text{max-ub}}$	(%-gap)
Original $P_v^{\max}$						
5	1	109.08	0.10	108.97	108.86	0.09
5	5	213.13	0.28	212.27	211.30	0.31
5	10	240.32	0.54	239.25	238.53	0.51
10	5	398.31	0.33	396.74	395.08	0.34
10	10	444.95	0.50	441.55	438.64	0.41
20	10	867.07	0.55	859.80	851.13	0.48
5-segment approx. $P_v^{\max}$						
5	1	109.10	0.10	108.98	108.88	0.10
5	5	213.10	0.28	212.45	211.53	0.31
5	10	240.41	0.50	239.34	238.67	0.48
10	5	398.27	0.31	396.78	395.15	0.33
10	10	444.74	0.48	441.38	438.67	0.40
20	10	867.02	0.54	859.65	851.57	0.47

Table 7.7: Objective value comparison using EVS-SOC-GLIN and different $E_v^{\max}$ functions based on the five-segment $P_v^{\max}$ approximation and the original $P_v^{\max}$; $P^{\text{gridmax}} = 40n$.

Observe that for fixed Δt and varying n, the relative median charging cost difference of $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ to $E_v^{\text{max-ex}}$ does not change significantly. One might notice that the relative median charging cost gaps are smaller for decreasing Δt values. Overall, the largest relative median charging cost difference is 0.55%, the differences are therefore negligible for the shown parameter groups.

It is worth mentioning that for fixed n and Δt, the five-segment approximation of P_v^{max} influences the charging costs only marginally, even for large instances. For example consider $n = 20$, $\Delta t = 10$ minutes and $E_v^{\text{max-ex}}$ and observe that the objective value differs by about 0.15 cent only between the original P_v^{max} and the five-segment approximation. This insight seems to be particularly relevant, since it shows that approximating P_v^{max} with a lower number of linear pieces is reasonable for practice.

When realizing a charging plan in practice with a different P_v^{max} function than used for scheduling, the specified target SOCs s_v^{dep} might not be reached for some vehicles. We measure this error by generating a charging schedule with $E_v^{\text{max-ub}}$ and simulating the actual maximum energy function with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$. In the simulation, the actually charged energy is set to be the minimum from the corresponding planned charged energy and the actual maximum energy function. The resulting mean deviation from the target SOC, the mean charging error, can be seen in Table 7.8. For a single instance, we determined the mean charging error over all vehicles, whereas for an instance group we again took the mean of the charging errors from the individual instances.

n	Δt (min)	Mean Charging Error (% SOC)			
		Original P_v^{max}		5-seg. approx. P_v^{max}	
		$E_v^{\text{max-lb}}$	$E_v^{\text{max-ex}}$	$E_v^{\text{max-lb}}$	$E_v^{\text{max-ex}}$
5	1	0.59	0.32	0.57	0.29
5	5	2.54	1.37	2.55	1.37
5	10	4.00	2.08	3.79	1.97
10	1	0.61	0.32	0.59	0.30
10	5	2.67	1.45	2.67	1.45
10	10	4.08	2.18	4.08	2.15
20	1	0.51	0.27	0.52	0.26
20	5	2.77	1.51	2.73	1.49
20	10	4.36	2.37	4.37	2.38
50	5	2.77	1.51	2.74	1.50
50	10	4.40	2.40	4.40	2.40
100	5	2.88	1.58	2.88	1.60
100	10	4.51	2.45	4.49	2.45

Table 7.8: Charging error comparison when scheduling with $E_v^{\text{max-ub}}$ using EVS-SOC-GLIN and realizing the schedule with $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ex}}$. $P^{\text{gridmax}} = 40n$.

Fixing n, Δt and the number of P_v^{max} segments, it can be seen that the mean charging error is higher for $E_v^{\text{max-lb}}$ than for $E_v^{\text{max-ex}}$. Also for a fixed n and E_v^{max}, the mean charging error decreases with smaller Δt. On the contrary, the number of vehicles does not seem to influence the mean charging error for fixed Δt and E_v^{max}.

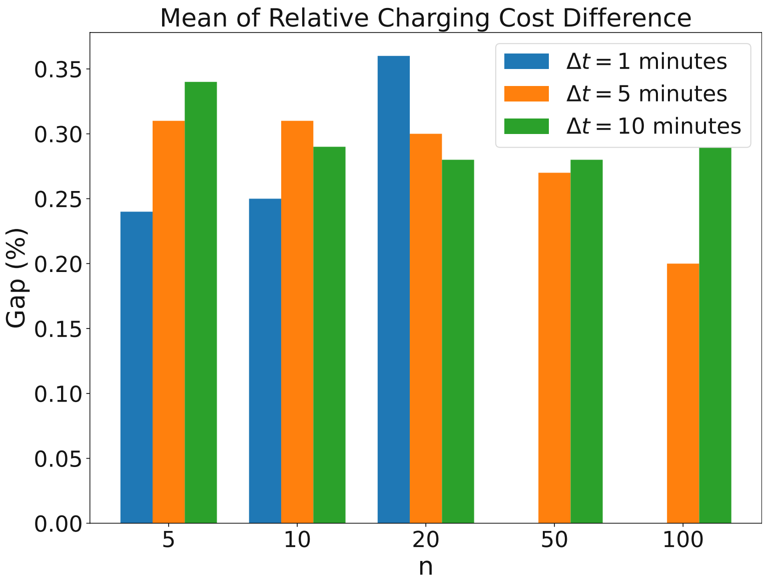


Figure 7.2: Mean charging cost gaps of EVS-SOC-LIN and EVS-SOC-GLIN with $P^{\text{gridmax}} = 40n$.

EVS-SOC-LIN and EVS-SOC-GLIN. Charging cost gaps between solutions of formulation EVS-SOC-LIN and EVS-SOC-GLIN can be found in Figure 7.2. As before, we only consider instances that were solved to optimality. For EVS-SOC-LIN we use $E_v^{\text{max-lb}}$ based on the concave P_v^{max}, whereas for EVS-SOC-GLIN we use $E_v^{\text{max-lb}}$ based on P_v^{max} with 5 segments. P^{gridmax} is again set to $40n$. For $n \in \{50, 100\}$ and $\Delta t = 1$ minute, all mean charging cost gaps are zero, therefore the respective bars are not shown in the figure. Comparing the gaps of both formulations, one can notice that there are no significant differences for varying n or Δt values. The charging costs differ at most 0.35% for $n = 20$, $\Delta t = 1$ minute. However, when it comes to runtime, both variants of EVS-SOC-LIN are significantly faster than any EVS-SOC-GLIN variant, as we have seen before in Table 7.1 and Table 7.3.

For the exact same setting as above, we also measure the charging error when scheduling with the convex $E_v^{\text{max-lb}}$ used in EVS-SOC-LIN and realizing the plan with the, in general, non-convex $E_v^{\text{max-lb}}$ used in EVS-SOC-GLIN. The mean charging error is shown in Figure 7.3. It can be said that for a fixed Δt, the mean charging error does not significantly change for a varying number of vehicles. However, for a fixed n, the mean charging error grows with decreasing Δt. An explanation for this observation might be that on instances with smaller Δt, solutions tend to be more precise in terms of the error induced by time discretization. Therefore the difference between a convex and non-convex E_v^{max} function could have more impact on solutions of instances with small Δt values. Overall, the mean charging cost difference does not exceed 1.5% SOC for any n and any Δt and may be neglected in practice.

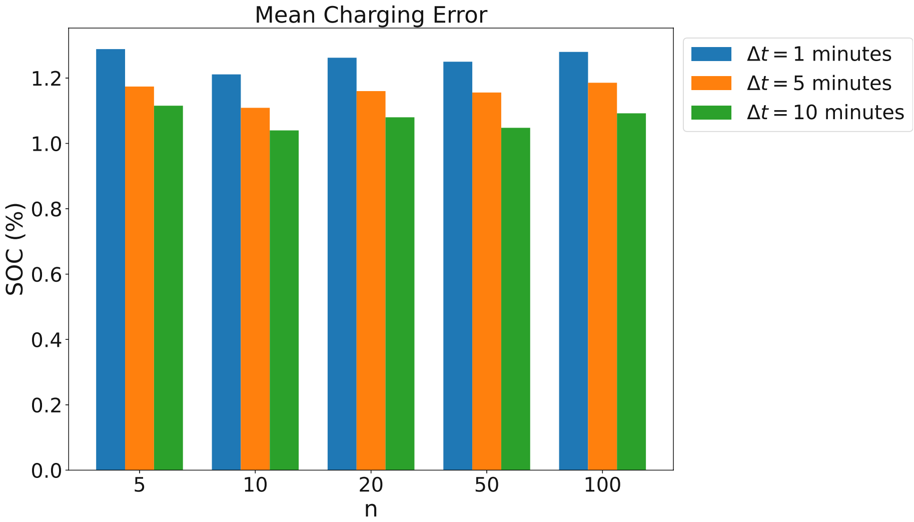


Figure 7.3: Mean charging error when scheduling with convex $E_v^{\text{max-lb}}$ and realizing the plan with non-convex $E_v^{\text{max-lb}}$ using $P^{\text{gridmax}} = 40n$.

Model Based Predictive Control Simulations. For the rolling horizon scenarios, we conduct experiments using formulation EVS-SOC-LIN and EVS-SOC-GLIN. We use $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ for both formulations, but for EVS-SOC-LIN the corresponding concave approximation of P_v^{max}, whereas for EVS-SOC-GLIN the five-segment approximation of P_v^{max}. P^{gridmax} is set to $40n$. The results of the experiment are shown in Table 7.9. Absolute charging cost differences are determined by subtracting the EVS-SOC-GLIN objective from the EVS-SOC-LIN objective. Relative charging costs are based on the absolute charging costs divided by the objective of EVS-SOC-GLIN.

Similar to before, for fixed n, Δt, and E_v^{max}, the charging costs of EVS-SOC-LIN and EVS-SOC-GLIN only differ marginally. The maximum gap is 0.27% for $n = 100$, $\Delta t = 5$ minutes, and $E_v^{\text{max-lb}}$. As expected, the absolute charging cost difference increases with a higher number of vehicles. The relative gaps, however, seem to remain in the same order of magnitude for growing n.

n	Δt (min)	E_v^{max}	Charging Cost Difference			
			Absolute (cent)		Relative (%)	
			Mean	StdDev	Mean	StdDev
5	5	$E_v^{\text{max-lb}}$	0.97	0.73	0.22	0.16
5	5	$E_v^{\text{max-ub}}$	0.70	0.54	0.16	0.12
5	10	$E_v^{\text{max-lb}}$	0.91	0.60	0.20	0.12
5	10	$E_v^{\text{max-ub}}$	0.54	0.49	0.13	0.11
10	5	$E_v^{\text{max-lb}}$	1.75	0.99	0.20	0.11
10	5	$E_v^{\text{max-ub}}$	1.18	0.80	0.14	0.09
10	10	$E_v^{\text{max-lb}}$	1.78	0.77	0.20	0.08
10	10	$E_v^{\text{max-ub}}$	0.80	0.54	0.09	0.06
20	5	$E_v^{\text{max-lb}}$	3.78	1.34	0.21	0.08
20	5	$E_v^{\text{max-ub}}$	2.52	1.07	0.14	0.06
20	10	$E_v^{\text{max-lb}}$	3.80	1.03	0.21	0.06
20	10	$E_v^{\text{max-ub}}$	1.81	0.70	0.10	0.04
50	5	$E_v^{\text{max-lb}}$	9.14	2.42	0.20	0.05
50	5	$E_v^{\text{max-ub}}$	6.38	1.98	0.14	0.04
50	10	$E_v^{\text{max-lb}}$	9.39	2.64	0.21	0.06
50	10	$E_v^{\text{max-ub}}$	4.28	1.13	0.10	0.03
100	5	$E_v^{\text{max-lb}}$	24.42	2.40	0.27	0.03
100	5	$E_v^{\text{max-ub}}$	10.67	5.45	0.12	0.06
100	10	$E_v^{\text{max-lb}}$	19.96	4.82	0.22	0.05
100	10	$E_v^{\text{max-ub}}$	8.75	2.27	0.10	0.02

Table 7.9: Rolling Horizon charging cost difference for EVS-SOC-LIN vs EVS-SOC-GLIN using $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$; $P^{\text{gridmax}} = 40n$.

CHAPTER 8

Conclusions

In this study, we formally introduced the EVS-SOC problem in which we put particular focus on dealing with vehicle-specific, state of charge dependent maximum charging power limitations. We addressed the issue that the maximum charging power $P_v^{\max}$ may be regulated within a single time step by turning towards considering the maximum amount of energy that can be charged in a time step. To this end, we proposed an exact derivation $E_v^{\text{max-ex}}$ as well as simpler lower and upper bounds $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$. One should keep in mind that the gap between $E_v^{\text{max-lb}}$ and $E_v^{\text{max-ub}}$ decreases with smaller time interval length Δt. Once more, we recall that charging schedules generated with $E_v^{\text{max-lb}}$ are guaranteed to be realizable in practice, whereas additional schedules generated with $E_v^{\text{max-ub}}$ help us with the estimation of the charging cost differences and charging errors induced by the time discretization.

Let us recapitulate the most important experimental results in the following. Four different mixed-integer linear programming formulations, EVS-SOC-LIN, EVS-SOC-GLIN, EVS-SOC-λ, and EVS-SOC-NET, were proposed. Although EVS-SOC-λ and EVS-SOC-NET might scale well with the number of vehicles, they pose the problem of selecting an appropriate energy unit ΔE and quickly run out of memory for smaller ΔE values. At this point it should be stressed again that the size of ΔE must be chosen in dependence of Δt. For this reason, comparing the energy discretized models to other formulations is difficult, since ΔE clearly has an impact on the runtime of EVS-SOC-λ, and EVS-SOC-NET. Comparing the energy discretized formulations to each other, EVS-SOC-NET has been shown to be a stronger model w.r.t. the LP relaxation and also performs significantly better than EVS-SOC-λ in practice on almost all parameter groups.

When taking a closer look at EVS-SOC-LIN, we realized that the static and the cutting plane approach are both quite fast. Compared to the other proposed formulations, EVS-SOC-LIN performs an order of magnitude faster in our experiments. Considering the runtime difference between the static and the cutting plane approach, a substantial performance benefit of the latter can be observed. Moreover, we have seen that the runtime of the cutting plane approach scales better with an increasing number of vehicles or decreasing Δt values. Its advantages become even more visible when the maximum charging energy of a vehicle has to be exploited, i.e., a large number of cuts has to be separated.

Concerning the static and branch-and-cut approach of EVS-SOC-GLIN, we found that the branch-and-cut (B&C) variant performs better with a small number of vehicles. For larger instances, however, the static variant is superior in terms of runtime. It also shows performance advantages for larger grid capacity (P^{gridmax}) values. Results of the experiments indicate that the B&C approach is slower than the static variant when a large number of cuts has to be separated. Nevertheless, there are a few instances where B&C is faster. Additionally, we realized that B&C finds more feasible solutions in the majority of the experiments. Overall, for both EVS-SOC-GLIN approaches, it is also worth mentioning that fewer $P_v^{\max}$ segments clearly reduced the runtime.

Different approximations of the maximum charging power (e.g. piecewise linear approximation or convex hull approximation), as well as the maximum charging energy ($E_v^{\text{max-lb}}$, $E_v^{\text{max-ub}}$) have been proposed. We studied the charging cost differences and the charging errors induced by these approximations. Regarding the charging cost differences, it turned out that there are only marginal charging cost differences between schedules generated with $E_v^{\text{max-lb}}/E_v^{\text{max-ub}}$ and schedules generated with $E_v^{\text{max-ex}}$. The number of vehicles did not show any noticeable impact on the cost differences for this comparison. Nevertheless, we observed that a smaller time interval length Δt reduces the charging cost differences. Additionally, it was shown that for our instances the approximation of $P_v^{\max}$ with 5 piecewise linear segments does not have any noticeable impact on the charging costs. We also inspected the charging cost differences when generating schedules based on the original $P_v^{\max}$ function and its concave approximation. It turned out that the charging cost differences are quite small, specifically the median charging cost differences did not exceed 0.35% for any shown parameter group.

As already mentioned, approximating the maximum charging energy might lead to the issue that vehicles do not reach their desired target state of charge. To measure these effects, we generated charging schedules with $E_v^{\text{max-ub}}$ and simulated the actual charging with $E_v^{\text{max-lb}}$. Experimental results have shown that the mean charging error does not exceed 4.5% SOC even for $\Delta t = 10$ minutes. For this experiments, we could also detect a correlation between the size of Δt and the charging error, more specifically the mean charging error decreases with smaller Δt. In another simulation setting, we considered the mean charging error when generating a charging schedule based on a concave $P_v^{\max}$ approximation and realizing it with the original $P_v^{\max}$. The mean charging error is rather small, the mean deviation from the vehicle's target state of charge are at most 1.5%.

To see whether the concave approximation of $P_v^{\max}$ accumulates large charging cost differences in a whole day scenario, we conducted model based predictive control simulations with the original $P_v^{\max}$ and its concave approximation. The relative charging cost gaps were even smaller with a maximum value 0.27% for 100 vehicles and $\Delta t = 5$ minutes.

In a bigger picture, where we utilize one of the formulations within a model based predictive control strategy, we would discourage the usage of the energy discretized models due to the aforementioned problems with ΔE. Instead, we recommend the usage of EVS-SOC-LIN or EVS-SOC-GLIN together with a reasonably small Δt value of few minutes, in order to reduce errors introduced by time discretization. Depending on whether EVS-SOC-GLIN is performant enough for a presumed problem setting (i.e., it finds a charging schedule within the reoptimization interval) its usage is advised to reduce the danger of significant charging cost differences and charging errors. It seems promising to approximate $P_v^{\max}$ with 5 to 10 piecewise linear segments to improve runtime in this scenario.

In case EVS-SOC-GLIN does not find charging schedules in reasonable time, one might fall back on EVS-SOC-LIN and its cutting plane approach to rapidly generate charging schedules with concave $P_v^{\max}$ functions. The introduced errors are insignificant as we have seen.

In future work it would be interesting to investigate whether the runtime of EVS-SOC-GLIN can be improved. On some instances, the branch-and-cut technique is slower than the static variant. One could try to add different cuts to the model and thereby speed up the solving process. Besides that, it might be possible to come up with a different formulation than EVS-SOC-GLIN, that can also handle non-concave, piecewise linear maximum charging power functions.

Concerning the computational hardness of EVS-SOC from a theoretical point of view, it is an open question whether the problem is tractable for general maximum power functions, i.e, whether it is in P or NP-hard. Especially the hardness of the problem if $P_v^{\max}$ is non-concave and piecewise linear is an important problem setting that should be investigated.

Another aspect worth pursuing is the question whether known vehicle arrival times have a significant impact on the charging costs of a rolling horizon schedule. In the presented scenario, successively arriving vehicles are simulated, however they are not incorporated into the schedule until arrival at the charging station. One may expect that priorly known arrival times lead to better exploitation of cheap charging time slots and therefore come along with cheaper total charging costs. Furthermore it would be interesting to study the effect of the rescheduling interval on charging costs and charging errors in the rolling horizon context.

One could also investigate a similar problem variant, where we allow discharging of vehicles in order to enable mutual charging of EVs. This idea has already been mentioned in [IL20], however its impact on the total charging costs has not yet been studied. One could further extend the model by allowing the charging station to supply energy to the electricity grid in exchange for monetary reward.

List of Figures

List of Tables

Bibliography

[BT97] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear organisation*, volume 6 of *Athena scientific optimization and computation series*. Athena Scientific, 1997.

[CA13] E.F. Camacho and C.B. Alba. *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2013.

[CTL+12] Yijia Cao, Shengwei Tang, Canbing Li, Peng Zhang, Yi Tan, Zhikun Zhang, and Junxiong Li. An optimized EV charging model considering TOU price and SOC curve. *IEEE Transactions on Smart Grid*, 3(1):388–393, 2012.

[EBMS+18] Claude Ziad El-Bayeh, Imad Mougharbel, Maarouf Saad, Ambrish Chandra, Dalal Asber, and Serge Lefebvre. Impact of considering variable battery power profile of electric vehicles on the distribution network. In *2018 4th International Conference on Renewable Energies for Developing Countries (REDEC)*, pages 1–8, 2018.

[Fas20] Fastned. Fastned – Supersnel laden langs de snelweg en in de stad: www.fastnedcharging.com, 2020.

[HHS10] Sekyung Han, Soohee Han, and Kaoru Sezaki. Development of an optimal vehicle-to-grid aggregator for frequency regulation. *IEEE Transactions on Smart Grid*, 1(1):65–72, 2010.

[HPL17] Jinil Han, Jongyoon Park, and Kyungsik Lee. Optimal scheduling for electric vehicle charging under variable maximum charging power. *Energies*, 10(7):933, 2017.

[IL20] Takahiro Ishihara and Steffen Limmer. Optimizing the hyperparameters of a mixed integer linear programming solver to speed up electric vehicle charging control. In Pedro A. Castillo, Juan Luis Jiménez Laredo, and Francisco Fernández de Vega, editors, *Applications of Evolutionary Computation*, volume 12104 of *LNCS*, pages 37–53. springer, 2020.

[JV19] Charles F. Jekel and Gerhard Venter. *pwlf: A Python Library for Fitting 1D Continuous Piecewise Linear Functions*, 2019.

[JZ19] Wei Jiang and Yongqi Zhen. A real-time EV charging scheduling for parking lots with PV system and energy store system. *IEEE Access*, 7:86184–86193, 2019.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[KCE18] Majid Khonji, Sid Chi-Kin Chau, and Khaled M. Elbassioni. Approximation scheduling algorithms for electric vehicle charging with discrete charging options. In Hartmut Schmeck and Veit Hagenmeyer, editors, *Proceedings of the Ninth International Conference on Future Energy Systems, e-Energy 2018*, pages 579–585. ACM, 2018.

[KS17] Nikita Korolko and Zafer Sahinoglu. Robust optimization of EV charging schedules in unregulated electricity markets. *IEEE Transactions on Smart Grid*, 8(1):149–157, 2017.

[KS20] Aastha Kapoor and Ankush Sharma. Optimal charge/discharge scheduling of battery storage interconnected with residential PV system. *IEEE Systems Journal*, 14(3):3825–3835, 2020.

[Lim20] Steffen Limmer. Electric vehicle charging control with consideration of hierarchical objectives. Technical report, Honda Research Institute Europe, 2020.

[MCDM20] T. Morstyn, C. Crozier, M. Deakin, and M. D. McCulloch. Conic optimization for electric vehicle station smart charging with battery voltage constraints. *IEEE Transactions on Transportation Electrification*, 6(2):478–487, 2020.

[MG15] Joy Chandra Mukherjee and Arobinda Gupta. A review of charge scheduling of electric vehicles in smart grid. *IEEE Syst. J.*, 9(4):1541–1553, 2015.

[MN19] Keisuke Mizuno and Toru Namerilkawa. Optimization of power flow and scheduling for EV charging based on distributed control. In *12th Asian Control Conference, ASCC 2019, Kitakyushu-shi, Japan*, pages 627–631. IEEE, 2019.

[PCdA14] Fabio Antonio V. Pinto, Luís Henrique M. K. Costa, and Marcelo Dias de Amorini. Modeling spare capacity reuse in EV charging stations based on the li-ion battery profile. In *International Conference on Connected Vehicles and Expo 2014, ICCVE*, pages 92–98. IEEE, 2014.

[SB10] Olle Sundström and Carl Binding. Optimization methods to plan the charging of electric vehicle fleets. In *Proceedings of the international conference on control, communication and power engineering*, pages 28–29. Citeseer, 2010.

[SHTT18] Bo Sun, Zhe Huang, Xiaoqi Tan, and Danny H. K. Tsang. Optimal scheduling for electric vehicle charging with discrete charging levels in distribution grid. *IEEE Transactions on Smart Grid*, 9(2):624–634, 2018.

[Wol98] L.A. Wolsey. *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1998.