Improving EV Aggregators' Workplace Charging

A Safe Reinforcement Learning Approach

MSc thesis

Ruben Eland

Delft University of Technology

TU Delft

Improving EV Aggregators' Workplace Charging

A Safe Reinforcement Learning Approach

by

Ruben Eland

to obtain the degree of Master of Science
at the Delft University of Technology,
on Thursday, June 12, 2025 at 9:00 AM

Student number:	4868463	
Project duration:	September 2024 – May 2025	
Thesis committee:	S. Orfanoudakis,	IEPG, ESE Department, Daily Supervisor
	Dr. P.P. Vergara,	IEPG, ESE Department, Supervisor
	Dr. S.H. Tindemands,	IEPG, ESE Department, Chair
	Dr. N. Yorke-Smith,	Algorithmics, External Member

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Horizon 2020
European Union funding
for Research & Innovation

This thesis is partly done within the Drive2X project with grant number 101056934, funded by the European Union. For more information, you can visit the corresponding webpage: https://drive2x.eu/.

Preface

In our current turbulent and war-filled world, one may easily forget about that other enormous challenge of this century: the energy transition. While well underway in terms of renewable energy production, this is only the start. And the challenge lies much further than the energy system. How do we decarbonize our plastic products? How do we reduce CO2 emissions in the building sector?

Fortunately, I can leave these questions to other scientists. As an electrical engineer my focus does lie in the power system. Throughout my studies one word kept on returning: flexibility. We need it on the power production side, and we need it on the power consumption side. We need it very quickly to prevent congestion, and we will need it forever to balance the variable power production of renewable power sources.

At one point in my master, it struck me: why do we not use the flexibility of electric vehicle charging? On paper it is perfect: electric vehicles have large batteries, they use much power for charging, and most importantly, they are connected to chargers often without charging–at least in the Netherlands with our high share of AC chargers and short trips.

During my thesis, I learned why this potential is barely utilized today. The electric vehicle charging sector is young and quite complex. There are numerous active players, some taking care of one specific part, others trying to fulfill the whole charging service. As is typical for any brand-new technology industry in the past 50 years (microprocessors, the internet, digital wireless telecommunication), the beginning of the industry is characterized by chaos and an unforeseeable future.

The result, in my opinion, is a fantastic playground for innovation and impact. Players who, in this phase, look three steps ahead of the rest have a chance to shape an entire sector. And in the case of electric vehicle charging, they can unlock all potential benefits for the energy transition. I am very happy to have been able to dive deep into these topics for my master's thesis.

I want to express sincere gratitude to my supervisor, Dr. Pedro P. Vergara, and daily supervisor, Stavros Orfanoudakis. They have efficiently guided me through the thesis, always stimulating me to ask the right questions and to keep moving forward. I want to thank Pedro especially for the interesting discussions about the broader unfolding of the energy transition. I want to thank Stavros for his huge patience in helping me with the coding. Surely, without him, I would never have been able to learn this much about applied EV charging optimization and Reinforcement Learning.

Now, with a job related to smart charging, I will continue to work on harnessing the full potential flexibility of electric vehicle charging. We need it as much as possible, as soon as possible.

Ruben Eland
Rotterdam, June 2025

Abstract

As the number of Electric Vehicles (EVs) and renewable energy sources (RES) increases rapidly, power grids struggle to adapt. In the coming years, power system flexibility is urgently required to use the limited capacity of the existing infrastructure efficiently. Over the long term, flexibility will remain essential to account for the variable and uncertain electricity production from RES. EVs have large batteries that are often only partly used for daily travel, particularly in densely populated areas. Smart charging and Vehicle-to-Grid (V2G) can harness the flexibility of EVs to support grid balancing and congestion management.

This thesis investigates the smart charging and V2G potential for EV aggregators, with a focus on workplace charging. State-of-the-art Reinforcement Learning (RL) techniques are applied to a case study involving a business parking lot. The objective is to maximize the profits of the EV aggregator while satisfying EV user and transformer power limit constraints. The modeled EV behavior is based on data from real EV measurements in the Netherlands. The real-time charging optimization problem is characterized by high uncertainty. RL is widely considered a promising algorithm for solving highly uncertain problems. However, the latest Deep RL algorithms often struggle to guarantee constraint-satisfying behavior. Safe RL, an emerging subfield, aims to reduce constraint violations in the learned behavior, thus making algorithms 'safer'. This thesis applies recent Safe RL algorithms and compares their performance to Deep RL baselines and conventional As-Fast-As-Possible (AFAP) charging.

The proposed method, Constrained Variational Policy Optimization (CVPO), achieved performance comparable to that of the optimal offline Gurobi solver in simulation scenarios where sufficient transformer capacity was available and overloads could not occur. The learned behavior generalized well to unseen levels of charger occupation. However, in scenarios with more inflexible loads and a smaller transformer power limit, transformer overloading risk made the problem more constrained, resulting in a decline in CVPO's performance.

The code used to generate the results of this thesis is publicly available at
`https://github.com/rubeneland/EV2Gym_safeRL`.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AFAP	As-Fast-As-Possible
BEV	Battery Electric Vehicle
BRP	Balancing Responsible Partner
BSP	Balancing Service Provider
CPO	Constrained Policy Optimization, Safe RL algorithm
CSP	Congestion Service Provider
CVPO	Constrained Variational Policy Optimization, Safe RL algorithm
DSO	Distribution System Operator
DDPG	Deep Deterministic Policy Gradient, RL algorithm
EV	Electric Vehicle (refers to fully electric BEV unless otherwise specified)
LFP	Lithium Iron Phosphate
MPC	Model Predictive Control
NMC	Nickel Manganese Cobalt
PHEV	Plug-in Hybrid Electric Vehicle
PPO	Proximal Policy Optimization, RL algorithm
PPO-L	PPO with Lagrangian cost function, Safe RL algorithm
RES	Renewable Energy Sources
RL	Reinforcement Learning
SAC	Soft Actor-Critic, RL algorithm
SAC-L	SAC with Lagrangian cost function, Safe RL algorithm
SOC	State-Of-Charge (Battery %)
TD3	Twin-delayed DDPG, RL algorithm
TOU	Time-Of-Use
TSO	Transmission System Operator
V2G	Vehicle-to-Grid

1

Introduction

The global pursuit of a fossil-free future has accelerated investments in renewable energy technologies, leading to unprecedented transformations in the energy and transportation sectors. Renewable Energy Sources (RES), such as PV, wind, and hydro, are projected to contribute 35% to global electricity generation by 2025, surpassing coal as the dominant energy source [1]. Similarly, electric vehicles (EVs) have surged in popularity, with their share of global vehicle sales increasing from 4% in 2020 to 18% in 2023 [2]. These rapid adoptions create new challenges within the power grid.

The energy production from RES is variable, uncertain, and often has a locational mismatch to energy demand [3]. Furthermore, their scattered distribution requires a revision of the current power network. The large electricity needs of EVs also require upgrades of the power grid. The pace of the transformations within the energy and transportation sectors troubles the adaptation of the grid, where upgrading existing infrastructure can take several years.

Many experts argue that a new approach to the operation of the power system is crucial to continue the progress of adopting RES in the energy mix [3] [4] [5] [6]. Besides enhancing transmission capacity, proper integration of RES means increasing flexibility and energy storage [3]. EVs are often incorporated into the challenges because of their large electricity needs. However, proper management could make them part of the solution. As EVs typically spend a lot of time connected to chargers but without charging [7], EV aggregators–entities controlling the charging of groups of EVs–could use their large batteries to provide part of the energy storage and flexibility which the power system desperately needs.

1.1. The power system

To improve the relevance of this thesis, the research is implemented in the context of the power system of the Netherlands. This chapter introduces key actors, market forces, and challenges within the Dutch power system.

1.1.1. Network operators

The operation of a power grid is usually divided between Transmission System Operators (TSOs) and Distribution System Operators (DSOs). In the Netherlands, TenneT acts as the sole TSO and is in charge of the high-voltage part of the grid. In addition to maintaining and expanding the high-voltage grid, TSOs are responsible for **grid balancing**, i.e. ensuring power production equals power consumption. Furthermore, the TSO facilitates the national power market and the TSOs in Europe work together in a European market to efficiently allocate energy resources throughout the continent [8].

Each DSO maintains and operates a section of the low- and medium-voltage grids. In the Netherlands Stedin, Enexis and Liander are some of the DSOs. DSOs also connect new producers and consumers to the power system. While grid balancing is a function of Tennet only, both the TSO and DSO are responsible for **avoiding congestion** in the grid [8]. Together, the TSO and DSOs facilitate the transportation of electricity from source to consumer.

The power grid of the Netherlands is highly congested as of 2025. A simple solution to this congestion would be to upgrade the existing infrastructure. While TenneT invests more than €6 billion each year to upgrade the grid [9], the duration of projects is too long to solve congestion issues promptly. The construction of a high-voltage substation typically takes five to ten years [9].

1.1.2. Electricity market

In the liberalized Dutch electricity market, energy is traded on the scale of MWh. In the **day-ahead market**, players can buy or sell electricity for the next day on an hourly basis. After the day-ahead market closes, the hourly prices are shared with market participants. The players can adjust their spot positions on the delivery day up to five minutes before the physical delivery in the **intraday market**. However, buying electricity is usually more expensive in the intraday market, compared to day-ahead [10].

In practice, electricity production and consumption forecasts will never be completely accurate, so there will always be some imbalance. The goal of the **imbalance market** is to solve imbalances before they can cause damage. TenneT can buy or sell energy in this market to solve any imbalance issue [10].

1.1.3. Key actors

Besides the TSO and DSO, several actors help keep the power system functioning properly. The **Balance Responsible Partner** (BRP) helps the TSO to maintain the power system's balance. Each BRP manages a portfolio with producers and consumers and tries to minimize the gap between production and consumption in their portfolio. TenneT charges or rewards the BRPs based on their performance. The **Balancing Service Provider** (BSP) offers balancing capacity or energy reserves, which TenneT can activate when required [8].

Electricity suppliers produce or buy power and sell it to consumers. The dominant contract form used to be a fixed-pricing scheme, in which suppliers charge consumers a fixed tariff per consumed kWh. In recent years, dynamic or time-of-use (TOU) pricing has emerged as an alternative. In dynamic electricity pricing, suppliers offer tariffs close to the day-ahead market prices, only adding a marginal compensation cost.

Figure 1.1 shows boxplots of average day-ahead market electricity prices in the Netherlands. Figure 1.1 was created with data from the ENTSO-E open data platform [11]. The data was downloaded with the Python package *entsoe-py* [12]. The script can be found in the Appendix: A.1. The electricity prices are first grouped by the hour to improve readability. Afterwards, the average hourly prices for each year are calculated and the boxplots are created. The Python script for creating the boxplots is also available in the Appendix: A.2. Figure 1.1 shows that after years of very stable electricity prices around 50 €/MWh, the prices skyrocketed up to 300 €/MWh around 2022 as a result of the war in Ukraine. Figure 1.1 also shows that although the mean electricity price has been cooling down in 2023 and 2024,

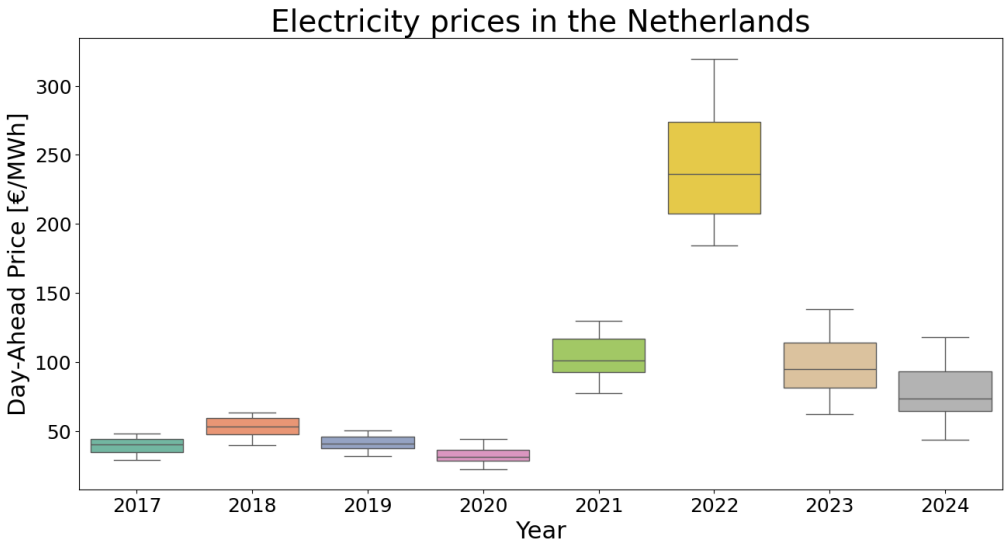


Figure 1.1: Boxplots of average electricity prices in the Netherlands show increased volatility. Created with open data from ENTSO-E [11].

the volatility of electricity prices remains. This can be explained by the increase of RES in the energy mix and the resulting imbalance between electricity production and consumption. Consequently, more than ever dynamic tariffs offer incentives to change one's electricity consumption.

The **Congestion Service Provider** (CSP) is a new entity in the power system. The CSP offers congestion management services to the TSO or DSO. Congestion can be relieved by actively changing the amount of power that flows through a congested area. These services are location-bound, unlike balancing services [8]. Since June 2022, a modification in the legislation allows four possible methods for CSPs to provide congestion services. GOPACS, a joint venture between TenneT and the DSOs, facilitates three of these four methods [13].

First, GOPACS implements congestion redispatch bids in a market environment, from now on referred to as the **congestion market**. Network operators can submit intraday redispatch bids in GOPACS in a congested situation. CSPs operating in the area with congestion can respond by placing a buy or sell order on an energy trading platform connected to GOPACS. If called, a buy order is coupled to a sell order outside the congested area and vice versa to prevent imbalance [13].

A CSP can also conclude an intraday bid obligation contract or a day-ahead capacity limitation contract with a grid operator. The first obliges CSPs to participate in redispatch bids. The limitation contract allows network operators to lower the capacity of a CSP the next day in return for monetary compensation. The contracts yield more certainty to both the operator and the customer, as congestion prevention is guaranteed and fees are agreed upon in the contract. [13].

The **Aggregator** is a relatively new actor in the power system as well. Aggregators combine multiple small consumers and producers in a portfolio and put the combined demand or production on the wholesale, balancing, or congestion market. Consumers allow aggregators to manage their assets, e.g. control the charging process of their EV, in return for financial compensation. An aggregator may also become a CSP, employing their aggregated capacity to mitigate local grid congestion [8]. The increasing penetration of RES and EVs into the grid makes the role of aggregators more important.

1.2. Flexibility

Several articles emphasize flexibility as a key factor in coping with many of the impacts of variability and uncertainty of RES [3] [4] [5] [6]. Flexibility can be defined as "the ability of a power system to reliably and cost-effectively manage the variability and uncertainty of supply and demand across all relevant timescales" [3].

In the coming years, more flexibility is required to prevent congestion while we continue the adoption of RES and EVs in anticipation of scheduled grid expansions. However, flexibility will always stay a key element in solving imbalances in the long term. TSOs still rely on conventional fossil-fuel power plants for grid balancing [6]. These plants' large inertia and easy controllability can be regarded as flexibility. In the future, as they are replaced with the uncertainty and scattered distribution of RES, flexibility has to be found in other areas, e.g. in energy storage or aggregators providing flexibility with their portfolio of EVs [6].

1.2.1. Demand response

Demand response implements flexibility on the consumer side: consumers shift flexible consumption from periods with power shortage to periods with excess power [6]. There are three main strategies to gain financial rewards from demand response.

The first and for consumers easiest strategy is through dynamic electricity tariffs. As dynamic tariffs follow the market price, low prices represent periods of power shortage and vice versa. Consumers with dynamic electricity contracts can thus easily provide flexibility without participating in market bidding processes. While this strategy is simple to implement and helps reduce peaks in the aggregated load of a country, it does not support grid operators in solving unforeseen imbalances or managing congestion. Therefore, only relying on this approach does not seem sustainable.

The second strategy is the imbalance market. Actors can receive payments through the imbalance market for shifting their flexible consumption to moments with power oversupply. This strategy is already implemented by energy-intensive industries [6]. Since the imbalance market is location-independent, this strategy could result in or worsen congestion issues.

Congestion services are the third strategy to monetize flexibility. Aggregators and consumers can receive compensation from the TSO or DSO for lowering their consumption during congestion events. As any congestion service is combined with an opposite power modification outside the congested area, this strategy will not create new imbalance issues. Aggregators could receive payments by actively participating in the congestion market. However, they could also eradicate the need to participate in bidding events by concluding a capacity limitation contract.

1.3. Electric vehicles

EVs have an electric motor and a means of energy storage, usually a battery or a fuel cell. However, the adoption of Fuel Cell Electric Vehicles (FCEVs) is negligible compared to battery EVs [2]. Battery EVs can be distinguished between Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs). PHEVs have an Internal Combustion Engine (ICE), a gas tank, and a smaller battery than BEVs. This study focuses on BEVs because PHEVs still rely on fossil fuels and thus are considered transition vehicles.

1.3.1. EV batteries

In recent years, lithium-ion EV batteries have been predominantly produced with Nickel Cobalt Manganese (NMC) compositions. However, Lithium Iron Phosphate (LFP) batteries are becoming more popular. In 2023, of all global EV sales over 40% of the battery capacity was supplied by LFP, more than doubling the share from 2020 [2]. LFP batteries only require lithium as a critical mineral, making them more than 20% cheaper to produce than NMC batteries [2].

In anticipation of breakthroughs that may improve lithium-ion batteries' performance, e.g. stable compositions with silicon anodes, NCM batteries still yield the largest energy density. However, LFP batteries are safer and their average lifetime is longer. Furthermore, LFP batteries are in general more suitable for fast-charging [14] [15] [16].

1.3.2. Smart charging and V2G

In 2025, the typical practice for charging EVs is uncontrolled and as quickly as possible. Smart charging is an alternative strategy that involves managing EV charging to reduce the impact on the power grid. Besides charging EVs' batteries from the grid, EVs could potentially give energy back to the grid. This concept is known as Vehicle-to-Grid (V2G) or bi-directional charging. Currently, few EVs and chargers

have V2G capabilities. However, most Dutch EV drivers want V2G to become a legally required feature of EVs [17].

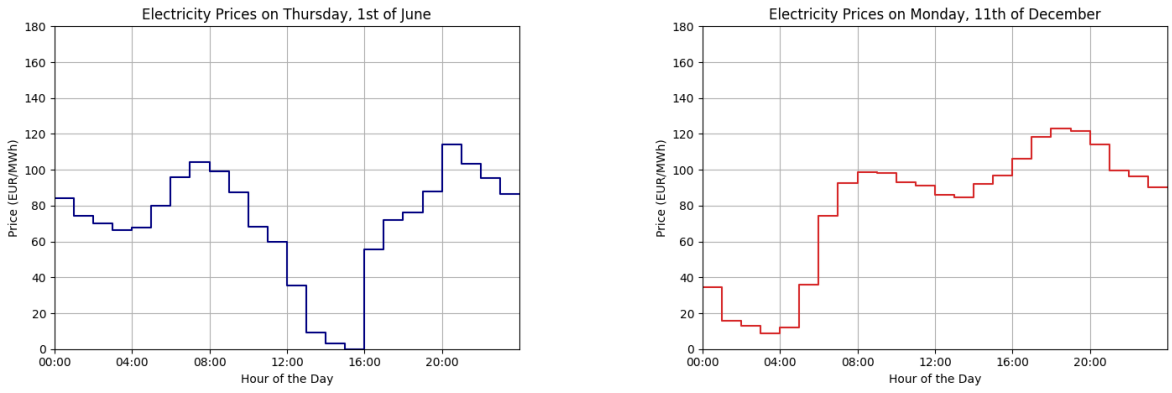
A recent study for the European Federation of Transport and Environment shows V2G-enabled smart charging has the potential to save EU energy systems €22 billion a year by 2040. Furthermore, it could reduce battery storage needs up to 92% while allowing an extra 40% of installed solar PV capacity [18]. European EV drivers using V2G at home could save 4-52% on annual electricity bills, ranging from €31 to €780 per year, excluding payment through the congestion and balancing markets [18]. However, social acceptance is an important challenge of V2G. Money is not the only motivator and range anxiety fuels the desire for EV owners to be able to control the V2G process [18].

Battery degradation is often noted as a drawback of V2G. Some studies indicate that smart charging with V2G could lead to less battery degradation than conventional As-Fast-As-Possible (AFAP) charging [18]. Furthermore, battery degradation may become less significant in the near future, as the shift toward LFP increases the cycle life of EV batteries. It remains an active research question whether the benefits of V2G outweigh challenges like social acceptance and drawbacks such as battery degradation. Although battery degradation is an important aspect, it is considered outside the scope of this research, as addressing it would complicate the optimization problem too much.

1.3.3. Charging impact on the grid

Figure 1.2 shows day-ahead electricity prices in the Netherlands for two days in 2023. The figures are created with data from the ENTSO-E open data platform [11]. The prices are related to electricity production and demand. One can observe typical peaks around 7:00 - 9:00 AM when people wake up and 5:00 - 9:00 PM when people get home from work. The electricity demand is high during these hours as people turn on the lights, cook, etc. As a result, electricity prices become higher. The low prices on the first of June during afternoon hours are probably caused by a large amount of PV power generation. Similarly, the low prices in the early morning of December 11 may indicate much wind power generation. These days again illustrate how the penetration of RES results in more electricity price volatility.

The yearly electricity demand of all EVs in the Netherlands is estimated to be 7.8 TWh by 2030 [7], while the total electricity demand of the Netherlands was approximately 110 TWh in 2022 and 2023 [19]. The impact of electric driving on the total electrical energy consumption will thus be marginal. However, as EVs are typically plugged into chargers at work around 9:00 AM and at home around 6:00 PM, by 2030 the two peaks of the typical daily load (Figure 1.2) could increase [7], resulting in even higher electricity prices and a greater risk of congestion.



(a) Thursday, 1st of June 2023

(b) Monday, 11th of December 2023

Figure 1.2: Day-ahead electricity prices of two days in the Netherlands in 2023. Created with open data from ENTSO-E [11].

1.3.4. EV aggregators and the flexibility potential

The rapid adoption of EVs means EV aggregators can have a significant impact. First, EVs consume a vast amount of electricity all year round; an average EV consumes more electricity than an average house in the Netherlands [7]. Furthermore, demand response could be implemented almost imperceptible

tibly, as EVs spend much time connected to a charger without charging [7]. Finally, EVs have large batteries; usually around 60 kWh (see Table 3.3). Therefore, especially in a small country such as the Netherlands, minor deviations in an EV battery's State Of Charge (SOC) at departure will have little consequences.

By 2030, in the Netherlands, 55% of charging is expected to be done at public chargers, 19% at home chargers, 17% at workplaces, and 9% at DC fast chargers [7]. Smart charging at home was already applied commercially in the Netherlands in 2023. Jedlix, a Dutch smart charging company, showed it could effectively reduce the charging peak of home chargers by 50% with its technology [7].

The simplicity of addressing one EV and the large connection times probably make home chargers the most practical option for smart charging. Moreover, in 2020 until 2023, most EV charging in the Netherlands occurred at private home chargers [17]. Since locality is key for congestion services and a portfolio of home chargers may well be distributed throughout the country, supplying congestion services with home chargers is more challenging. Home chargers will mostly provide flexibility at night, as EVs are typically connected to home chargers at night.

Workplace charging points can provide flexibility during the day. Furthermore, as chargers in one company parking lot are all in the same part of the low-voltage grid, fulfilling congestion services is more feasible for EV aggregators operating in workplace scenarios. The typical load peaks in the morning and afternoon, in combination with the cheapest electricity around noon when PV power generation is at its highest (Figure 1.2a), suggest there is a potential for EV aggregators to profit by implementing smart charging and V2G during the day.

Public charging perhaps has the greatest flexibility potential because of its expected charging share of 55% by 2030. Public chargers in residential neighborhoods may provide flexibility at night, while public chargers near offices may provide flexibility during the day. However, the implementation of smart charging at public chargers is more difficult. The large variation in arrival and departure times at public charging points intensifies the uncertainty [7]. Furthermore, as anyone can connect to public chargers, smart charging algorithms may be less effective due to a lack of trends in charging behavior.

1.4. Research objectives

While the potential to provide flexibility with EVs is evident, optimizing real-time charging control can be challenging. This thesis aims to investigate and develop state-of-the-art algorithms to maximize the profits of an EV aggregator that coordinates the charging of V2G-enabled EVs in a business building parking lot. This study assumes that an EV aggregator does not model or monitor the grid. Instead, its algorithms use dynamic electricity prices as inputs, effectively placing the responsibility of monitoring the grid on the network operators. Furthermore, this thesis does not address how the EV aggregator prices its customers and may increase profit by adding a margin to charging costs.

The most important constraints used in the problem are the transformer power limit and EV user preferences. In most related articles the EV behavior is modeled unrealistically and charging efficiency is assumed to be constant, while this efficiency varies for different currents. In this thesis, EV behavior is based on real-world data. Furthermore, current-dependent charge and discharge efficiencies are considered. While RL is a promising algorithm for complicated optimization problems with high uncertainty, it often does not ensure that constraints are satisfied. In the problem formulation of this thesis, constraint satisfaction is crucial to prevent unhappy EV users or damaging transformer overloads. Safe RL aims to improve constraint satisfaction and the most recent Safe RL algorithms have yet to be applied to real-time EV charging control. The contributions of this thesis can be summarized as follows:

- **Safe RL for EV charging.** Apply the latest Safe RL algorithms to optimize V2G-enabled EV charging control.
- **Realistic scenario.** Address many constraints and EV behavior based on measured data.
- **Charging efficiency.** Consider current-dependent charge and discharge efficiencies.

The main research question is:

How can an EV aggregator's charging and V2G profits be maximized using Reinforcement Learning, considering transformer limits, EV user preferences, current-dependent charging efficiencies, and uncertainty?

Furthermore, the following subquestions are addressed:

1. How to model the transformer limit and EV user constraints?
2. How to model current-dependent charging and discharging efficiencies?
3. How to define the profit maximization problem as a Constrained Markov Decision Process (CMDP)?
4. How does the proposed method perform compared to baseline methods in experiments?

1.5. Thesis outline

In Chapter 2, the EV charging optimization problem is introduced and the literature review on related articles is given. In Chapter 3, the specific problem of this thesis is formulated, the simulation environment is presented, and the proposed method and baseline methods are described. In Chapter 4, the proposed methods are verified with simulations. Finally, the conclusions and limitations are discussed in Chapter 5.

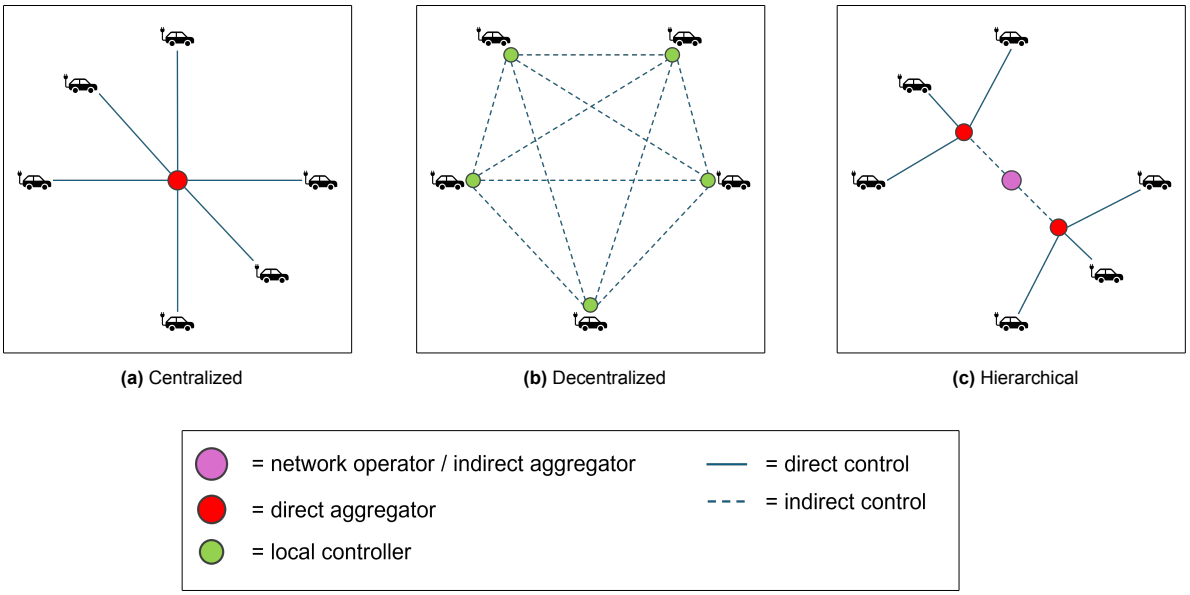
2

Coordinated EV charging: literature review

The coordination of EV charging can result in complex optimization problems. In recent years, many scenarios, algorithms, and constraints have been researched. This chapter introduces the charging optimization problem and summarizes recent related literature. First, Section 2.1 explains different charging control strategies. Then, Section 2.2 introduces the charging optimization problem. The discussed algorithms are grouped in metaheuristic methods (Section 2.3), mathematical optimization (Section 2.4), and Reinforcement Learning (Section 2.5).

2.1. EV charging control schemes

There are three main strategies to control the charging behavior of EVs, as shown in Figure 2.1: centralized, decentralized, and hierarchical control [20] [21]. In the centralized scheme, one central entity collects the inputs and lets its algorithm direct the charging. In the decentralized scheme, each EV (user) decides when to charge or discharge without a central controller. Hierarchical or hybrid centralized-decentralized control places EVs, aggregators, and possibly network operators in a tree structure [21].



(a) Centralized **(b)** Decentralized **(c)** Hierarchical

Figure 2.1: Charging control of EVs can occur in several charging schemes. Based on Figure 3 from [21].

Each strategy has its benefits and drawbacks. Centralized control usually finds better charging schedules [20] [21]. However, centralized systems are less robust as one error will lead to a failing system [21]. Additionally, centralized control lacks scalability [20] [21]. Decentralized control is more robust and scalable, but struggles to create good charging schedules due to the uncertain nature of the dynamic EV charging problem [20] [21]. Furthermore, as the EVs are not controlled directly, it is difficult for an EV aggregator to implement demand response with decentralized control.

Hierarchical control combines the best of centralized and decentralized strategies. The scalability issue of centralized control is mitigated by delegating computational load and communication to multiple aggregators [21]. Each aggregator controls a group of EVs. A group may be bounded by location, like EVs chargers at a parking lot [21]. Because aggregators control the charging of their groups of EVs in a centralized manner, results will be better than with decentralized control. Also, it will be feasible to implement demand response. In recent literature, most authors focus on hierarchical control [20] [21].

2.2. Coordinated EV charging optimization

Coordinated EV charging optimization problems aim to optimize the charging of EVs. The problems can have one or multiple objectives and often are subject to various constraints. Typical inputs are battery SOC or battery energy, electricity prices, on-site renewable energy generation, and total charging load [20] [22]. Some articles also include power flows and node voltages. However, considering grid simulations lessens the feasibility of centralized control due to the larger computation requirements [20].

The charging optimization problem can be considered from the perspective of the grid operator, the aggregator, or the EV user [20] [23]. Typical objectives are minimizing charging costs, maximizing profits, maximizing customer satisfaction, maximizing PV self-consumption, or balancing the load profile [20] [22] [23]. Often multi-objective problems are defined, combining several of these objectives into one optimization problem. Typical constraints are EV user preferences, battery capacity, charging limits, transformer load, and node voltage deviations [20]. However, few articles consider all these constraints in one problem definition.

In reality, coordinated EV charging faces a lot of uncertainty. An EV aggregator does not know exactly when EVs will arrive, how long they will stay, and how long it will take to charge their batteries. Considering this uncertainty makes the problem more realistic but difficult to solve. The EV charging scheduling problem can be static or dynamic [24]. The static problem definition assumes that the arrival, charging, and departure times are known beforehand. Static problem definitions lead to algorithms that can only perform offline planning. Because in reality these parameters cannot be known beforehand, these models are not adequate to control the charging of EVs in real-time. Algorithms trained to solve dynamic problems are suitable for real-time control. They recalibrate many times throughout the day, iteratively processing new input data when it becomes available.

2.3. Metaheuristic methods

Metaheuristic methods are approximate solvers that are not guaranteed to find an optimal solution. Often inspired by natural phenomena, their approach applies well to non-linear, non-convex, and high-dimensional optimization problems [23]. For EV charging problems, some consequences are that metaheuristic methods scale well and may better handle scenarios with large fleets of EVs. However, they are not guaranteed to find optimal solutions and to satisfy constraints. Metaheuristic methods are more suitable for offline planning than real-time control.

In [25], an Ant-based Swarm Algorithm (ASA) and Particle Swarm Optimization (PSO) were used to reduce peak loads of EV charging. Simulation results from a scenario with 500 EVs showed ASA and PSO reduced peaks compared to uncontrolled charging with similar performance. In [24], an Artificial Bee Colony (ABC) algorithm that minimizes tardiness was implemented. The work considered single-phase charging, making the load imbalance of the three lines their main constraint. Numerical results from scenarios with 180 EVs showed their algorithm outperformed two other metaheuristic methods in most simulations. More recently, a genetic algorithm (GA) was applied in [26] to minimize the total charging cost of a logistics company while ensuring all heavy-duty EVs are fully charged. The algorithm considers dynamic energy tariffs and a limited number of chargers for the amount of EVs. Several

numerical simulations with 4 up to 10 chargers and 15 up to 30 EVs showed the proposed method reduced charging cost compared to uncontrolled charging. The authors of [27] implemented PSO and GA to minimize grid loading. They considered demand response events by decreasing the power limit of the charging station at predefined times. The charging behavior of the EVs was based on real data. In their simulation with 100 EVs the GA slightly outperformed PSO.

2.4. Mathematical optimization

Many studies address the charging optimization problem as a mathematical programming problem and use classic optimization techniques. Stochastic optimization can address the charging problem dynamically, thus without knowing EV arrival, charging, and departure times beforehand. Many consider dynamic programming (DP) the only tool to solve stochastic control optimization problems adequately [28]. However, DP suffers from "the curse of dimensionality", which describes how computations grow exponentially with the number of variables [28]. Mathematical programming is suitable for offline planning, but Model Predictive Control (MPC) is required to apply it to real-time EV charging control.

The authors of [29] applied Linear Programming (LP) to minimize the EV charging cost of 100 EVs in a residential neighborhood. The study addresses dynamic electricity prices, battery degradation, and charger and transformer power limits. The work showed the potential for cost and load peak reduction. In [30], Mixed-Integer Programming (MIP) was used to reduce charging cost and increase RES utilization in scenarios with up to 150 EVs. Constant prices for electricity from RES and the grid were assumed, the latter being slightly larger. RES utilization was increased by introducing a virtual EV that encourages connected EVs to store renewable energy for EVs that may arrive later. The authors of [31] applied Mixed-Integer Linear Programming (MILP) to maximize the profits of an off-grid EV charging station powered by PV panels and a BESS. The proposed method slightly increased the profit of the charging station. The study is partly stochastic by addressing the arrival of EVs as a distribution. In [32], an automatic demand response strategy was implemented with DP. The goal was to maximize an EV aggregator's profit while considering onsite PV power generation, EV user preferences, and grid voltage deviations. The authors proposed a dynamic price vector formation method that predicts electricity prices. Simulation results with 60 EVs showed increased profits and PV power consumption while ensuring customer satisfaction.

2.4.1. Model predictive control

Model Predictive Control (MPC) implements mathematical programming into real-time control. Being a dynamic approach that constantly reconfigures strategies toward optimal solutions, MPC can handle complex V2G-enabled EV charging problems with multiple constraints like EV user preferences and grid constraints [33] [34].

The authors of [35] proposed a stochastic MPC-based Energy Management System (EMS) for integrating PV power, EV charging, and BESS within residential complexes. The proposed EMS controls the charging and discharging of the BESS. Furthermore, the EMS can provide voltage regulation. Simulations with 100 homes under various scenarios showed that the proposed method results in higher profits and lower battery degradation than two other systems. In [36], a robust MPC algorithm was used to maximize the profits of a charging station. The decision-making was modeled by Mixed-Integer Non-Linear Programming (MINLP) problems. Dynamic energy prices, PV power generation prediction, EV charging forecasts, and grid voltages are some of the inputs used in the problems. In [34], four comprehensive MPC algorithms were implemented. They are available in the EV charging and V2G simulation tool *EV2Gym* [33], the environment used in this thesis. The algorithms aim to maximize the profit of an EV aggregator while accounting for dynamic electricity prices, inflexible loads, PV power generation, transformer power limits, demand response events, EV user preferences, and EV battery degradation. MILP models the decision-making. The same algorithms can readily be employed in various scenarios with different parameter values.

Although MPC can be used for real-time V2G-enabled EV charging control, it requires significantly more computation time when problems become more complex. Furthermore, as MPC suffers from the curse of dimensionality, computation times increase sharply when either the number or size of parameters is increased [33] [34] [37]. As a result, the scalability of MPC is limited.

2.5. Reinforcement learning

Reinforcement Learning (RL) is a subfield within Machine Learning and is an alternative technique to unsupervised and supervised learning. RL is considered a model-free method, i.e., the algorithm does not model the environment. Instead, a learning agent is set free in the environment without prior knowledge and gathers experiences by iterating through the environment. The agent is not told which actions to take, but it should learn the best strategy from past experiences [28]. One major advantage of RL is that it can break the curse of dimensionality by its approximation [28]. Deep RL algorithms use neural networks as approximate functions that optimize during training. With proper training, the neural networks can partially generalize to unseen states, increasing scalability.

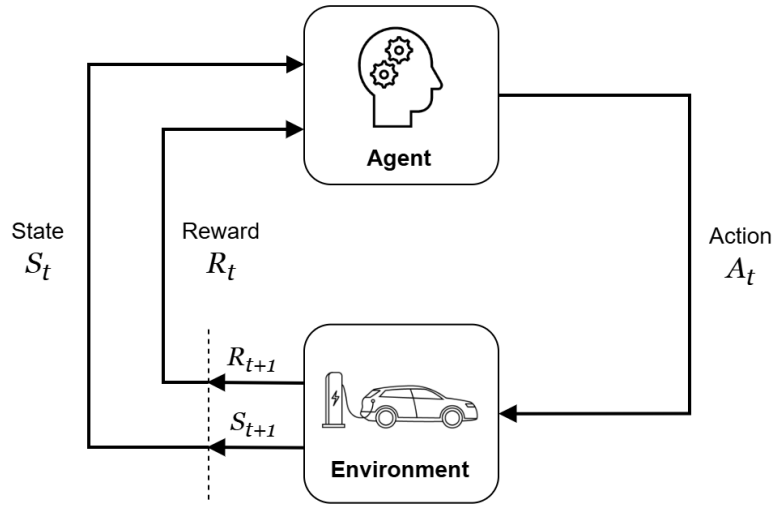


Figure 2.2: Markov Decision Process. An agent obtains new states and rewards from the environment after choosing actions.

2.5.1. Introduction

A Markov Decision Process (MDP) is an effective mathematical framework for the problem of learning from interactions [28]. In an MDP, an agent interacts with the environment at discrete time steps t = 0, 1, 2, 3 ... At each time step t, the agent sees the environment's *state* S_t and chooses an *action* A_t. Subsequently, the agent receives a *reward* R_{t+1} and finds itself in a new state S_{t+1}. Figure 2.2 shows an illustration of this process. MDPs represent a traditional approach to decision-making over time, in which actions affect both short- and long-term rewards, resulting in a trade-off [28]. The trade-off can be tuned by adjusting the *discount rate* $\gamma \in [0, 1]$. The total *return* G_t of all steps is discounted by γ:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2.1}$$

RL algorithms aim to find the best *policy* $\pi_t(a|s)$, which technically is a probability of choosing action $A_t = a$ given the agent is in state $S_t = s$ [28]. The optimal policy is thus a strategy that selects a sequence of concurrent actions that yields the agent the most reward. The agent chooses actions A_t to maximize $J(\pi)$, which is the expected return given the agent follows policy π:

$$\max_{\pi} J(\pi) = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right] \tag{2.2}$$

Additionally, most RL algorithms estimate *value functions*, to approximate the value of the current state or the value of choosing an action given the current state. This value is determined by how much future reward can be expected. State values are determined by the *state-value function* V_{π}:

$$V_\pi(s) = \mathbb{E}_\pi \left[G_t \mid s_0 = s \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_0 = s \right], \tag{2.3}$$

denoting the expected return, given the agent follows policy π and starts in state s. State-action values are determined by the *action-value function* Q_π:

$$Q_\pi(s,a) = \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_0 = s, a_0 = a \right], \tag{2.4}$$

denoting the expected return, given the agent follows policy π and starts by choosing action a from state s. RL uses value functions to improve the search for the theoretical optimal policy [28]. The theoretical optimal policy, π_*, is defined to have a larger expected return than any other policy for all states. Therefore, the optimal policy is accompanied by an optimal state-value function and an optimal action-value function [28]:

$$V_*(s) = \max_\pi V_\pi(s), \tag{2.5}$$

$$Q_*(s,a) = \max_\pi Q_\pi(s,a), \tag{2.6}$$

for all $s \in S$ and $a \in A(s)$. The optimal action-value function Q_* can be written into terms of V_* in the optimal Bellman equation:

$$Q_*(s,a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma V_*(S_{t+1}) \mid s_0 = s, a_0 = a \right]. \tag{2.7}$$

2.5.2. Challenges

Every RL implementation encounters a trade-off of *exploitation* versus *exploration*. To optimize the policy, an agent prefers to be 'greedy', i.e. choose actions that provide a good reward. However, the agent should also try unseen actions to find new valuable states and actions. The agent thus has to exploit what it has experienced to maximize total return, but simultaneously has to explore new actions [28]. The $\epsilon - greedy$ method is an example of dealing with this trade-off. This method chooses random unseen actions with a small probability ϵ, usually smaller than 0.1. Greedy actions are selected with the probability $1 - \epsilon$, the majority of chosen actions.

A second option that improves exploration is the *off-policy* search. It is a key characteristic of RL algorithms whether they use on-policy or off-policy methods. On-policy means one policy simultaneously has to exploit and explore. The $\epsilon - greedy$ method is an example of an on-policy search. Off-policy algorithms have two policies. The *behavior policy* focuses on exploring new valuable states and actions, the *target policy* on improving the final policy as much as possible. On-policy searches are simpler to implement, while off-policy searches often generalize better [28].

Another challenge that often arises is the problem of *sparse rewards*. Delivering non-zero rewards frequently enough to steer the agent in the right direction can be challenging even for simple goals [28]. Typically, only long, specific sequences of actions will result in large rewards. If the agent chooses a wrong action somewhere before the end of the sequence, it will correlate the full sequence to low rewards or even penalties. However, many actions from the sequence may have been in the right direction. This is one of the reasons why RL usually needs huge numbers of iterations to converge.

To mitigate the problem of sparse rewards and to increase learning speed, implementing RL often includes *reward shaping*. The design of the reward function is usually a trial-and-error process until results are satisfying [28]. The need for custom reward shaping for different problems is a time-consuming effort and an often-mentioned drawback of RL.

Finally, *safety* is a challenge for RL. Typically, some constraints cannot be enforced because this would reduce the learning flexibility of the RL agents too much. Agents are allowed to violate these constraints

and have to learn not to violate them from reward subtractions. Consequently, obtaining a 'safe' policy that guarantees constraint satisfaction is difficult. Furthermore, the approximation of deep RL can lead to unexpected behavior. An unseen state close to a very valuable state does not necessarily have to be valuable and may even result in a penalty when constraint limits are exceeded.

2.6. Reinforcement Learning for coordinated EV charging

Much research has been done on Deep RL applied to EV charging optimization. One advantage is that RL can be readily employed for real-time charging. Another is that its approximation can handle the uncertain nature of the dynamic EV charging optimization problem without experiencing the curse of dimensionality. This section summarizes related articles using RL to solve the problem. The articles have been divided into three segments: Classic Deep RL, Multi-Agent RL (MARL), and Safe RL.

2.6.1. Classic Deep RL

Deep RL algorithms implement techniques to improve the learning of the agent. For example, many use an experience replay buffer to store experiences from off-policy searches. This reduces the likelihood of forgetting knowledge from old transitions, thus increasing stability. In recent years, many new methods have been proposed, each aiming to outperform the others.

The authors of [38] proposed using a Deep Q-Network (DQN) to minimize the charging cost of one EV at home, considering battery degradation and V2G. Their algorithm uses a Long Short-Term Memory (LSTM) network to process trends from previous dynamic electricity prices. Simulations showed the method outperforms benchmark MPC algorithms. A similar LSTM-based network was used in [39]. Additionally, two replay buffers were implemented to mitigate the result of sparse rewards. The authors used a Deep Deterministic Policy Gradient (DDPG) algorithm to minimize charging cost while satisfying users' target SOC. Simulations showed the proposed method outperforms DQN and regular DDPG. The authors of [37] applied double DQN (DDQN), parametrized DQN (PDQN), and DDPG to an EV charging problem with one charger. The goal was to increase PV self-consumption and SOC at departure. The algorithms' performance was compared against several benchmarks, including MPC. Simulations showed the MPC algorithms slightly outperformed the RL algorithms. In [40], a custom deep policy gradient algorithm based on Proximal Policy Optimization (PPO) was proposed. The aim was to flatten the load profile of the charging EVs. In the proposed method invalid actions outside the viable action space are penalized. The superiority of this strategy compared to a related method that adjusts invalid actions to the nearest viable action was proven in simulations with 15 and 20 EVs.

2.6.2. Multi-agent RL

As the name implies, MARL considers multiple agents instead of one. In the coordinated EV charging problem, MARL could be implemented by controlling every charger with a separate agent instead of having one agent address all chargers simultaneously. For the EV charging problem of this thesis, customer satisfaction may be improved with MARL compared to single-agent RL. Since each MARL agent controls a single charging session, user preferences are easier to address. However, it would probably be more difficult to respect the transformer's power limit because the total power consumption is now a result of the actions of multiple agents instead of one.

In [41], MultiAgent Selfish-COllaborative (MASCO) was proposed, closely related to Distributed W-Learning (DWL). The objectives were to maximize battery SOC, minimize charging cost, and minimize transformer overload. In addition to DWL, an adjustable preference vector implements EV user preferences. Simulations with 30 chargers showed better performance than DWL. The algorithm of [42] addresses the same three objectives. The algorithm was based on the actor-critic method and a Communication neural Network (CommNet). Numerical experiments showed increased scalability compared to DQN. The authors of [43] applied Multi-Agent DDPG (MADDPG) and Multi-Agent DQN (MADQN) to minimize the charging cost of a charging station. Furthermore, the study included fairness of charging, battery degradation, and on-site PV power generation. A noisy network was employed to increase training speed.

2.6.3. Safe RL

Safe RL is a subfield that aims to improve constraint satisfaction. Safe RL often employs an additional cost function that generates a cost based on the severity of constraint violations. This detachment from the reward function allows for stricter constraint enforcement. According to [44], three types of safe RL are primarily used. The first type is Lagrangian-based and also known as primal-dual policy optimization. The second is a trust-region-based method. Constrained Policy Optimization (CPO) [45] is a traditional algorithm of this type. The third type includes external knowledge as shields.

In [46], the same authors of [38] extended their work by implementing CPO. In their new work [46], they no longer included battery degradation. Simulations showed CPO yielded adequate charging cost while sharply improving constraint satisfaction. The authors of [47] proposed AL-SAC: an algorithm that combines off-policy soft-actor-critic (SAC) with an augmented Lagrangian method. The proposed method achieved the lowest charging cost compared to many baseline algorithms, including CPO. Furthermore, the authors demonstrated a huge increase in learning speed compared to CPO, mainly because AL-SAC is off-policy while CPO is on-policy. Of all examined studies, the content of [46] and [47] align best with this thesis. In [44], an EV charging problem was formulated as a partially observable CMDP (PO-CMDP). The work also modeled node voltages and power flows of the distribution grid. Their Multi-Agent PPO (MAPPO) algorithm is forced to prioritize safe actions by local and global shields. Simulations showed the proposed method outperformed CPO and other safe RL algorithms.

2.6.4. Overview of related articles

In Table 2.1, related articles that use RL are summarized. Note that constraints embedded in the environment to make it work properly, such as the maximum battery capacity, are omitted from the constraints column. Furthermore, it is assumed that every article addresses a target constraint for the battery level at departure, either in terms of SOC or energy. Therefore, this target constraint is also not included in the constraints column of Table 2.1.

RL is a promising method for real-time coordinated EV charging because it can handle the large uncertainties inherent to the problem and has fast real-time execution. However, it is difficult for RL agents to learn charging behavior that complies with constraints. Safe RL is a promising extension of RL that improves constraint satisfaction. Although some traditional safe RL algorithms have been applied to EV charging optimization problems, to the author's best knowledge, there are no studies that use the newest safe RL algorithms in a problem setting similar to this thesis.

Furthermore, many related articles make unrealistic assumptions in their environment models. The arrival time, departure time, and SOC at arrival are often sampled from simple distributions that are not based on real measurements. Furthermore, most related articles consider only one EV model with one EV battery size. Finally, the charging and discharging efficiency is usually assumed to be constant. This thesis makes the environment model more realistic by sampling the EV user behavior from distributions based on real measurements. Moreover, this thesis considers multiple EV models with different battery sizes and current-dependent charging efficiencies based on real measurements for each EV model.

Reference	Objective	Charger Type	n Chargers	Method	States	Constraints	V2G
Wan et al., 2019 [38]	Min cost	Residential	1	DQN	SOC, past 24h electricity prices	Battery degradation, range anxiety	Yes
Silva et al., 2019 [41]	Max SOC, min cost, min transformer overload	Residential	30	MASCO	SOC, transformer load	Transformer power limit	No
Wan et al., 2020 [46]	Max profit EV owners	Residential	1	CPO	SOC, past 24h electricity prices	-	Yes
Zhang et al., 2020 [39]	Max profit EV owners	Public	1	DQN, DDPG, custom DDPG	Residual energy demand, previous electricity prices	-	Yes
Dorokhova et al., 2021 [37]	Max PV self-consumption	Residential	1	DDQN, DDPG, PDQN	SOC, PV generation, total load	-	No
Zhang et al., 2022 [42]	Max SOC, min cost, min transformer overload	Residential	6-60	Custom MARL	SOC, load of EV, past 24h electricity prices	Transformer power limit	No
Jiang et al., 2022 [40]	Flatten the load profile	Workplace	15, 20	Custom algorithm based on PPO	Residual energy demand	-	No
Chen et al., 2022 [47]	Min cost	Residential	1	MPC, DDPG, SAC, CPO, AL-SAC	SOC, past 24h electricity prices	Transformer power limit	Yes
Fan et al., 2023 [43]	Min cost	Not Specified	20	MADDPG, MADQN	Residual energy demand, electricity prices, PV oversupply	Fairness, battery degradation	Yes
Guan et al., 2024 [48]	Min cost	Not Specified	10	MAPPO with safety shields	SOC, PV power, electricity prices	Node voltage deviations	Yes

Table 2.1: Overview of Related Articles that use Reinforcement Learning to optimize Coordinated EV Charging.

3

Methodology

This chapter presents the optimization problem addressed in this thesis in Sections 3.1 and 3.2. The modeling of EV behavior, current-dependent charging efficiency, and PV & inflexible loads are described in Sections 3.3, 3.4, and 3.5. The MDP and the constrained MDP (CMDP) are defined in Sections 3.6 and 3.8. Finally, baseline algorithms and the proposed method are discussed in Sections 3.7 and 3.9.

3.1. Problem formulation

An EV aggregator is responsible for coordinating the charging and discharging of a group of V2G-enabled EVs at a business parking lot. Each charging station consists of a single-port charger, simplifying the electrical modeling by avoiding the need to account for port power limitations. The chargers are connected to a low-voltage grid that facilitates the power distribution between the chargers, inflexible loads, PV panels, and one transformer. The aggregator's objective is to maximize profits while ensuring EV user preferences are met and the power limit of the transformer is not exceeded. It is assumed that, upon arrival, each EV user shares their intended departure time and desired SOC for departure. The actual SOC of each EV is known during charging. Additionally, the aggregator has access to dynamic electricity prices, as well as forecasts of PV power generation and inflexible load demand. Historical data is used to train an RL agent, which is then deployed for real-time charging control.

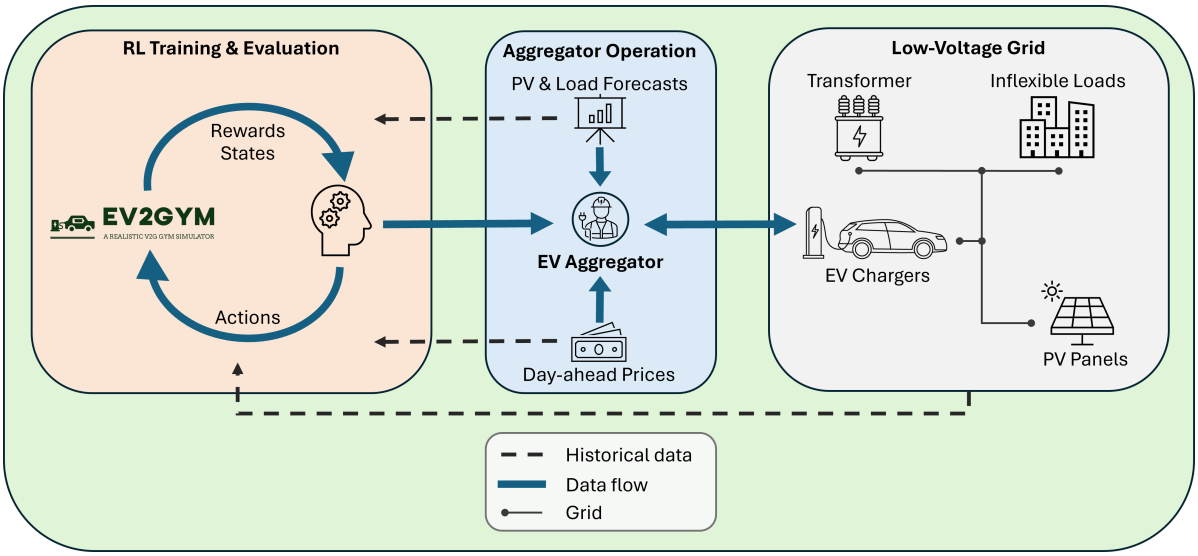


Figure 3.1: Problem overview: An EV aggregator uses RL for charging control of V2G-enabled EVs at a workplace parking lot.

The problem is modeled in the simulation environment *EV2Gym* [33], a framework implemented in Python that offers a wide range of simulation settings and baseline algorithms for EV charging research. *EV2Gym* is selected primarily because of its compatibility with RL implementations, support for V2G functionality, and its realistic modeling of EV behavior based on empirical measurement data. Several extensions and modifications have been made to the environment for this study. First, the electricity price dataset is expanded to include hourly prices from 2023 and 2024 using the ENTSO-E open data [11]. Second, the collection of EVs is revised to the ten most commonly registered EV models in the Netherlands as of December 2024. Third, the functionality for current-dependent charging efficiencies is implemented. Finally, the option to train and evaluate Safe RL algorithms is added. Figure 3.1 provides an overview of the setup, where an EV aggregator uses RL algorithms trained within *EV2Gym* to control EV charging and discharging at a workplace parking lot, subject to transformer power constraints, inflexible load consumption, and on-site PV power generation.

Table 3.1 lists key parameters of *EV2Gym* and the corresponding values used in this thesis. The maximum three-phase charging power and current are assumed to be 11 kW and 16A, respectively, following the standard of ElaadNL, a Dutch EV research organization [7]. Although V2G functionality is not yet widely deployed in three-phase AC chargers, it is expected to become a standard feature in the near future. Therefore, the discharging is also modeled as three-phase, with power and current limits equal to those of charging.

Model	**Input Parameters**	**Symbol**	**Value**
Simulation	Timescale	Δt	15 minutes
	Simulation Length	T	60 steps - 15 hours
	EV Properties		(Table 3.3)
	Scenario (Residential, Workplace, or Public)		Workplace
EV	Time of Arrival & Time of Departure	$t^{\text{arr}}, t^{\text{dep}}$	[5:00 am - 8:00 pm]
	Min. Time of Stay		120 minutes
	Min. & Max. Charging Power (kW)	$\underline{P}_{ch}^{AC}, \overline{P}_{ch}^{AC}$	0, 11
	Min. & Max. Discharging Power (kW)	$\underline{P}_{dis}^{AC}, \overline{P}_{dis}^{AC}$	0, 11
	Min. Battery Capacity (kWh)	$\underline{E}$	5
	Max. Battery Capacity (kWh)	$\overline{E}$	(Table 3.3)
	Battery SOC at Arrival	SOC^{arr}	(Figure 3.5)
	Target SOC for Departure	SOC^{*}	100%
	Min. V2G SOC	$SOC^{\text{min,V2G}}$	50%
	Charge & Discharge Efficiency	$\eta^{\text{ch}}, \eta^{\text{dis}}$	(Figure 3.7)
	Binary EV Coefficient	$u_{i,t}$	0, 1
Charging Point	Min. & Max. Charging Current (A)	$\underline{I}^{\text{ch}}, \overline{I}^{\text{ch}}$	0, 16
	Min. & Max. Discharging Current (A)	$\underline{I}^{dis}, \overline{I}^{dis}$	0, 16
	Voltage (V) & Phases	V, ϕ	230, 3
	Type of Chargers (AC or DC)		AC
	Charging & Discharging Prices (€/kWh)	$c^{\text{ch}} = c^{\text{dis}}$	
	Binary Charging Variable	$\omega_{i,t}^{\text{ch}}$	0, 1
	Binary Discharging Variable	$\omega_{i,t}^{\text{dis}}$	0, 1
Transformer	Min. & Max. Transformer Power (kW)	$\underline{P}_{T}^{tr} = -\overline{P}_{T}^{\text{tr}}$	90
	Inflexible Loads (kW)	P^{L}	
	PV Power Generation (kW)	P^{PV}	
	Demand Response Event (kW)	P^{DR}	
	Set of Connected Charging Points	C	10, 30

Table 3.1: Key parameters of *EV2Gym* with corresponding symbols and values used in this thesis.

3.2. Objective function and constraints

The objective function and most constraints in this work are obtained from the profit maximization problem formulation introduced in *EV2Gym* [33]. The goal is to maximize the EV aggregator's profits under dynamic charging & discharging prices $c^{\text{ch}}, c^{\text{dis}}$ by choosing suitable charging and discharging actions for all connected EVs at time step t:

$$\max_{I^{\text{ch}}_{i,t}, I^{\text{dis}}_{i,t}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{C}} \left(P^{\text{ch}}_{i,t} \cdot c^{\text{ch}}_{i,t} + P^{\text{dis}}_{i,t} \cdot c^{\text{dis}}_{i,t} \right) \cdot dt, \tag{3.1}$$

subject to soft constraints (3.2) and (3.3), and hard constraints (3.4) - (3.18).

Table 3.2 shows the classification of constraints between hard constraints and soft constraints. Hard constraints are embedded in the environment and cannot be violated. For instance, the SOC of each EV will always be within its minimum and maximum bounds, as defined in (3.8). The soft constraints are not strictly enforced, but RL agents should learn not to violate them from reward subtractions or constraint violation costs. The EV target SOC for departure (3.2) and the transformer power limit (3.3) are treated as soft constraints. The soft constraints can also be interpreted as objectives, giving RL agents a multi-objective optimization problem: the agent must not only maximize profits but also maximize EV user satisfaction and minimize transformer overloads. The soft constraints could be added as components to the objective function, through Lagrangian relaxation, for example. This is unnecessary for hard constraints since they cannot be violated.

The minimum V2G SOC constraint (3.18) was added to the default constraint set from *EV2Gym* [33]. Without this constraint, an agent may discharge an EV to its technical minimum SOC. If an EV user unexpectedly needs their EV before the set departure time–for instance, if an emergency happens–a near-empty battery would be undesirable. Constraint (3.18) ensures that EVs are never discharged below the minimum V2G SOC, thus eliminating this risk. The minimum V2G SOC is set at 50% in this thesis. This constraint has several implications. On the one hand, it helps reduce battery degradation by limiting the depth of discharge. On the other hand, it restricts the agent's flexibility in selecting discharging actions, which may hinder its ability to fully explore the potential benefits of V2G. To simplify the optimization problem, the minimum V2G SOC was implemented as a hard constraint.

Soft Constraint	Hard constraint
(3.2), (3.3)	(3.4) - (3.18)

Table 3.2: Classification of constraints between hard and soft constraints.

Soft constraints [1]

At the time of departure t_{dep}, the battery SOC should be larger than or equal to the target SOC for departure:

$$SOC_{i,t} \geq SOC^{*}_{i,t} \quad \forall \forall i, \forall t | t = t^{\text{dep}}_{i,t} \tag{3.2}$$

The total power being consumed or generated should not exceed the limits of the transformer:

$$\underline{P}^{\text{tr}}_{t} \leq P^{\text{EVs}}_{t} + P^{L}_{t} + P^{\text{PV}}_{t} \leq P^{\text{tr}}_{t} - P^{\text{DR}}_{t} \quad \forall t \tag{3.3}$$

Hard constraints

The charging power consumed from each charging point i at each time step t is calculated by:

$$P^{\text{ch}}_{i,t} = I^{\text{ch}}_{i,t} \cdot V \cdot \sqrt{\phi} \cdot \eta^{\text{ch}}_{i,t} \cdot \omega^{\text{ch}}_{i,t} \quad \forall i, \forall t \tag{3.4}$$

The discharging power injected into each charging point i at each time step t is calculated by:

[1] Soft constraints are not strictly enforced, but violations are incorporated in the RL process through costs (Safe RL) or reward subtractions (Classic RL).

$$P_{i,t}^{\text{dis}} = I_{i,t}^{\text{dis}} \cdot V \cdot \sqrt{\phi} \cdot \eta_{i,t}^{\text{dis}} \cdot \omega_{i,t}^{\text{dis}} \quad \forall i, \forall t \tag{3.5}$$

The charging efficiency is a function of the charging current $I_{i,t}^{\text{ch}}$ as per Figure 3.7:

$$\eta_{i,t}^{\text{ch}} = f(I_{i,t}^{\text{ch}}) \tag{3.6}$$

Again, for the discharging efficiency:

$$\eta_{i,t}^{\text{dis}} = f(I_{i,t}^{\text{dis}}) \tag{3.7}$$

The battery capacity of each EV connected to charging point i is bounded by the lower and upper capacity limits of that EV:

$$\underline{E}_i \leq E_{i,t} \leq \overline{E}_i \quad \forall i, \forall t \tag{3.8}$$

The battery capacity of each EV connected to charging point i at time step t is calculated by:

$$E_{i,t} = E_{i,t-1} + \left(P_{i,t}^{\text{ch}} + P_{i,t}^{\text{dis}}\right) \cdot dt \quad \forall i, \forall t \tag{3.9}$$

At the time of arrival t^{arr}, the battery capacity of each EV is set equal to E^{arr}:

$$E_{i,t} = E_{i,t}^{\text{arr}} \quad \forall i, \forall t | t = t_{i,t}^{\text{arr}} \tag{3.10}$$

The charging current at each charging point i is bounded by the lower and upper charging current limits:

$$\underline{I}_{i,t}^{\text{ch}} \leq I_i^{\text{ch}} \leq \overline{I}_i^{\text{ch}} \quad \forall i, \forall t \tag{3.11}$$

The discharging current at each charging point i is bounded by the lower and upper discharging current limits:

$$\underline{I}_{i,t}^{\text{dis}} \geq I_i^{\text{dis}} \geq \overline{I}_i^{\text{dis}} \quad \forall i, \forall t \tag{3.12}$$

The binary variables $\omega_{i,t}^{\text{ch}}$ and $\omega_{i,t}^{\text{dis}}$ make sure charging and discharging can not happen simultaneously at one charging point:

$$\omega_{i,t}^{\text{ch}} + \omega_{i,t}^{\text{dis}} \leq 1 \quad \forall i, \forall t \tag{3.13}$$

$$\omega_{i,t}^{\text{ch}} = \omega_{i,t}^{\text{dis}} = 0 \quad \forall i, \forall t | u_{i,t} = 0 \tag{3.14}$$

$$\omega_{i,t}^{\text{ch}}, \omega_{i,t}^{\text{dis}} \in \{0, 1\} \tag{3.15}$$

$$I_{i,t}^{\text{cs}} = \left(I_{i,t}^{\text{ch}} \cdot \omega_{i,t}^{\text{ch}} + I_{i,t}^{\text{dis}} \cdot \omega_{i,t}^{\text{dis}}\right) \quad \forall i, \forall t \tag{3.16}$$

The total power consumed from or injected into the EVs is calculated by:

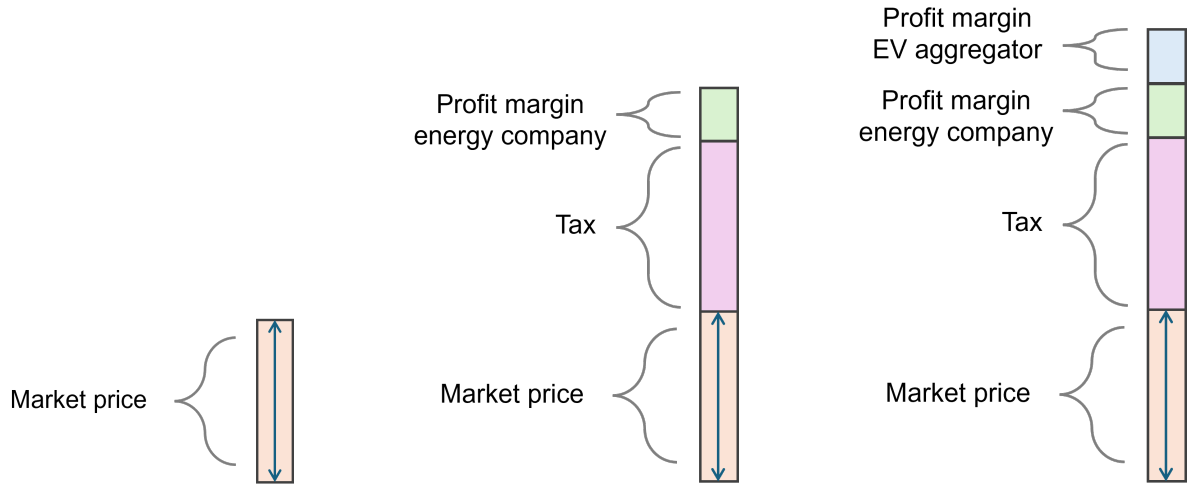
$$P_t^{\text{EVs}} = \sum_{i \in C} \left(P_{i,t}^{\text{ch}} + P_{i,t}^{\text{dis}}\right) \quad \forall i, \forall t \tag{3.17}$$

The battery SOC must be above the minimum V2G SOC before the EV is allowed to discharge:

$$SOC_{i,t} > SOC_{i,t}^{\text{min,V2G}} \quad \forall i, \forall t | \omega_{i,t}^{\text{dis}} = 1 \tag{3.18}$$

3.2.1. EV aggregator profit

The objective function (Equation 3.1) is formulated from the perspective of the EV aggregator. The aggregator aims to increase profits by applying charging schemes that leverage price valleys in dynamic electricity prices. In this thesis, the dynamic electricity prices are based on day-ahead market prices and exclude taxes or profit margins. Furthermore, both EV charging and discharging prices are set equal to these prices. These modeling choices reflect the assumptions that tax policies vary across countries and that the tax rates for purchasing and selling electricity are approximately equal. Figure 3.2 illustrates the difference between the electricity prices used in this thesis (Figure 3.2a), those charged by energy companies in practice (Figure 3.2b) and those passed on to EV users by the aggregator (Figure 3.2c). According to the objective function defined in Equation 3.1, the EV aggregator incurs charging costs equal to market prices multiplied by charged energy and earns revenue equal to market prices multiplied by discharged energy. This formulation allows RL agents to increase profit by charging EVs less or discharging them more.



(a) This thesis **(b)** Charge by energy companies **(c)** Charged by the EV aggregator

Figure 3.2: Dynamic electricity prices

In reality, however, the aggregator's economic incentives are different, as the EV users would typically bear the cost of charging. The aggregator would pass on the charging cost and discharging profit to the EV users, potentially adding a small margin to the cost or subtracting a small margin from the profit. In such a setting, charging EVs less would reduce–rather than increase–the aggregator's profit, as less revenue would be collected from margins on charging transactions. This discrepancy is a limitation of the simplified problem formulation of this thesis. In future work, this limitation could be overcome by considering more realistic aggregator revenue models that incorporate pricing strategies. This issue is discussed more in Section 5.2. Until such problem formulations are adopted, the results obtained under the current objective function should be carefully interpreted: policies that increase profit by undercharging EVs may not reflect desirable charging behavior for aggregators in practice.

3.3. EV user behavior

Uncertainty plays a central role in real-time EV charging control. In *EV2Gym*, several parameter values are sampled from distributions of ElaadNL to make simulations more realistic. These distributions are created from measured EV charging sessions in the Netherlands and are publicly available [49]. Each EV transaction is initialized with the following parameters: arrival time (t^{arr}), departure time (t^{dep}), battery SOC at arrival (SOC^{arr}), target SOC at departure (SOC^{*}), and EV type. To simplify the optimization problem, all EV users are assumed to desire a departure SOC of 100%. Table 3.3 lists the ten most frequently registered EVs in the Netherlands as of December 2024, based on data from RVO-NL [50]. The battery capacities and three-phase charging power ratings were obtained from [33]. The EV type is sampled with a probability proportional to the number of registrations in Table 3.3, giving the Tesla Model 3 the highest likelihood of being selected. The weighted mean battery capacity

of the EV types listed in Table 3.3 is 59.8 kWh. This value was computed in Python with the script `weighted_mean_EV_battery.py`, which is provided in Appendix A.3.

EV Type	Registrations	$\overline{E}$ (kWh)	$\overline{P}_{AC}^{ch}$ (kW)
Tesla Model 3	47783	57.5	11
Tesla Model Y	39216	57.5	11
Kia e-Niro	28028	64.8	11
Volkswagen ID.3	23033	58	11
Skoda Enyaq	21186	58	11
Hyundai Kona	19815	64	11
Volvo XC40	19307	66	11
Peugeot e-208	17785	46.3	7.4
Volkswagen ID.4	16449	77	11
Renault Zoe	14545	52	11

Table 3.3: Top 10 EVs in the Netherlands with battery capacity and max. three-phase charging power. Data from [33], [50].

The ElaadNL measurements were obtained between 2018 and 2020 for all electric vehicles in the total fleet of the Netherlands, thus also including PHEVs. For most of the period from 2018 to 2020, there were more PHEVs than BEVs in the Netherlands [50]. As described in Section 1.3, this thesis focuses on BEVs because PHEVs are regarded as temporary in the transition towards a fully electric fleet. PHEVs are assumed to have arrival and departure times similar to BEVs. However, the ElaadNL data is a limitation for the SOC at arrival (SOC^{arr}) because of the smaller batteries of PHEVs. Consequently, this thesis does not accurately represent a parking lot where most of the charging sessions are with BEVs. However, it can act as a starting point for further research that applies more accurate BEV data or compensates for the PHEVs in the ElaadNL data.

Figure 3.3, adopted from [33], displays the distributions of arrival and departure times for public, workplace, and residential chargers. Figure 3.3 shows that EV users usually arrive in the morning and leave in the evening at workplace chargers, while the opposite trend is observed at home chargers. Public charger behavior appears to be a mixture of the two. As this thesis focuses on workplace charging, t^{arr} and t^{dep} are sampled from the middle distribution, “Work”.

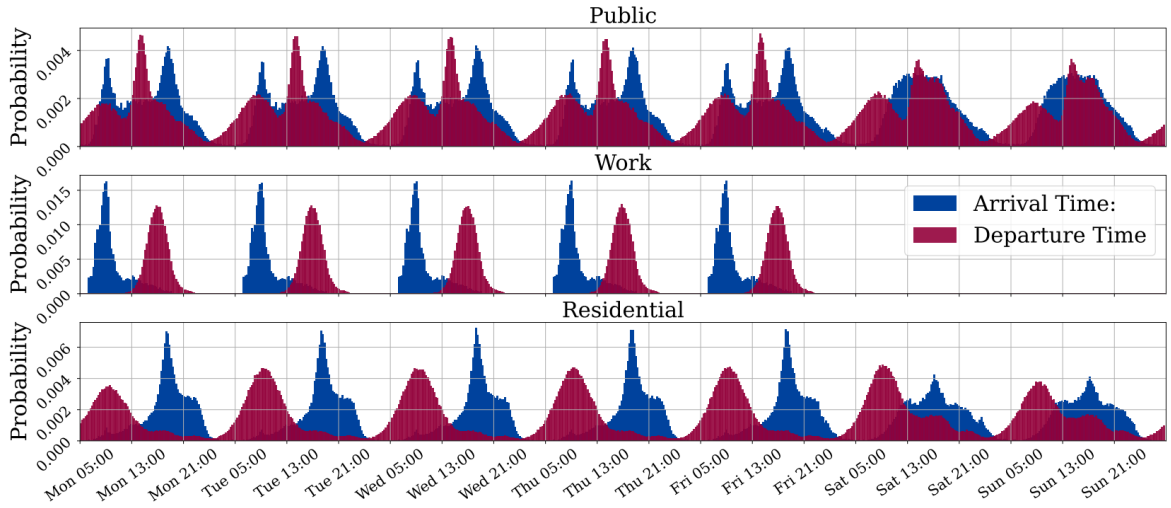


Figure 3.3: Distribution of arrival and departure times of EVs in the Netherlands. Figure obtained from [33].

The parameter `min_time_of_stay` imposes a lower bound on the duration of EV charging sessions. EVs with a stay time shorter than this threshold are assumed to be as charged as fast as possible because there is not enough time to do smart charging or V2G. In reality, these charging sessions

would be added to the inflexible loads. In *EV2Gym*, however, EVs with a shorter stay time have their departure time adjusted to ensure their duration matches `min_time_of_stay`. The default value for this parameter in the profit maximization setting of *EV2Gym* is 180 minutes. For this thesis, it was reduced to 120 minutes to increase the proportion of EV sessions with shorter stay times and better reflect the variability observed in real-world EV charging behavior.

The initial SOC of EVs (SOC^{arr}) is derived from an ElaadNL dataset containing the mean energy required at arrival to get a full battery. The dataset provides mean energy demand values for each half-hour interval between 05:00 and 18:30. As these discrete mean values are too sparse for direct sampling, *EV2Gym* converts them into distributions. For each EV arrival SOC^{arr}, the energy demand is sampled from a normal distribution $\mathcal{N}(\mu, \frac{\mu}{2})$, where μ is the mean energy demand for the nearest lower half-hour time block in the dataset. For example, an EV arriving at 9:27 would be assigned a sample from the 9:00 distribution. The sampled energy value is then converted to the corresponding SOC^{arr}. Figure 3.4 shows the probability density function for energy demand at arrival when μ = 14.87 kWh, the mean energy demand at 9:00. By default, *EV2Gym* imposes a minimum energy demand of 5 kWh. However, to avoid truncating the distribution excessively, this thesis reduces the minimum energy demand to 1 kWh.

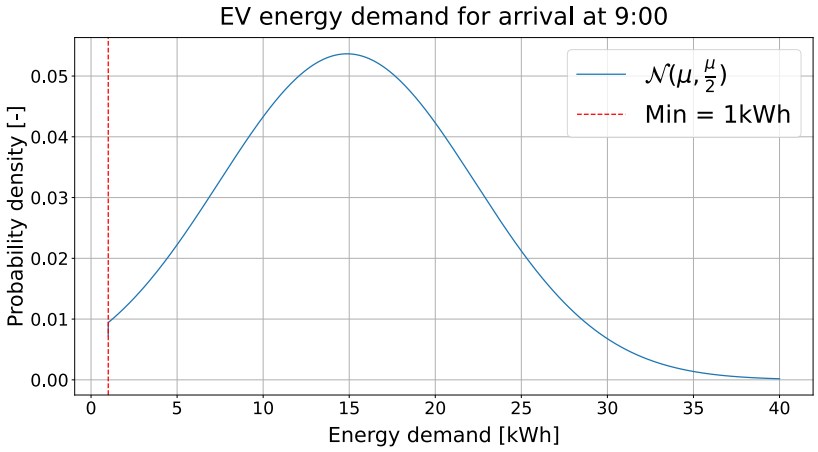


Figure 3.4: The probability density of energy demand for EV arrival at 9:00; μ = 14.87

Figure 3.5 presents the distribution of energy demand at arrival for EVs at home and workplace chargers. Figure 3.5 was created from the ElaadNL data [49]. Figure 3.5 shows that the measured energy demand upon arrival at home chargers was significantly larger than the demand at work—twice as high at the 20th percentile: 40 kWh at home compared to 20 kWh at work. This difference may come due to national charging behavior trends in the Netherlands. According to [17], around 50% of charging occurred at home in recent years, whereas only 10-20% took place at workplace chargers. This suggests that EV users who charged at work often also charged at home, resulting in lower energy demand upon arrival at the workplace. However, another possible explanation is the share of PHEVs in the ElaadNL data. If a larger part of the charging sessions at workplace chargers compared to home chargers were from PHEVs, the energy demand at arrival would become lower at workplace chargers because of the smaller batteries.

Figure 3.5 indicates that 80% of EVs arriving at workplace chargers require less than 20 kWh of energy. For the problem of thesis without PHEVs, this is equivalent to 33% SOC on average, based on the weighted mean of 59.8kWh of the collection of BEVs from Table 3.3. This implies that most EVs arrive at work with an SOC^{arr} above 67% SOC, creating a substantial opportunity for V2G at workplace chargers in the setting of this thesis, even under the minimum V2G SOC constraint of 50%. However, since the ElaadNL data contains more data of PHEVs than BEVs, this conclusion cannot be extended to workplace chargers in general. The V2G potential at a workplace parking lot where predominantly BEVs come to charge may be different.

The number of EVs arriving at the parking lot can be adjusted using the EV `spawn_multiplier` in *EV2Gym*. This parameter scales the base EV spawn probability, such that a higher value increases

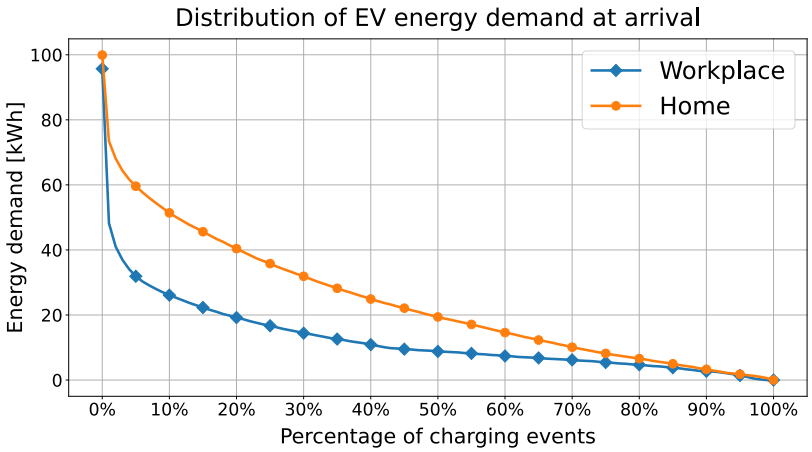
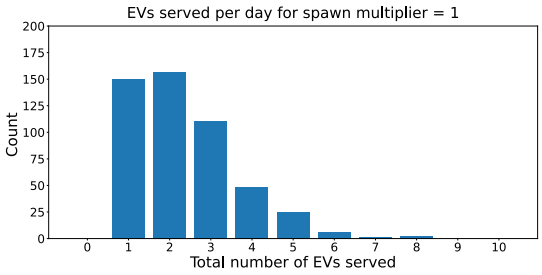
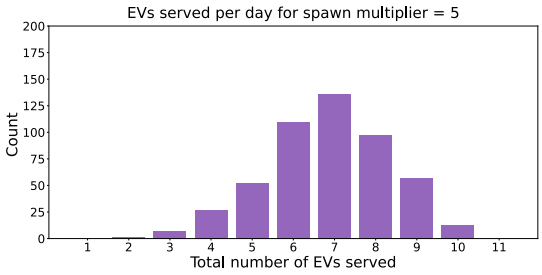


Figure 3.5: Distribution of EV energy demand at arrival in the Netherlands. Data obtained from [49].

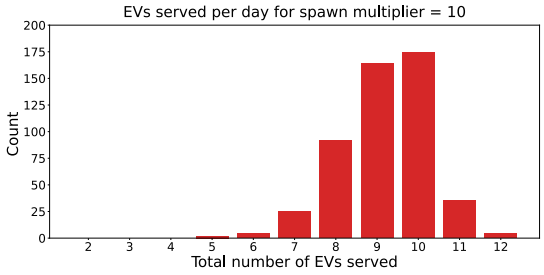
the likelihood of an EV arriving at a charger during each simulation step. Figure 3.6 compares charger occupancy levels for a spawn multiplier of 1, 5, and 10. Figure 3.6 is based on 500 simulation days in a workplace setting, with each day running from 05:00 to 20:00. For each day, the total number of EVs connecting to a charger was counted. EV spawn events are independent across chargers and time steps. As a result, increasing the simulation time window (e.g., from 09:00–17:00 to 07:00-19:00) would generally result in a higher number of daily EV arrivals, even if the `spawn_multiplier` remains constant.



(a) *Low charger occupancy level*



(b) *High charger occupancy level*



(c) *Very high charger occupancy level*

Figure 3.6: Comparison of charger occupancy level for different values of EV spawn multiplier.

Figure 3.6a shows that with a spawn multiplier of 1, usually 1-3 EVs arrive during the day. The charger occupancy level thus is defined as low for a spawn multiplier of 1. Figure 3.6b indicates that with a spawn multiplier of 5, usually 6-8 EVs arrive during the day, reflecting a high charger occupancy level. When the spawn multiplier is set to 10, usually 9-10 EVs arrive per day as shown in Figure 3.6c, thus corresponding to a very high charger occupancy level. Overall, Figure 3.6 demonstrates that the spawn multiplier serves as an effective modeling tool for different EV behavior trends in the parking lot. It is noteworthy that in Figure 3.6a there is never a day with zero EVs arriving, even though one could expect

this based on the count of only one EV arriving. Indeed, the option of no EVs arriving throughout a day is restricted in *EV2Gym*. Although there may be zero EVs arriving in reality, algorithms can only control the charging and discharging of EVs, so it is not interesting to simulate a day without any EV.

3.4. Current-dependent charging efficiency

By default, charging and discharging efficiencies are assumed to be constant in *EV2Gym*. This thesis extends the environment by adding the functionality of having current-dependent efficiencies. The implemented charging efficiencies are based on empirical measurements from [51]. The variation of the charging efficiency for different current levels is the largest for single-phase charging, where the difference between the worst and best efficiency exceeds 10% for many EVs. With three-phase charging, there is typically a 3% difference between the worst and best efficiency. Despite the smaller variation, this thesis investigates the impact of modeling the three-phase current-dependent efficiency, as even modest differences may influence the obtained profits or constraint satisfaction.

The three-phase charging efficiencies for different charging currents are obtained from [51]. When an EV type appears multiple times in [51], the most recent version is chosen. The standard range, single-motor version of the Tesla Model 3 (Tesla Model 3 SRSM) is used in this thesis. Two EV types from Table 3.3 are excluded. The Volvo XC40 is omitted because it is not included in [51]. The Renault Zoe is excluded due to its inability to charge at currents below 6A. The remaining eight EVs of Table 3.3 are included. Figure 3.7, created using data extracted from [51], shows the charging efficiencies of these eight EV types across different current levels.

In [51], the efficiencies are displayed as ranges rather than precise values. For example, the efficiency of the Peugeot e-208 for currents between 5 and 7A lies somewhere between 83% and 86%. To simplify the modeling, this thesis assigns fixed efficiency values to each color band: yellow corresponds to 84%, light green to 87%, mid green to 90%, and dark green to 93%. For currents below 5A, the same efficiency as the 5 - 7A range is assumed. These modeling assumptions are not sufficient to claim accurate modeling of current-dependent charging efficiencies. However, they are considered adequate for the exploratory purpose of this study: to evaluate whether incorporating variable charging efficiency has a meaningful impact. If significant performance differences are observed between the constant and current-dependent efficiency settings, future work should incorporate more precise modeling. Conversely, if the observed differences are minimal, this simplified approach may be sufficient to demonstrate that the added value of current-dependent efficiency modeling for three-phase charging in this context is limited.

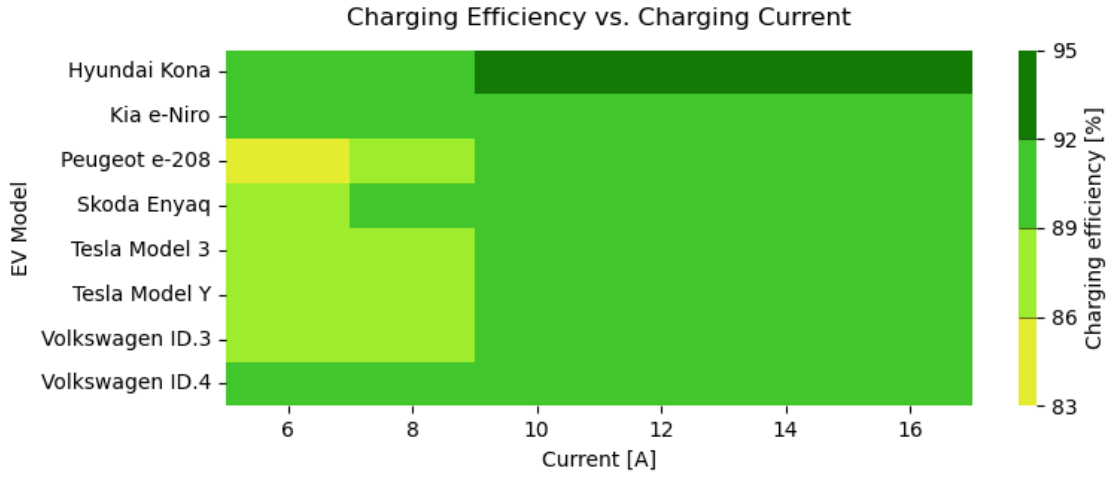


Figure 3.7: Charging efficiency versus current for the eight EVs used in the problem. Data obtained from [51].

This study assumes three-phase AC charging and discharging. However, this reflects an ideal setting as V2G functionalities have hardly been implemented in AC chargers yet. Due to the lack of empirical data, it is assumed that the three-phase AC discharging efficiencies are equal to the charging

efficiencies obtained from [51]. This assumption is partly supported by [52], an article that measured efficiency curves for a DC-DC charger under varying charging and discharging currents. The authors of [52] showed that the discharging efficiency curve was approximately the inverse of the charging efficiency curve, both showing lower efficiency at lower currents. While this data was obtained from a DC-DC charger, the similarity in the curves suggests that using symmetric charging and discharging efficiencies is a reasonable approximation for the purpose of this study.

3.5. PV power generation and inflexible loads

EV2Gym includes on-site PV power generation and inflexible load functionalities by default. The PV data is derived from the GitHub repository *renewables-ninja* [53], which provides historical PV generation data. The aggregated PV power generation in the Netherlands is scaled down to fit the power transformer limit in the simulation. The inflexible load data is based on the Pecan Street dataset [54], which offers randomized load data from real measurements of households. An EV aggregator can never have perfect forecasts of the PV or inflexible loads. To include this uncertainty, *EV2Gym* allows the forecast accuracy to be adjusted. The default mean error is 30% for the loads with a standard deviation of 5%. The default mean error is 20% for PV, also with a standard deviation of 5%. The default accuracy is not adjusted in this thesis.

The severity of the PV power generation and inflexible loads can be adjusted in the *EV2Gym* with the PV and load `capacity_multiplier_mean`. This parameter defines the expected magnitude of PV output or load demand relative to the transformer's rated power. For example, setting the mean load capacity multiplier to 0.5 yields load multiplication factors sampled from the distribution $\mathcal{N}(0.5, 0.1) \cdot \overline{P}_T^{\text{tr}}$, where $\overline{P}_T^{\text{tr}}$ is transformer's power limit.

Figure 3.8 shows the probability density function for the load multiplication factor with mean 0.5 and standard deviation 0.1. If a value of 0.5 is sampled, the Pecan Street load profile is scaled such that the peak load on that day reaches 50% of the transformer's capacity. To prevent overloads from PV or loads alone, the sampled multiplication factors are truncated at 1. To avoid them from becoming negative, they are also truncated at 0. The default distribution's standard deviation of 0.1 has been adjusted to 0.05 for the PV multiplication factor, as a small mean PV capacity multiplier is applied in this thesis. Figure 3.9 displays the probability density of the PV multiplication factor for a mean of 0.1 and standard deviation of 0.05.

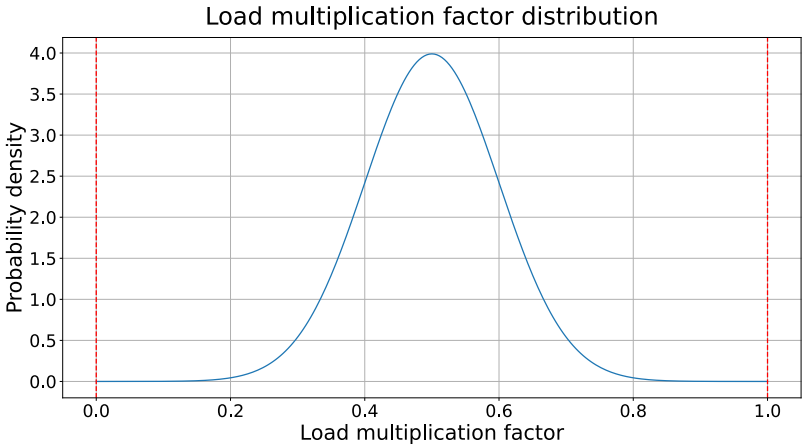


Figure 3.8: Probablity of load multiplication factor for a mean capacity multiplier of 0.5 and standard deviation equal to 0.1.

In practice, an EV aggregator operating at a business parking lot would likely encounter inflexible loads from office buildings rather than residential neighborhoods. Additionally, the variability in PV generation at a single parking lot is expected to be higher than in nationally aggregated data, where localized weather fluctuations are averaged out. Moreover, the forecast errors for both PV generation and the loads are not derived from real data. Consequently, the load and PV inputs considerably limit how realistic simulations will be. Nevertheless, in the absence of more representative data, this thesis

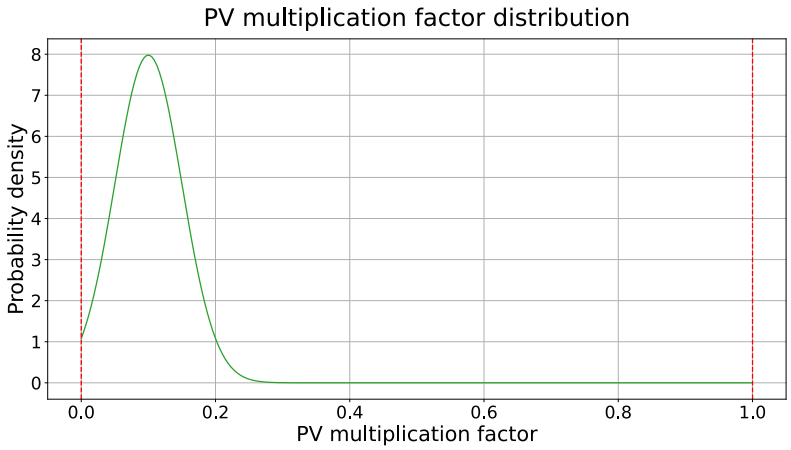


Figure 3.9: Probablity of PV multiplication factor for a mean capacity multiplier of 0.1 and standard deviation equal to 0.05.

adopts the default load and PV profiles provided by *EV2Gym*.

3.6. Markov Decision Process

The MDP used in this thesis is based on the default configuration of *EV2Gym* [33]. Through extensive simulation, the state function was iteratively updated in a mostly trial-and-error process. A key guiding concept was the normalization of input parameters, which was found to improve learning stability and efficiency. The following state function yielded the best performance in scenarios without transformer overloading:

$$\mathbf{S_t} = \left[\frac{t}{T}, c^{ch}_{t,t+h}\right] \cup \left[SOC_i, \frac{t^{dep}_i - t}{T}\right] \quad \forall i, \tag{3.19}$$

where t is the current time step, $T = 60$ is the total number of steps per episode, $h = 28$ is the electricity price horizon $c^{ch}_{t,t+h}$ is the sequence of charging and discharging prices from the current step up to h steps ahead, SOC_i is the current SOC of connected EV i, t^{dep}_i is the time step where the EV will depart. As the number of SOC_i and t^{dep}_i increases with i, the state size grows with the number of chargers modeled. In more complex scenarios that include PV, loads, and transformer overloading, the following state function proved most effective:

$$\mathbf{S_t} = \left[\frac{t}{T}, c^{ch}_{t,t+h}, \frac{P^{tot}_{t-1}}{\overline{P}^{\text{tr}}_T}, \frac{\hat{P}^{L}_{t:t+h} - \hat{P}^{PV}_{t:t+h}}{\overline{P}^{\text{tr}}_T}\right] \cup \left[SOC_i, \frac{t^{dep}_i - t}{T}\right] \quad \forall i, \tag{3.20}$$

where the horizon $h = 20$ now applies to the electricity prices as well as PV and load forecasts, P^{tot}_{t-1} is the transformer loading in the previous step, $\overline{P}^{\text{tr}}_T$ is the transformer power limit, $\hat{P}^{L}_{t:t+h}$ and $\hat{P}^{PV}_{t:t+h}$ are the load and PV power forecasts for the current step up to $h = 20$ steps ahead. The horizon of Equation 3.20 is smaller than that of Equation 3.19, as the agent's learning ability is reduced by larger horizons. The smaller horizon $h = 20$ was found to be more appropriate for the more complicated problems of scenarios related to Equation 3.20.

The latest EV chargers can provide smart charging with continuous power control [40]. As continuous power control can yield better solutions through increased flexibility, the actions of the RL agent are chosen to be continuous. The following action function is used in both experiments:

$$\mathbf{a_t} = [-1, 1] \tag{3.21}$$

The agent controls the charging power through the current. If $\mathbf{a_t} = 1$, an EV will be charged with the maximum charging current of 16A, or maximum charging power of 11kW. Conversely, $\mathbf{a_t} = -1$ represents maximum discharging at 16 A. Intermediate values yield proportionally scaled current levels.

Shaping the reward function is a critical design step in any classic RL implementation. As the reward is the only feedback in the agent's training process, the reward function should adequately incorporate all learning goals. In the optimization problem of this thesis, defined in Section 3.2, this means the reward function should direct agents toward high profit. However, agents should also learn to respect the soft constraints: the target SOC of EV users and the transformer power limit. The following reward function is the default reward function of *EV2Gym*:

$$\mathbf{r_t} = -100 \cdot \epsilon^{\text{tr}}_{t-1} + \sum_{i \in C} \pi_{i,t-1} - 100 \cdot \exp\left(-10 \cdot \epsilon^{\text{usr}}_{i,t}\right), \tag{3.22}$$

for any time step t, where $\epsilon^{\text{tr}}_{t-1}$ is the transformer overloading in kWh, π_{t-1} is the charging profit of connected EV i, and $\epsilon^{\text{usr}}_{i}$ is the user satisfaction score of EV i at departure. The user satisfaction score is determined by:

$$\epsilon^{\text{usr}} = \frac{SOC^{\text{dep}}}{SOC^{*}}, \tag{3.23}$$

the actual SOC at departure divided by the desired SOC at departure. Since all EVs in this thesis request a full battery at departure, a vehicle departing with 90% SOC yields a user score of $\epsilon^{\text{usr}} = 0.9$. The reward function of this thesis was based on the default function of *EV2Gym* and was improved in a trial-and-error process. As pointed out in [33], the default reward function of *EV2Gym* did not lead the classic RL algorithms to desirable behavior in the V2G profit maximization problem. Most algorithms showed relatively high profits but low user satisfaction. A likely cause is the user satisfaction term of the default reward function (3.22), which takes the form:

$$-a \cdot \exp(b \cdot \epsilon^{\text{usr}}). \tag{3.24}$$

Figure 3.10 plots the default user satisfaction term for all possible values of the user score. Figure 3.10 reveals that agents only incur a substantial penalty when the departure SOC of an EV is below 40%. To improve the agent's sensitivity to insufficient charging, two adjustments were made to the default user satisfaction term. First, the value of b was increased from -10 to -3, flattening the curve to increase the penalty for moderate user scores. Second, the value of a was increased from 100 to 1000, amplifying the overall magnitude of the penalty. After extensive experimentation, the following revised reward function was found to be most effective:

$$\mathbf{r_t} = \sum_{i \in C} \pi_{i,t-1} - 1000 \cdot \exp\left(-3 \cdot \epsilon^{\text{usr}}_{i,t}\right) + 1000 \cdot \exp\left(-3\right), \tag{3.25}$$

for any time step t, where the term $+1000 \cdot \exp(-3)$ counters the offset reward subtraction $-1000 \cdot \exp(-3 \cdot 1) = -49.8$ for a user satisfaction score ϵ^{usr} = 1. Figure 3.11 visualizes the proposed reward subtraction across the full range of user scores. Figure 3.11 shows that by corresponding to a flatter curve, the revised reward function indeed increases the penalty of moderately low user scores. Moreover, comparing Figure 3.10 with Figure 3.11, overall scores result in more severe penalties in the proposed user satisfaction term. For example, a score of $\epsilon^{\text{usr}} = 0.2$ results in a reward subtraction of approximately -500 under the proposed reward function, whereas the same score incurs a penalty of around -15 under the default formulation.

In scenarios with transformer overloading, a reward subtraction for transformer overloading also has to be considered. The proposed reward function for scenarios with transformer overloads is:

$$\mathbf{r_t} = -100 \cdot \epsilon^{\text{tr}}_{t-1} + \sum_{i \in C} \pi_{i,t-1} - 1000 \cdot \exp\left(-3 \cdot \epsilon^{\text{usr}}_{i,t}\right) + 1000 \cdot \exp\left(-3\right), \tag{3.26}$$

for any time step t, where $\epsilon^{\text{tr}}_{t-1}$ is the transformer overloading in kWh, π_{t-1} is the charging profit of connected EV i, and $\epsilon^{\text{usr}}_{i}$ is the user satisfaction score of EV i at departure.

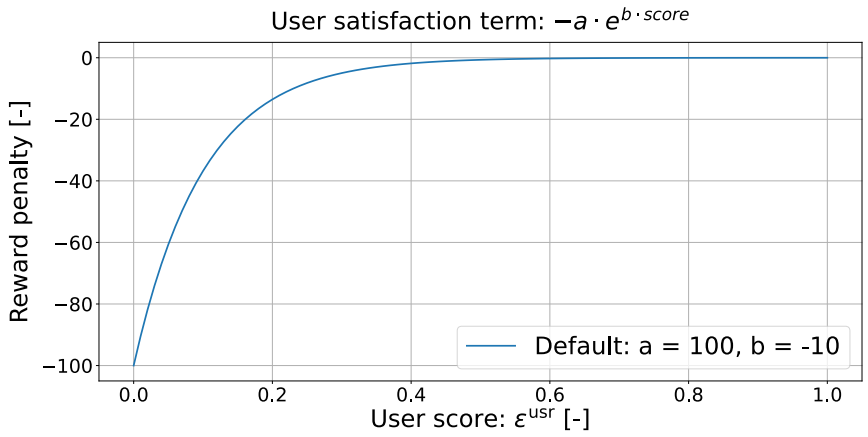


Figure 3.10: User satisfaction term in default reward function for all values of user satisfaction score

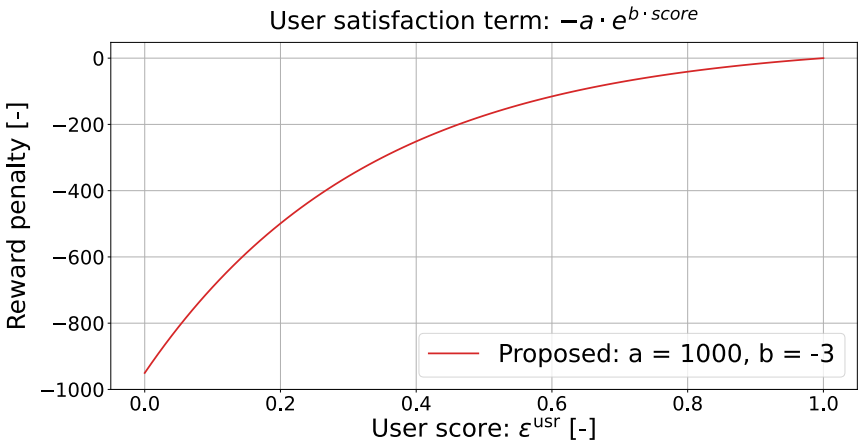


Figure 3.11: User satisfaction term in proposed reward function for all values of user satisfaction score

3.7. Baseline methods

SAC, TD3, and PPO were selected as Classic RL methods to solve the MDP. These algorithms are widely considered among the most stable and sample-efficient Deep RL algorithms. Furthermore, they can be applied in environments with continuous state and action spaces. In *EV2Gym*, agents can be trained with these algorithms in the Python script `train_stable_baselines.py`, which makes use of Deep RL implementations from the package *stable-baselines3* [55]. In experiments without transformer overloading, the state function defined in Equation 3.19 and reward function defined in Equation 3.25 are used for the Classic RL agents. In experiments with transformer overloading, agents are trained with the state function defined in Equation 3.19 and reward function defined in Equation 3.25.

In addition to the RL baselines, AFAP charging, and the offline Gurobi solver are also applied as benchmarks. AFAP represents the current conventional charging strategy and thus serves as a practical baseline. The Gurobi solver provides an offline solution under full knowledge of future events and is used as a reference for optimal performance. As discussed in the literature review in Chapter 2, MPC has demonstrated comparable performance to RL in real-time EV charging control. A comparison between MPC and RL would therefore be interesting. While *EV2Gym* includes several implementations of MPC, all these implementations consider the unrealistic static problem with knowledge of future events. Consequently, including these MPC results would not provide meaningful insights into the real-world performance of MPC.

3.8. Constrained Markov Decision Process

Most Safe RL algorithms incorporate a cost function related to constraint violation, enabling the agent to learn constraint satisfaction detached from reward maximization. In this framework, the agent is not only tasked with maximizing expected reward but must also ensure that the expected cost is below a predefined cost limit. The agent's new objective is:

$$\max_{\pi} J_r(\pi), \quad s.t. \quad J_c(\pi) \leq d, \tag{3.27}$$

where d is the cost limit, $J_r(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t\right]$ is the expected discounted reward over the simulation day, and $J_c(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t c_t\right]$ is the expected discounted cost over the simulation day [56]. In the rest of this thesis, the term 'cost' corresponds to constraint violation costs, not to be confused with the cost of charging, which is referred to as charging cost.

By formulating constraint violations into an actual inequality constraint through a cost function, rather than penalizing them in the reward function, the Safe RL approach allows more strict constraint adherence than Classic RL. To enable the application of Safe RL algorithms in *EV2Gym*, the MDP formulation must be extended to a CMDP, which requires the original reward function to be decoupled into separate reward and cost components.

In the problem formulation of this thesis, as defined in Section 3.2, the cost emerges from violations of the two soft constraints: the target SOC for departure and the transformer power limit. The cost function for this scenario is based on the reward subtraction terms of the MDP related to constraint violations, and updated in a trial-and-error process. The following function had the best performance:

$$\mathbf{c_t} = 5 \cdot \epsilon^{\mathsf{tr}}_{t-1} + \sum_{i \in C} 20 \cdot \exp\left(-3 \cdot \epsilon^{\mathsf{usr}}_{i,t}\right) - 20 \cdot \exp(-3), \tag{3.28}$$

for all transformer overloads $\epsilon^{\mathsf{tr}}_{t-1}$ at time step $t-1$ and user scores $\epsilon^{\mathsf{usr}}_{i}$ at time step t. The compensation term $-20 \cdot \exp(-3)$ is now negative because the cost function is positive. In a problem formulation where there is plenty of capacity available, transformer overloading cannot happen, thus making the target SOC for departure the only constraint that results in a cost when violated. For these scenarios the following cost function was found to be most effective:

$$\mathbf{c_t} = \sum_{i \in C} 20 \cdot \exp\left(-3 \cdot \epsilon^{\mathsf{usr}}_{i}\right) - 20 \cdot \exp(-3), \tag{3.29}$$

for all user scores $\epsilon^{\mathsf{usr}}_{i}$ at time step t. As the cost limit restricts the allowed cost during training, the transformer overloading and user satisfaction terms in the cost function can be defined at a smaller scale than in the reward function of the MDP. Decoupling these terms from the reward function removes the need to scale them artificially high to steer the agent toward constraint-satisfying behavior. The reward function in the CMDP is the total charging profit, effectively making profit maximization the only reward objective for Safe RL agents:

$$\mathbf{r_t} = \sum_{i \in C} \pi_{i,t-1} \tag{3.30}$$

According to Equation 3.27, the Safe RL algorithms aim to keep the expected, discounted sum of all costs in a simulation day below the cost limit. Figure 3.12 shows the CMDP cost function as defined in Equation 3.29. When the cost limit is equal to 1 and a user score of 0.7 is obtained, or the departure SOC of a single EV is 70%, the resulting cost exceeds the limit. Therefore, when the cost limit is equal to 1, the departure SOC of each EV in a simulation day is expected to rarely be below 70% for good Safe RL policies. To verify this, the number of occurrences of a minimum departure SOC below 70% is added as a metric in the final evaluation. Figure 3.12 shows that a cost limit of 2 is less strict, as lower user scores are possible within the limit. The expectation is that agents trained with higher cost limits will achieve more reward at the cost of worse user satisfaction.

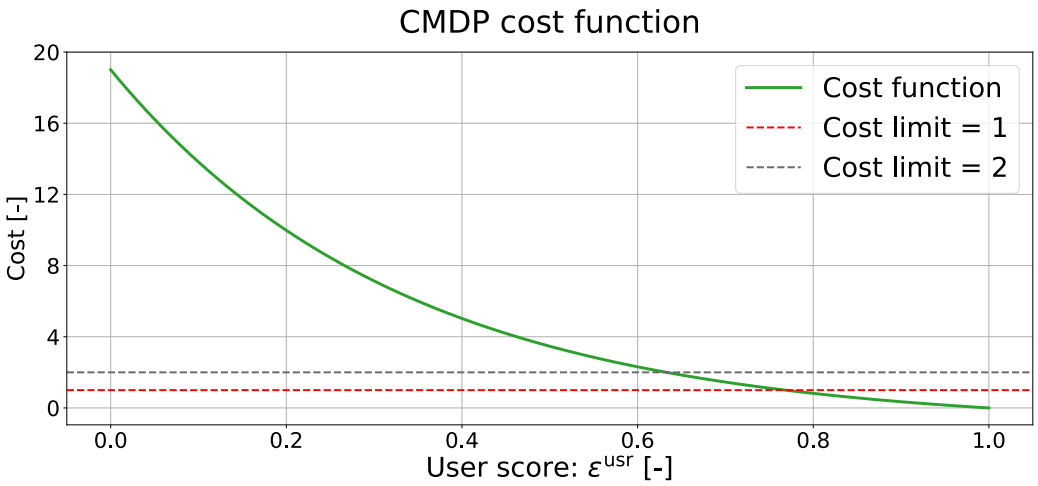


Figure 3.12: CMDP cost function without the transformer overloading term for all values of the user satisfaction score

3.9. Proposed method: CVPO

This thesis proposes the state-of-the-art Safe RL algorithm Constrained Variational Policy Optimization (CVPO) to solve the coordinated EV charging problem formulated as a CMDP. CVPO was selected because of its superior sample efficiency, training stability, and constraint satisfaction, as demonstrated in [56]. The algorithm learns new policies with a trust-region updating method similar to CPO. However, unlike CPO, CVPO formulates the objective as a probabilistic inference problem and improves policies in an off-policy fashion. CVPO splits the constrained problem into a convex optimization phase, the Expectation step (E-step), and a supervised learning phase, the Maximization step (M-step). The E-step aims to find the optimal variational distribution q that maximizes reward, is part of a feasible distribution family, and is within the trust region of the previous policy. The E-step is formulated as a regularized convex optimization problem and is solved analytically. After the distribution q is obtained in the E-step, a new policy is trained on this distribution in a supervised learning fashion in the M-step, effectively making the training process off-policy [56].

The Python script `train_safe_RL.py` was added to *EV2Gym* to facilitate the training of CVPO and other Safe RL algorithms. The implementation makes use of the Fast Safe Reinforcement Learning package, *fsrl*, developed by the authors of [56] and available on GitHub [57]. In addition to CVPO, *fsrl* offers implementations of alternative Safe RL algorithms, including CPO and custom Deep RL methods that incorporate a PID-based Lagrangian relaxation method as proposed in [58]. In this thesis, CPO, SAC with the Lagrangian function (SAC-L), and PPO with the Lagrangian function (PPO-L) are applied and compared to CVPO and the benchmarks. In all experiments, the reward function defined in Equation 3.30 is used for the Safe RL agents. In experiments without transformer overloading, the Safe RL agents use the state function defined in Equation 3.19 and cost function defined in Equation 3.29. In experiments with transformer overloading, they use the state function defined in Equation 3.20 and cost function defined in Equation 3.28.

In some training results of [56], the implementation of SAC-L converged to rewards exceeding those achieved by CVPO with only slightly higher costs, thus proving that SAC-L can be a viable alternative for CVPO. However, across all experiments of [56], CVPO converged to a lower cost. These observations lead to the hypothesis that SAC-L policies may favor reward more in the reward versus cost trade-off, whereas CVPO yields stricter constraint satisfaction. In Chapter 4, numerical simulations investigate how this hypothesis holds in the coordinated EV charging problem of this thesis.

Some parts of the CVPO and SAC-L algorithms are comparable. Both methods use critics that estimate future returns of cost and reward, and both explore new policies in an off-policy fashion. However, other elements are very different, particularly how each algorithm handles constraints. SAC-L employs the conventional primal-dual approach, approximating the constrained problem as unconstrained through

Lagrangian relaxation. The cost is added to the RL objective–maximizing expected reward–as a penalty term, scaled by a Lagrange multiplier: λ. This multiplier is adjusted dynamically: it increases when the observed cost exceeds the cost limit and decreases when the cost remains below the cost limit.

In contrast, CVPO embeds the cost constraint in a constrained optimization problem that is solved analytically, instead of approximating the constraint violation cost as a penalty term. The E-step ensures the policy is updated exclusively on an explored action distribution shown to respect the cost limit. As the constraint handling of CVPO is more strict than SAC-L, CVPO will generally find safer policies. However, the dynamic updating of the Lagrangian method of SAC-L may yield more rewarding performance than CVPO. This fundamental difference in constrained handling explains the observed difference between CVPO and SAC-L in the training results of [56].

Equation 3.28 combines the violation cost of two constraints into one cost function, as the algorithms of *fsrl* have one cost function implemented. Alternatively, it is possible to formulate a CMDP where each constraint is associated with a distinct cost function. For future work, it may be interesting to consider a cost function for each constraint. This thesis investigates whether the *fsrl* implementations with one cost function can learn behavior that finds good profit while respecting both the target SOC and transformer power limit constraints.

4

Results

This chapter introduces the setup of two experiments, discusses Safe RL training, and compares CVPO's performance in the experiments to several benchmarks.

4.1. Experimental setup

Experiment 1 represents a simplified scenario where transformer capacity is abundant. As transformer overloading cannot occur in this setting, inflexible loads and PV power generation are excluded. The objective is to maximize the EV aggregator's profit while ensuring the EV users are satisfied with their departure SOC. Experiment 2 is a more constrained scenario that includes PV, inflexible loads, and transformer overloading.

In each experiment, all algorithms were evaluated over the same set of 100 randomly sampled simulation days. The RL results were averaged over five agents trained with different random seeds. During training, the dynamic electricity prices were sampled from the 2023 prices. To make the evaluation more realistic, trained RL agents were tested on unseen electricity prices from 2024. In addition to the main performance comparisons, sensitivity analyses and ablation studies were conducted for the best-performing algorithms.

	Algorithms	Chargers	Charger Occupation	$\eta_{\text{charge}}, \eta_{\text{discharge}}$	Cost Limit
Exp. 1.1	TD3, SAC, PPO, CPO, PPO-L, SAC-L, CVPO	10	High	Variable	1, 10 (SAC-L)
Exp. 1.2	CPO, SAC-L, CVPO [1]	10	Low, very high	Variable	1, 10 (SAC-L)
Exp. 1.3	SAC-L, CVPO	30	High	Variable	2, 20 (SAC-L)
Exp. 1.4	SAC-L, CVPO	10	High	Variable	1, 2, 3 (CVPO) 10, 20, 30 (SAC-L)
Exp. 1.5	CVPO	10	High	Constant = 0.9	1

Table 4.1: Experiment 1 setup including sensitivity analyses and an ablation study.

Table 4.1 lists the five sub-experiments of Experiment 1. Experiment 1.1 was the main experiment, in which the performance of all algorithms was compared in a scenario with 10 chargers and an EV spawn multiplier of 5. As discussed in Section 3.3, this spawn multiplier resulted in a high charger occupation level. In reality, charger usage may fluctuate throughout the year. For example, during

[1] The trained agents from Exp. 1.1 were evaluated in the environment setting of Exp. 1.2.

vacation, parking lots will probably experience much lower charger occupation. To discover how the best-performing algorithms from Experiment 1.1 would perform in periods with deviating EV behavior, their agents were evaluated in Experiment 1.2 in situations with different levels of charger occupation. As discussed in Section 3.3, a low charger occupation was modeled by an EV spawn multiplier of 1, and a very high charger occupation by a spawn multiplier of 10. In Experiment 1.2, the aim was to examine how well the charging behavior of an agent trained on a certain charger occupation level generalized to days when the occupation is very different.

In Experiment 1.3, new CVPO and SAC-L agents were trained and evaluated in a scenario with 30 chargers to research the scalability of CVPO and SAC-L. The cost limit was increased in Experiment 1.3 to give the agents more flexibility. In experiment 1.4, new agents of CVPO and SAC-L were trained with the cost limit equal to 2 and 3, to study their sensitivity to the cost limit. In Experiment 1.5, the effect of the introduced variable charging and discharging efficiencies was investigated in an ablation study with constant η_{charge} = $\eta_{\text{discharge}}$ = 0.9. New CVPO agents were trained with $\eta = 0.9$ and evaluated in the setting of Experiment 1.1.

Table 4.2 summarizes the two sub-experiments of Experiment 2. In both experiments, the charging and discharging efficiencies were modeled as current-dependent. The cost limit was increased to 2 to give agents more flexibility in their learning process. Experiment 2.1 evaluated the performance of CVPO against SAC and SAC-L. SAC was included to have one Classic RL algorithm for comparison. SAC-L was selected because of its strong performance in Experiment 1.

	Algorithms	Chargers	Charger occupation	Tr limit	Load factor	PV factor	Cost limit
Exp. 2.1	SAC, SAC-L, CVPO	10	High	90 kW	0.5	0.1	2
Exp. 2.2	SAC, SAC-L, CVPO [2]	10	High	90 kW	0.6, 0.7	0.1	2

Table 4.2: Experiment 2 setup including a sensitivity analysis for the mean load capacity multiplier.

Experiment 2.1 was conducted in a setting with 10 chargers, an EV spawn multiplier of 5, a transformer power limit of 90 kW, a mean inflexible load capacity multiplier of 0.5 (standard deviation: 0.1), and a mean PV capacity multiplier of 0.1 (standard deviation: 0.05). This PV multiplier in theory corresponds to a small PV system, which can produce a maximum of approximately 9 kW–equivalent to about 20 PV panels, assuming 450Wp per PV panel. As discussed in Section 3.3, an EV spawn multiplier of 5 typically resulted in 6-8 EVs arriving at the parking lot. Given the maximum charging power of 11 kW per EV, the peak demand from 7 EVs could reach 77 kW. Combined with the peak inflexible load being 45 kW on average, transformer overloads were likely in this setting.

In Experiment 2.2, the agents from 2.1 were evaluated in scenarios with increased inflexible load levels. The goal was to assess how the obtained policies under a mean load capacity multiplier of 0.5 would generalize to a more constrained setting. The agents were tested in scenarios where the mean load multiplier was increased to 0.6 and 0.7.

4.2. Safe RL training

In *fsrl*, experiences can be collected from parallel train environments before updating the policy. The `episode_per_collect` parameter specifies the number of steps collected from the train environments. By adjusting `episode_per_collect` and the number of train environments, a balance between exploration and learning frequency can be tuned. Increasing these parameters will make the algorithm collect more episodes before updating the policy, thus increasing the exploration per gradient step and decreasing the learning frequency.

After every epoch, the policy is evaluated in test simulation days collected from parallel test environments to get insight into the policy's performance. The *EV2Gym* environment is very dynamic because every simulation day involves new samples of electricity prices, EV behavior, PV generation, and inflexible load consumption. Consequently, the environment changes considerably after every reset.

[2] The trained agents from Exp. 2.1 were evaluated in the environment setting of Exp. 2.2.

Therefore, to better represent the agents' performance, a large number of test environments is recommended, preferably at least 50. In the final training runs for the experiments of this thesis, the number of test environments was set at 100.

By default, *fsrl* saves new best policies in the following manner: if the mean test cost is below the cost limit and the mean test reward is better than that of the previously best-performing policy, the policy is saved as the new best policy. Until the algorithm learns policies that achieve costs below the cost limit, a new best policy is saved when the current reward is better than the previous best reward, regardless of the cost. Therefore, if algorithms throughout the training run never obtain costs below the limit, algorithms are assumed to be inadequate to solve the constrained optimization problem, since the best policy will be unsafe.

Most parameter values of the Safe RL algorithms were kept at the default configuration values of *fsrl*. Table 4.3 lists the adjusted parameters with the new values. The number of simulation steps per epoch was reduced from 10,000 to 3000 steps in Experiment 1 to get more frequent insight into the agents' learning. In Experiment 2, 9000 simulation steps per epoch were used because more steps were required to learn the more complicated problem. As one simulation day consists of 60 steps, one epoch contains $\frac{3000}{60}$ = 50 simulation days in Experiment 1. One epoch contains $\frac{9000}{60}$ = 150 simulation days in Experiment 2. The repeat per collect parameter of PPO-L was adjusted from 4 to 10 to increase learning speed. The buffer size was increased to 400,000 in Exp 1.3 because more epochs were required to learn adequate behavior. In Experiment 2, transformer overload costs occurred much less frequently than user satisfaction costs. To prevent the agent from forgetting the sparse transformer overload costs, the buffer size was increased to 5 million.

Discount factor γ	0.99
Steps per epoch	3000 (Exp 1) 9000 (Exp 2)
Repeat per collect (PPO-L only)	10
Random seed	1025, 1918, 1986, 3894, 6651
Buffer size	200,000 (Exp 1) 400,000 (Exp 1.3) 5,000,000 (Exp 2)
Train environments	10
Test environments	100

Table 4.3: Safe RL parameters with different values than the default configurations of *fsrl*.

Experiment 1.1, 1.2, 1.4, and 1.5 Safe RL training runs were executed on 2 CPU cores with 24GB of allocated memory on Delft University of Technology's supercomputer [59]. It took 3.5 hours to run 300 epochs, or 900,000 simulation steps, of Exp 1.1 training with CVPO and SAC-L. The Deep RL baselines were trained on 1 CPU core with 10GB of memory for 1 million simulation steps. TD3 and SAC training took 7 hours, PPO training 3.5 hours. Experiment 1.3 and Experiment 2.1-2.3 Safe RL training runs were executed on 4 CPU cores with 48GB of allocated memory. It took 12 hours to run 500 epochs, or 4,500,000 simulation steps, of Exp 2.1 training with CVPO and SAC-L. Experiment 2 SAC training runs were executed on 2 CPU cores with 24GB allocated memory for 3 million simulation steps and took around 17 hours. The seeds 1025, 1918, 1986, 3894, and 6651 were sampled randomly from the interval [1, 10,000]. These seeds were used to train five distinct agents for the final results.

4.2.1. Experiment 1.1: no transformer overloading

The number of train environments turned out to have a crucial effect on the learning performance of the safe RL agent. Through many simulations, it was found that in Experiment 1, CVPO usually converged to AFAP charging behavior, thus limiting the cost to zero and missing potential reward. When CVPO adopted AFAP behavior, it usually became trapped in this overly conservative policy and did not

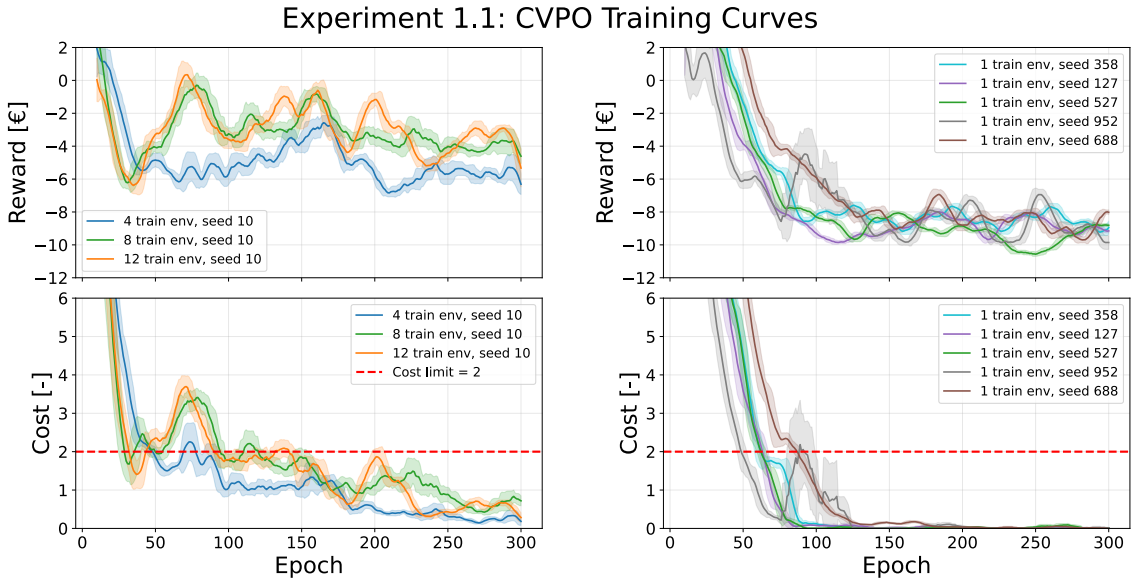


Figure 4.1: CVPO test cost and reward in Exp 1.1 training for different seeds and numbers of train environments. The data is averaged over a rolling window of 10.

discover behavior that may yield better profits at an acceptable higher cost. However, in some training runs, CVPO learned desired behavior in earlier stages before converging to AFAP. Desired behavior means having a cost just below the cost limit, and a reward substantially larger than AFAP charging.

Figure 4.1 visualizes the test cost and reward achieved by CVPO during Experiment 1.1 training for different seeds and train environments. Each graph represents a rolling average over the last 10 epochs of that run, with the center line representing the mean and the shaded area indicating one standard deviation above and below the mean. For all runs, 100 test environments were used, and the value of *episode_per_collect* was set equal to the number of train environments. Figure 4.1 shows training performance with 4, 8, and 12 train environments is superior to training performance with 1 train environment: the plots on the left stay close to the cost limit from the 50th to the 250th epoch while exploring valuable states that yield reward up to 0 €. All training runs with 1 train environment, the plots on the right, converge to AFAP before the 130th epoch for 5 different random seeds, while the best policies achieve no rewards higher than -4 €. Figure 4.1 shows that training performance with 8 and 12 train environments is slightly superior to the training performance with 4 train environments: the reward is higher, while the cost is still below the cost limit. As the difference between 8 and 12 train environments is minor, 10 train environments were used in all final training runs.

The reward potential of any simulation is dependent on the environment's initialization. Large fluctuations in electricity prices throughout the day offer promising conditions for a smart-charging algorithm, while flat price profiles offer limited reward potential. In contrast, by charging EVs to a higher SOC, the agent can always reduce cost regardless of the electricity prices. It is hypothesized that the reduced learning performance observed with a lower number of train environments is caused by the environment's dynamics. When the number of train environments is too low, the agent may see too few valuable state-action combinations to learn profitable behavior. Instead, policies would quickly converge to overly conservative policies by only focusing on cost reduction.

The scale of the cost function was also found to affect the training performance of Safe RL agents. Figure 4.2 shows the test cost and reward of CVPO's final Experiment 1.1 training runs for different scales of the cost function. The parameter values as defined in Table 4.3 were used for these runs. Each curve shows the mean of the five seeds, with the shaded area indicating one standard deviation above and below the mean. In the plots on the left, a cost limit of 2 was used and the factor of the user cost was 20, making the user satisfaction term of the cost function:

$$-20 \cdot \exp(-3 \cdot \epsilon^{\text{usr}}) \tag{4.1}$$

In the plots on the right, the cost limit was 20 and the factor of the user cost was 200, making the user satisfaction term of the cost function:

$$-200 \cdot \exp(-3 \cdot \epsilon^{\text{usr}}) \tag{4.2}$$

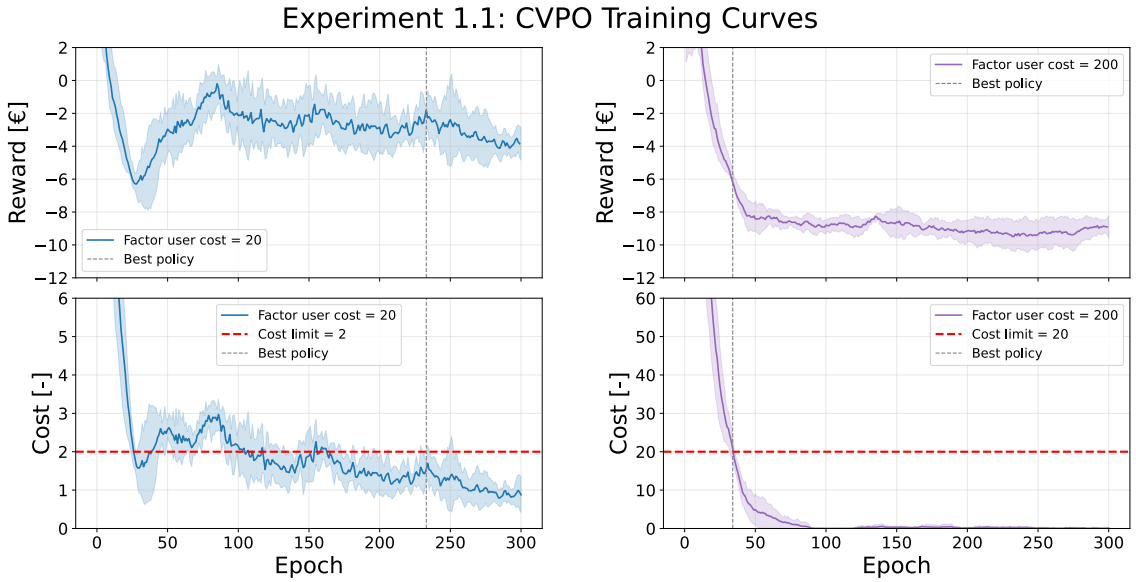


Figure 4.2: CVPO test cost and reward in Exp 1.1 training for different scales of user cost. The data is averaged over five random seeds.

Apart from scaling, these configurations are theoretically identical. However, as shown in Figure 4.2, in practice CVPO achieved better policies with a smaller scale. The grey dashed line approximates the mean best policy found with both configurations. In the plots on the left for the smaller scale of the cost function, the mean test reward was -2 € and the mean cost was 1.7 for the best policy. In the plots on the right, they were -6 € and 20 for the best policy. Again, with the larger scale of the cost function, CVPO quickly converged to AFAP and missed the opportunity to learn profitable charging behavior.

As discussed in Section 3.9, CVPO handles constraints very strictly. The absolute cost values become a magnitude larger when the factor of the user term is set to 200 (Equation 4.2), while the reward scale remains unchanged. This may cause CVPO to heavily penalize actions with any cost, thus resulting in overly conservative behavior. These observations support the suggestion that, at least for CVPO in this problem setting, it is beneficial to have the cost function on a similar scale as the reward function.

However, this suggestion did not apply to all Safe RL agents. Figure 4.3 shows the training curves obtained by SAC-L in Experiment 1.1 with parameters as defined in Table 4.3. Except for some outliers in early epochs, SAC-L with the user term factor equal to 20 failed to produce policies that satisfied the cost limit. The algorithm converged to positive rewards with the cost above the limit of 2, thus indicating increased profit by not charging EVs sufficiently. In contrast, the right-hand plots in Figure 4.3–corresponding to a user term factor of 200 and cost limit of 20–show that SAC-L found policies yielding a cost lower than the limit with an adequate reward in many of the 300 epochs, thus proving in these training runs SAC-L performed better with the user satisfaction term as defined in Equation 4.2 and cost limit equal to 20.

A possible explanation is that SAC-L, under the default configuration of *fsrl*, creates less aggressive penalization for cost limit violations than CVPO. Consequently, SAC-L may benefit from the larger cost scale, which better amplifies cost penalties relative to the reward. PPO-L and CPO showed better learning performance with a user term factor of 20 and cost limit equal to 2. Table 4.4 lists the cost function and cost limit used for each Safe RL algorithm in Experiment 1.

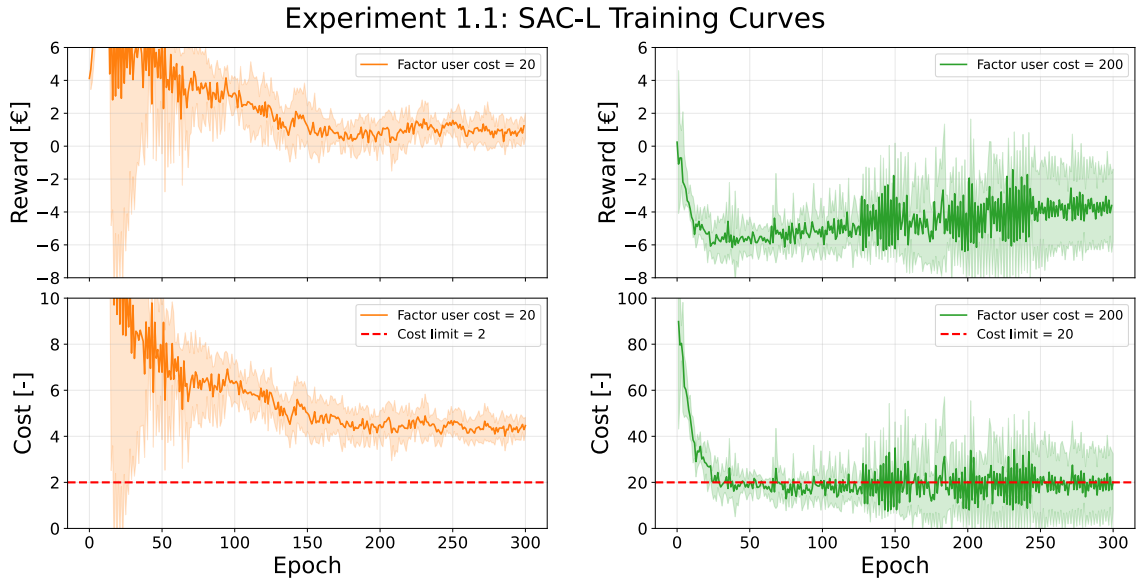


Figure 4.3: SAC-L test cost and reward during Exp 1.1 training for different scales of user cost. The data is averaged over five random seeds.

Algorithm	Cost function	Cost limit
SAC-L	$\mathbf{c_t} = \sum_{i \in C} 200 \cdot \exp\left(-3 \cdot \epsilon_i^{\text{usr}}\right) - 200 \cdot \exp(-3)$	10
PPO-L	$\mathbf{c_t} = \sum_{i \in C} 20 \cdot \exp\left(-3 \cdot \epsilon_i^{\text{usr}}\right) - 20 \cdot \exp(-3)$	1
CPO	$\mathbf{c_t} = \sum_{i \in C} 20 \cdot \exp\left(-3 \cdot \epsilon_i^{\text{usr}}\right) - 20 \cdot \exp(-3)$	1
CVPO	$\mathbf{c_t} = \sum_{i \in C} 20 \cdot \exp\left(-3 \cdot \epsilon_i^{\text{usr}}\right) - 20 \cdot \exp(-3)$	1

Table 4.4: Cost function and cost limit of the Safe RL algorithms in Experiment 1.

4.2.2. Experiment 1.3: scalability

Figure 4.4 shows the training runs of CVPO and SAC-L for Experiment 1.3, which investigates scalability in a setting with 30 chargers. The left-hand plots show the average of five CVPO runs trained under the random seeds defined in Table 4.3. CVPO demonstrated stable performance across the five random seeds. While for all seeds, CVPO converged towards AFAP charging in the end, the algorithm seemed to explore desired behavior with costs near the cost limit between epochs 500 and 600.

On the right-hand side of Figure 4.4, the SAC-L runs are plotted separately because of the considerable deviation between different runs. Although SAC-L managed to get the cost below the cost limit faster than CVPO for seeds 6651, 3894, and 1025, it did not achieve costs below the cost limit for seeds 1918 and 1986. Instead, SAC-L seemed to value reward improvement over cost reduction for these seeds, as it reached rewards up to 40 €. However, these rewards came at costs exceeding 500, over 25 times the cost limit. Undoubtedly, the SAC-L agents achieved very low customer satisfaction at these costs. These unstable training results indicate that SAC-L with the cost function and limit defined in Table 4.4, is not scalable to a scenario with 30 chargers.

4.2.3. Experiment 2: with PV, inflexible loads, and transformer overloading

Figure 4.5 presents the training curves obtained by CVPO and SAC-L in Experiment 2. A comparison between the CVPO and SAC-L curves with the load multiplier equal to 50% of the transformer power limit reveals that SAC-L reached costs below the limit earlier than the CVPO. However, CVPO demonstrated more stability, with lower variance across training runs and more consistent convergence to policies that respected the cost limit. Furthermore, from epoch 300 to 500, CVPO runs rarely experienced a cost value greater than 4. For all 500 epochs, the SAC-L runs maintained a constant frequency of costs above 40. These results suggest that CVPO may be more suitable for the constrained setting

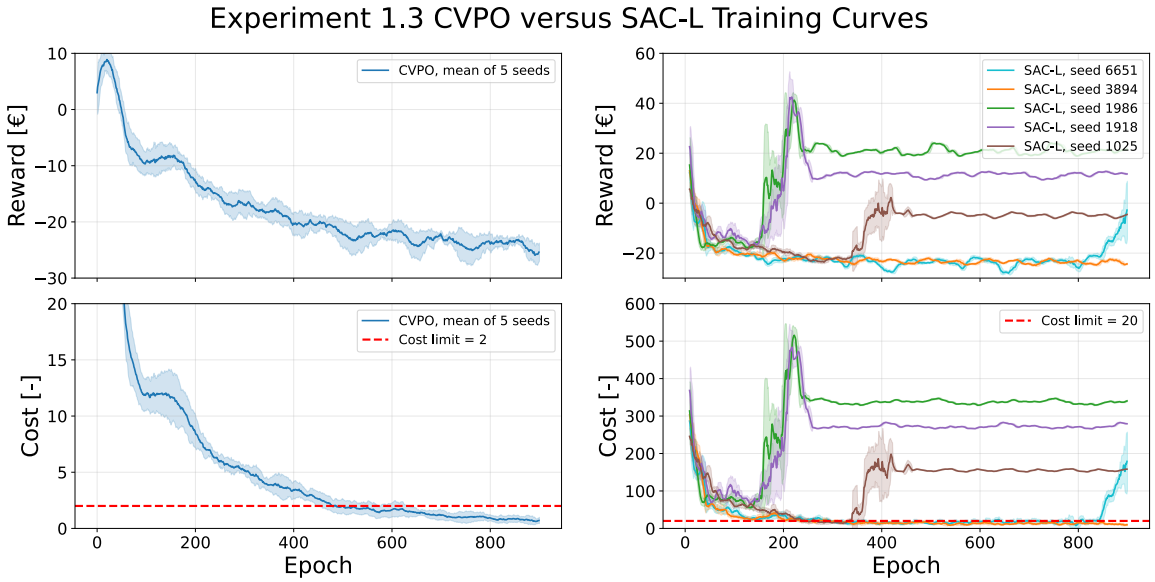


Figure 4.4: CVPO versus SAC-L test cost and reward during Exp 1.3 training. SAC-L results are averaged over a rolling window of 10.

of Experiment 2.1, as it offers a more stable learning process.

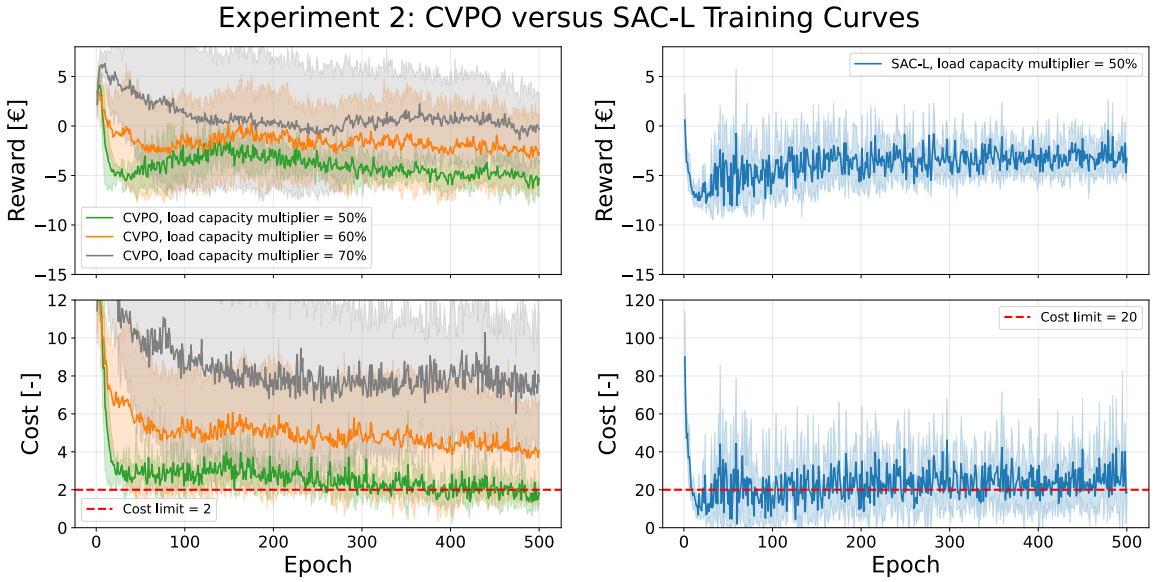


Figure 4.5: CVPO versus SAC-L test cost and reward during Exp 2 training. The data is averaged over five random seeds.

In addition to the final training runs of Experiment 2.1 with a mean load capacity multiplier of 50%, Figure 4.5 also shows CVPO's training performance in more constrained scenarios where the mean load capacity multiplier is 60% and 70% of the transformer power limit. The observation that CVPO with a load capacity multiplier of 50% only began achieving costs below the cost limit after approximately 400 epochs, proves that CVPO struggled to find good behavior in the setting of Experiment 2. Two hypotheses were formulated to explain this increased complexity compared to Experiment 1. The first hypothesis argued that, given the sparse nature of the transformer overload costs, the agent may have experienced too few transformer overloads to learn how to manage charging effectively around peaks in the inflexible load. The second hypothesis speculated that the optimization problem became overly complex due to the transformer overloads, and CVPO could not learn rewarding behavior.

If the first hypothesis were true, increasing the mean load capacity multiplier would improve training performance because the agent would experience transformer overloads more frequently. Figure 4.5 shows that training performance deteriorated substantially with higher load multipliers. For a multiplier of 60%, CVPO was barely able to achieve costs below the limit. When the load capacity multiplier was increased to 70%, none of the CVPO training runs learned policies yielding costs below the cost limit. These training results thus support the second hypothesis, and a load capacity multiplier of 60% was identified as an approximate upper limit for CVPO. If the load capacity multiplier becomes 60% or higher, the optimization problem under the parameters of Experiment 2 becomes too complicated for CVPO to learn constraint-satisfying behavior in the problem formulation of this thesis.

4.3. Evaluation

Table 4.5 presents the evaluation results of Experiment 1.1. The reported metrics are profit, the mean user score across all EV users, the mean minimal user score (i.e., the lowest individual departure SOC), the percentage of simulation days where the minimal score was below 0.7, the total energy charged, the total energy discharged, and the execution time per simulation day. Except for the metric $\epsilon^{\text{usr}}_{\text{min}} < 0.7$, the displayed values are the mean and standard deviation across the 100 simulation days. For RL, the results are first averaged over the five seeds before taking the mean and standard deviation.

Algorithm	Profit (€)	$\epsilon^{\text{usr}}_{\text{avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min,avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min}} < 0.7$ (%)	Energy Ch. (kWh)	Energy Dis. (kWh)	Execution Time (s)
AFAP	–6.4 ±4.7	100 ±0	100 ±0	0.0	100 ±28	0.0 ±0	0.02 ±0.01
TD3	5.6 ±7.3	74 ±10	51 ±8	97.6	44 ±25	118 ±67	0.23 ±0.11
PPO	3.1 ±6.4	75 ±10	51 ±6	97.4	40 ±24	73 ±54	0.21 ±0.12
SAC	2.1 ±4.6	78 ±9	51 ±7	98.0	47 ±23	72 ±44	0.20 ±0.08
PPO-L	3.4 ±12.2	81 ±19	69 ±21	58.0	85 ±60	145 ±103	0.06 ±0.01
CPO	–5.1 ±4.5	97 ±4	86 ±18	24.0	110 ±33	31 ±33	0.08 ±0.03
SAC-L	–5.4 ±3.9	98 ±3	91 ±14	10.8	136 ±41	59 ±45	0.10 ±0.03
CVPO	–5.0 ±4.0	99 ±3	92 ±14	11.6	125 ±40	42 ±40	0.05 ±0.01
Optimal	–4.9 ±4.7	99 ±0	98 ±1	0.0	116 ±33	22 ±11	1.10 ±0.65

Table 4.5: Experiment 1.1 evaluation results from 100 simulation days, RL results averaged over five random seeds.

The execution time was measured to assess the feasibility of deploying the agents in real-time applications. Since all agents completed the 60 steps of a simulation day well under one minute, these results indicate that the RL agents determine actions fast enough for real-time EV charging scenarios. As the optimal offline Gurobi solver, denoted as 'Optimal' in Table 4.5, cannot be applied in real-time, its execution time has limited meaning. Nonetheless, it was added to the table to facilitate comparison with the RL agents. Table 4.5 shows that the optimal solver did not achieve perfect user satisfaction, i.e. it did not reach 100% SOC for all EVs at departure. This is due to a modeling simplification: the current-dependent charging and discharging efficiencies are not modeled in the Gurobi solver to reduce the computational burden. Instead, fixed efficiencies $\eta_{\text{charge}} = \eta_{\text{discharge}} = 0.9$ are used.

Table 4.5 shows the Classic RL algorithms–TD3, PPO, and SAC–achieved the highest and only positive mean profits. However, these profits came at a low mean user satisfaction. Furthermore, for all Classic RL agents, almost every day at least one EV user had an unacceptable low departure SOC of 50%. While an average departure SOC of 75% might appear acceptable at first, these outcomes must be interpreted in the context of the problem formulation described in Section 3.2. As the formulation allows profit to increase unrealistically when EVs are insufficiently charged, only agents that achieve an average user satisfaction close to 100% can be meaningfully compared to the AFAP and optimal benchmarks. These results indicate that even in the simplified setting of Experiment 1, the Classic RL algorithms fail to effectively balance the trade-off between profit and user satisfaction, despite the

strong reward subtractions for low user scores in the reward function. This is further demonstrated by the charged and discharged energy values of Table 4.5: for TD3, PPO, and SAC, total discharged energy exceeds charged energy.

Table 4.5 shows that only CPO, SAC-L, and CVPO, on average, charged more than they discharged, had a user satisfaction close to the desired 100%, and charged all EVs to a departure SOC above 70% in most of the simulation days. These findings indicate that CPO, SAC-L, and CVPO are the only algorithms capable of learning constraint-satisfying charging behavior in the problem of Experiment 1.1. Among them, CVPO performed best overall, achieving the highest mean profit and mean user satisfaction. Compared to AFAP, CVPO had a mean profit improvement of 22% while the mean user score was only reduced by 1%. Although SAC-L had on average 0.3 € less profit than CPO, it showed better compliance with the user satisfaction constraint by yielding a minimum departure SOC below 70% in only 10.8% of the simulation days versus 24.0% for CPO. Thus, SAC-L in Experiment 1.1 is concluded to be superior to CPO.

However, CVPO did not guarantee a full EV battery for every user. In 11.6% of simulation days, one EV user had a departure SOC below 70%. This shortcoming is attributed to the approximation of RL, which remains in Safe RL algorithms. As a result, it is challenging to enforce strict constraint satisfaction in RL. Nevertheless, the results of Table 4.5 indicate that Safe RL's addition of a cost function and cost limit significantly improves constraint satisfaction compared to Classic RL algorithms. In real applications, an EV aggregator could force EVs to be at least 70% with a fallback mechanism, also referred to as a shield, that charges an EV at maximum power to 70% whenever the departure SOC risks falling below 70%. However, new evaluations would have to be done in this setting, as these interventions would likely reduce profit. Moreover, the reduced flexibility due to the shield might affect overall learning performance.

Algorithm	Δ Profit (€)	$\Delta\epsilon^{\text{usr}}_{\text{avg}}$ (%)	$\Delta\epsilon^{\text{usr}}_{\text{min,avg}}$ (%)	Δ Energy Ch. (kWh)	Δ Energy Dis. (kWh)
AFAP	–1.50 ±1.11	+0.01 ±0.00	+0.02 ±0.01	–16 ±10	–22 ±11
TD3	+10.56 ±8.04	–0.24 ±0.04	–0.47 ±0.04	–72 ±22	+95 ±35
PPO	+8.07 ±8.00	–0.23 ±0.05	–0.46 ±0.04	–76 ±23	+51 ±32
SAC	+7.01 ±6.01	–0.21 ±0.03	–0.47 ±0.03	–69 ±22	+50 ±24
PPO-L	+8.33 ±7.56	–0.17 ±0.02	–0.29 ±0.07	–31 ±18	+123 ±47
CPO	–0.17 ±1.52	–0.01 ±0.02	–0.11 ±0.07	–6 ±11	+9 ±15
SAC-L	–0.43 ±5.84	–0.00 ±0.01	–0.07 ±0.07	+20 ±49	+37 ±34
CVPO	–0.05 ±2.35	+0.00 ±0.02	-0.05 ±0.08	+9 ±21	+20 ±35
Optimal	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0

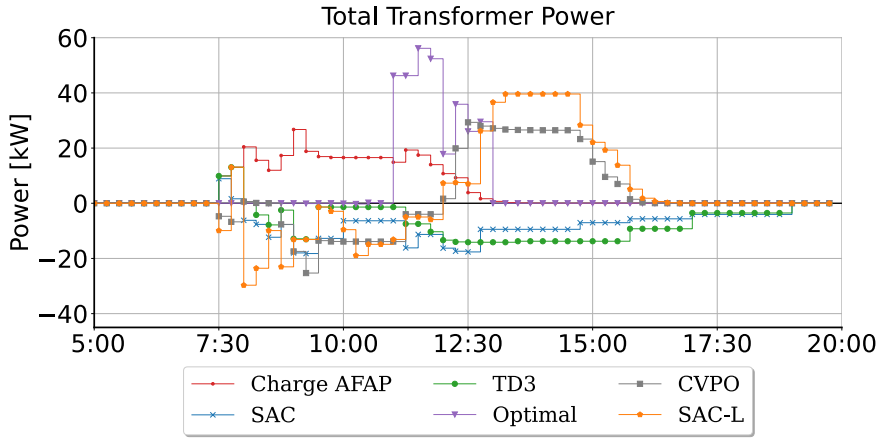
Table 4.6: Experiment 1.1: Mean difference per simulation day compared to optimal case.

The standard deviation of most results in Table 4.5 is large. Take, for example, the profit of CVPO, which is on average –5.0 €. Still, because of the standard deviation of 4.0 €, it may just as well be 0 € one day and –10 € another. In static environments, the standard deviation should be as small as possible to ensure that algorithms perform consistently. However, in the dynamic environment of this thesis, this is not the case. Each simulation day requires a different charging strategy and has a substantially different reward potential. The standard deviation is still reported because it provides some information about the distributions. For instance, the profit distribution of CVPO is more similar to the optimal distribution than the PPO-L profit distribution, as its standard deviation of 12.2 € is much more than the standard deviation of 4.7 € of the optimal profits.

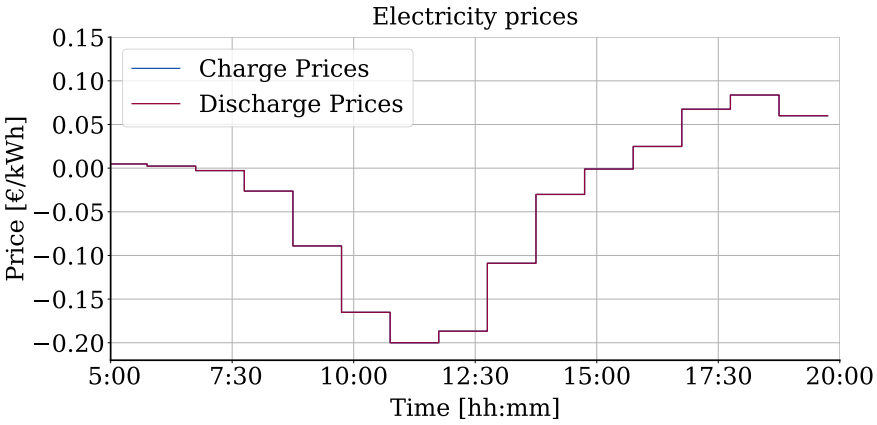
Table 4.6 presents the results of Experiment 1.1 in a different format. First, for each metric, the algorithms' difference per simulation day from the optimal case is calculated. Then, the differences are averaged across the five seeds and 100 simulation days. While the standard deviation is still very

large, CVPO's mean daily profit difference of 0.05 € suggests that CVPO's charging schemes are most similar to the optimal offline solver.

Figure 4.6 presents example charging schemes for a random simulation day generated by SAC, TD3, CVPO, SAC-L, AFAP charging, and the optimal offline solver. The simulation day is sampled from the set of simulation days used for Experiment 1.1. Therefore, as there are no PV power generation or inflexible loads, total transformer power equals the parking lot's total EV charging power. The electricity price profile, shown in Figure 4.6b, displays a significant price drop early in the day. Prices are mostly negative, meaning that discharging during these periods results in a net cost, and agents should favor charging actions to make a profit.



(a) Charging schemes of SAC, TD3, CVPO, SAC-L, AFAP charging, and the optimal offline solver.



(b) Electricity prices of the simulation day

Figure 4.6: Different algorithms' charging schemes for an exemplary simulation day of Experiment 1.1.

Figure 4.6a shows that the Classic RL algorithms fail to find effective charging schemes. Both SAC and TD3 constantly choose discharging actions, leading to low user satisfaction and a net cost. The agents do not incorporate prices into their policies and fail to value user satisfaction, despite the heavy reward subtraction imposed in the reward function. These results further demonstrate the unsuitability of Classic RL algorithms for the EV charging task addressed in this thesis. Figure 4.6a demonstrates that the CVPO and SAC-L agents achieve more promising charging behavior by choosing primarily V2G actions in the morning and charging actions in the afternoon. This reflects a better balance between user satisfaction and profitability. However, Figure 4.6b shows that the electricity price was negative during all their discharging actions. Therefore, Figure 4.6 indicates that Safe RL agents may also fail to process price information in their decision-making and merely learn one charging strategy that, on

average, leads to good results.

Charger Occupation	Algorithm	Profit (€)	ϵ^{usr}_{avg} (%)	$\epsilon^{usr}_{min,avg}$ (%)	$\epsilon^{usr}_{min} < 0.7$ (%)	Energy Ch. (kWh)	Energy Dis. (kWh)
Very low (Spawn = 1)	AFAP	–2.1 ±2.5	100 ±0	100 ±0	0.0	31 ±18	0.0 ±0
	CPO	–1.5 ±2.5	97 ±9	93 ±15	11.8	32 ±20	10 ±18
	SAC-L	–1.6 ±2.4	98 ±6	95 ±11	6.2	41 ±25	19 ±22
	CVPO	–1.5 ±2.2	97 ±7	94 ±13	8.8	40 ±24	19 ±20
	Optimal	–1.5 ±2.5	99 ±0	98 ±1	0	33 ±21	4 ±6
Very high (Spawn = 10)	AFAP	–9.1 ±5.6	100 ±0	100 ±1	0.0	135 ±28	0.0 ±0
	CPO	–7.4 ±5.2	98 ±3	88 ±16	17.8	154 ±34	40 ±33
	SAC-L	–6.9 ±4.6	99 ±2	91 ±13	10.4	190 ±42	82 ±50
	CVPO	–6.5 ±4.4	99 ±2	93 ±14	9.8	173 ±41	59 ±52
	Optimal	–6.5 ±5.7	99 ±0	98 ±1	0	158 ±31	31 ±10

Table 4.7: Experiment 1.2 evaluation results from 100 random simulation days, RL results averaged over five random seeds.

Table 4.7 presents the results of Experiment 1.2, where the best-performing RL agents from Experiment 1.1 were evaluated on unseen EV arrival distributions. Regardless of the charger occupation level, all Safe RL algorithms have mean profit and user satisfaction close to the optimal solver. These results indicate that the Safe RL algorithms, at least for smaller parking lots up to 10 chargers, learn behavior that generalizes well to unseen levels of charger occupation. The mean profit improvement of CVPO compared to AFAP is 29% for both cases of charger occupation. For all Safe RL agents, the average user satisfaction in the case of the very high charger occupation is larger than in the case of the very low charger occupation. Furthermore, the standard deviation of the mean user satisfaction is smaller. This further demonstrates that while the agents on average deliver full EVs at the end of they, they fail to do so for every EV user.

Chargers	Algorithm	Profit (€)	ϵ^{usr}_{avg} (%)	$\epsilon^{usr}_{min,avg}$ (%)	$\epsilon^{usr}_{min} < 0.7$ (%)	Energy Ch. (kWh)	Energy Dis. (kWh)
30 (Cost limit = 2)	AFAP	–18.8 ±12.2	100 ±0	100 ±0	0.0	297 ±49	0.0 ±0
	SAC-L	1.5 ±20.9	88 ±10	61 ±20	75.6	243±93	245 ±172
	CVPO	–17.0 ±10.6	98 ±2	77 ±19	39.6	351 ±65	98 ±58
	Optimal	–13.9 ±12.6	99 ±0	97 ±2	0.0	338 ±54	59 ±21

Table 4.8: Experiment 1.3 evaluation results from 100 random simulation days, RL results averaged over five random seeds.

Table 4.8 shows the results from Experiment 1.3, the scalability study with 30 chargers. The cost limit was increased to 2 in this experiment to provide the agents more flexibility in the learning process. Although the CVPO agents improved the mean profit compared to AFAP by 10%, in 39.6% of the simulation days, CVPO's charging behavior resulted in at least one EV having a departure SOC below 70%. This can be explained by Figure 3.12, which shows that user scores below 0.7 are allowed under a cost limit of 2. Still, as the profit improvement compared to AFAP is lower than CVPO's improvement of 22% in Experiment 1.1, and user satisfaction was guaranteed less, the scalability of CVPO to much larger parking lots seems to be limited. However, with a mean user satisfaction 10% lower than CVPO, SAC-L performed significantly worse in the scalability experiment. As Figure 4.4 showed, SAC-L failed to find constraint-satisfying behavior for every random seed in Experiment 1.3. These results indicate that, in the context of this thesis, CVPO has limited scalability, but is more scalable than SAC-L.

Cost Limit	Algorithm	Profit (€)	$\epsilon^{\text{usr}}_{\text{avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min,avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min}} < 0.7$ (%)	Energy Ch. (kWh)	Energy Dis. (kWh)
	AFAP	–6.4 ±4.7	100 ±0	100 ±0	0.0	100 ±28	0 ±0
	Optimal	–4.9 ±4.7	99 ±0	98 ±1	0.0	116 ±33	22 ±11
10	SAC-L	–5.4 ±3.9	98 ±3	91 ±14	10.8	136 ±41	59 ±45
1	CVPO	–5.0 ±4	99 ±3	92 ±14	11.6	125 ±40	42 ±40
20	SAC-L	–2.8 ±3.3	97 ±0	81 ±17	27.4	151 ±45	109 ±63
2	CVPO	–3.4 ±4.2	96 ±1	86 ±18	23.8	129 ±41	67 ±61
30	SAC-L	–1.7 ±3.9	95 ±5	82 ±17	24.6	157 ±44	130 ±70
3	CVPO	–2 ±5.4	95 ±7	83 ±20	30.2	131 ±43	83 ±76

Table 4.9: Experiment 1.4 evaluation results from 100 random simulation days, RL results averaged over five random seeds.

Table 4.9 lists the results of Experiment 1.4, where the sensitivity of SAC-L and CVPO to the cost limit was researched. Table 4.9 shows that for both algorithms, the mean profit was improved for higher cost limits, but user satisfaction was reduced. After proving that the cost function and limit of Safe RL compared to Classic RL algorithms greatly improved constraint satisfaction in Experiment 1.1, Experiment 1.4 demonstrates that the cost limit is an effective tool for adjusting the trade-off between reward and constraint satisfaction.

Table 4.9 also shows that CVPO only outperforms SAC-L in the most constrained scenario with a cost limit of 1 (10 for SAC-L): SAC-L has a higher mean profit and similar user satisfaction in the other scenarios. This indicates that SAC-L may have better learning performance in less constrained problem settings, while CVPO is more suitable for very constrained scenarios.

Algorithm	Profit (€)	$\epsilon^{\text{usr}}_{\text{avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min,avg}}$ (%)	$\epsilon^{\text{usr}}_{\text{min}} < 0.7$ (%)	Energy Ch. (kWh)	Energy Dis. (kWh)
AFAP	–6.4 ±4.7	100 ±0	100 ±0	0.0	100 ±28	0.0 ±0
CVPO (variable η)	–5.0 ±4.0	99 ±3	92 ±14	11.6	125 ±40	42 ±40
CVPO ($\eta = 0.9$)	–5.0 ±4.0	99 ±3	93 ±15	11.4	122 ±38	39 ±39
Optimal	–4.9 ±4.7	99 ±0	98 ±1	0.0	116 ±33	22 ±11

Table 4.10: Experiment 1.5 evaluation results, the simulation days from Experiment 1.1 are used. RL results are averaged over five random seeds.

Table 4.10 shows the results from Experiment 1.5, where CVPO agents were trained in the same setting as Experiment 1.1 but now with a constant charging and discharging efficiency of $\eta = 0.9$. Afterward, the agents were evaluated on the set of simulation days used for Experiment 1.1, where the efficiencies were current-dependent. Table 4.10 shows that the agents trained with constant efficiency had the exact same average user satisfaction and profit, improved the average minimum user score by 1%, and had a minimum user score below 70% in 0.2% less of the simulation days. These results indicate that modeling current-dependent charging efficiencies for a parking lot with three-phase chargers does not result in significant performance changes. In fact, RL algorithms may even find better behavior when a constant efficiency is assumed.

Table 4.11 presents the results of Experiment 2.1. With a mean user satisfaction of 83%, SAC still did not value user satisfaction sufficiently. SAC did not experience any transformer overloads, but this can be explained by the charging behavior of SAC shown in Figure 4.6. As SAC primarily chooses discharging actions, transformer overloads will rarely happen.

Algorithm	Profit (€)	ϵ^{usr}_{avg} (%)	$\epsilon^{usr}_{min,avg}$ (%)	$\epsilon^{usr}_{min} < 0.7$ (%)	Tr Ov. (kWh)	Tr Ov. (%)
AFAP	–7.3 ±5.6	100 ±0	100 ±1	0.0	0.18 ±1.4	3.0
SAC	–1.0 ±4.4	83 ±9	55 ±12	87.4	0.00 ±0.0	0.0
SAC-L	–1.8 ±5.4	95 ±5	82 ±14	21.2	1.64 ±6.8	13.2
CVPO	–3.9 ±5.5	95 ±5	78 ±18	35.4	0.08 ±1.0	1.0
Optimal	-5.2 ±6.5	99 ±0	97 ±2	0.0	0.00 ±0.0	0.0

Table 4.11: Experiment 2.1 evaluation results from 100 random simulation days, RL results averaged over five random seeds.

Table 4.11 shows that by having a minimal departure SOC below 70% in 21.2% of the simulation days, compared to 35.4% of CVPO, SAC-L guaranteed better user satisfaction in Experiment 2.1. On the other hand, by having transformer overloads in 1.0% of the simulation days versus 13.2% for SAC-L, CVPO guaranteed better transformer power limit compliance. As the mean user satisfaction for CVPO and SAC-L is equal, and preventing transformer damage is assumed to be more important than guaranteeing a high departure SOC for a single EV, the performance of CVPO in Experiment 2.1 is concluded to be superior to SAC-L. This confirms the earlier notion based on Figure 4.5 that CVPO found better policies in the constrained problem setting of Experiment 2. CVPO seems especially more capable of learning to respect the transformer limit constraint.

Although Table 4.11 shows that CVPO compared to AFAP improved the mean profit by 47% and reduced the number of transformer overloads by 2%, the mean user satisfaction was reduced by 5% and in 35.4% of the simulation days at least one EV had a departure SOC below 70%. The training results shown in Figure 4.5 suggested that CVPO for a mean load multiplier of 50% of the transformer limit was capable of finding rewarding and constraint-satisfying behavior and a multiplier of 60% could be an upper limit. However, based on the low user satisfaction in Table 4.11, it was concluded that the load multiplier of 50% may already be an upper limit for CVPO.

Load Mult. (%)	Algorithm	Profit (€)	ϵ^{usr}_{avg} (%)	$\epsilon^{usr}_{min,avg}$ (%)	$\epsilon^{usr}_{min} < 0.7$ (%)	Tr Ov. (kWh)	Tr Ov. (%)
60%	AFAP	–6.2 ±4.2	100 ±0	100 ±0	0.0	0.28 ±1.5	6.0
	SAC	–0.4 ±3.8	82 ±9	54 ±10	92.0	0.00 ±0.0	0.0
	SAC-L	–2.1 ±3.5	97 ±4	85 ±14	16.2	2.46 ±8.1	17.0
	CVPO	–4.1 ±3.5	96 ±4	81 ±18	29.4	0.19 ±1.4	3.2
	Optimal	–4.5 ±4.1	98 ±0	97 ±1	0.0	0.0 ±0	0.0
70%	AFAP	–7.4 ±6.4	100 ±0	100 ±0	0.0	2.46 ±10.5	10.0
	SAC	0.0 ±6.0	81 ±9	55 ±11	89.6	0.06 ±0.6	1.2
	SAC-L	–0.5 ±17.2	95 ±6	83 ±14	17.4	8.59 ±22.5	33.4
	CVPO	–2.9 ±12.8	95 ±6	78 ±18	36.4	0.68 ±4.1	6.8
	Optimal	–5.7 ±5.9	99 ±0	98 ±1	0.0	0.00 ±0.0	0.0

Table 4.12: Experiment 2.2 evaluation results from 100 random simulation days, RL results averaged over five random seeds.

Table 4.12 presents the results of the final experiment, Experiment 2.2. By having substantially fewer transformer overloading than SAC-L and higher customer satisfaction than SAC, CVPO also had the best performance in Experiment 2.2. CVPO compared to AFAP had 47% and 32% less transformer overloads for the situation with a mean load multiplier of 60% and 70%, respectively. As the learned

behavior with a mean load multiplier of 50% seems to scale well to situations with higher inflexible load, EV aggregators could perhaps use CVPO policies trained with a lower level of inflexible loads than they are experiencing in practice, if the real level of inflexible load would constrain the learning of CVPO so that it cannot find policies yielding cost below the cost limit.

As CVPO does not guarantee transformer limit satisfaction, in practice a shield should be implemented to prevent the agents from creating transformer overloads. In this case, the fall-back mechanism would curtail the charging power of EV chargers when the aggregated power consumption exceeds the transformer's power limit. As user satisfaction is already suboptimal, and this shield will further reduce the performance of CVPO, more research is required to conclude whether CVPO could be applied in real EV charging scenarios similar to the setting of Experiment 2.

5

Conclusion

5.1. Answers to the research questions

How can an EV aggregator's charging and V2G profits be maximized using Reinforcement Learning, considering transformer limits, EV user preferences, current-dependent charging efficiencies, and uncertainty?

The thesis's main research question was raised after an introduction to the need for flexibility in the power grid and the potential flexibility of EV charging. Furthermore, workplace EV charging was identified as a good candidate for applying smart charging algorithms in general, but especially for congestion services. The following chapter, Chapter 2, summarized an extensive literature review on related studies. The most important findings from the literature review were that many articles oversimplify the EV charging optimization problem, and many RL algorithms have been used to address the problem. However, few of the articles applied Safe RL. Thus, the act of applying the most recent Safe RL algorithms to real-time charging control of an EV aggregator was identified to be promising in terms of profit improvement and constraint satisfaction, and was researched only to a limited extent in related works.

In the methodology chapter, the optimization problem was formulated in the context of a workplace parking lot. The objective was to increase the EV aggregators' profits while complying with user satisfaction and power transformer constraints. The EV behavior was modeled by distributions based on real-life measurements, increasing the relevance of the results by considering the uncertainty. Furthermore, the following subquestions were addressed:

1. How to model the transformer limit and EV user constraints?

The transformer power limit and EV target SOC for departure were modeled as soft constraints. The constraints were not strictly enforced by the environment, as doing so could hinder the learning capabilities of the RL agents. Instead, the agents had to learn to comply with them through reward subtractions (Classic RL) or constraint violation costs (Safe RL).

2. How to model current-dependent charging and discharging efficiencies?

The charging and discharging efficiencies were extracted from an article measuring the real three-phase charging efficiencies of the EVs used in this thesis. The discharging efficiencies were set equal to the measured charging efficiencies.

3. How to define the profit maximization problem as a Constrained Markov Decision Process (CMDP)?

The MDP was based on an article researching a similar optimization problem. However, the state and reward functions were updated through many iterations in a trial-and-error process. A key finding was that normalizing the terms in the state function improved the RL agents' learning capabilities. The soft constraints were included by heavy reward subtractions in the MDP in an attempt to guide the Classic RL agents toward constraint-satisfying behavior. The MDP was transformed into a CMDP by addressing constraint violations through a cost function and cost limit instead of reward subtractions. The cost

function was based on the reward subtractions due to constraint violations of the MDP reward function. Adequate factors for the cost function and cost limit were found through extensive experimentation.

Finally, in Chapter 4, the training and evaluation results were presented. Two experiments were performed, both with subexperiments such as sensitivity and scalability studies. In the setting of Experiment 1, PV power generation and inflexible loads were omitted. Consequently, transformer overloads could not happen. In the setting of Experiment 2, PV, inflexible loads, and transformer loads were included. Chapter 4 answered the last research subquestion:

4. How does the proposed method perform compared to baseline methods in experiments?

Safe RL algorithms were compared to Classic Deep RL algorithms, conventional AFAP charging, and an optimal offline solver in a case study of a business place parking lot with ten chargers. The proposed method, the Safe RL algorithm CVPO, proved superior to the other algorithms in terms of constraint satisfaction and scalability across the experiments. However, results indicate that SAC-L may perform better in less constrained problems. CVPO, CPO, and SAC-L all learned behavior that generalized well to unseen levels of charger occupation. In the training process, it was discovered that CVPO can be overly conservative in the constrained problem setting of this thesis. As a result, steering the algorithm towards profitable behavior required careful tuning of the cost function and the number of train environments.

No Classic RL algorithm achieved satisfactory user satisfaction in the more trivial case of Experiment 1, even when low user satisfaction was heavily penalized in the MDP reward function. As each Safe RL algorithm had higher user satisfaction than the Classic RL baselines, it is concluded that in the EV charging problem, Safe RL strongly improves constraint satisfaction compared to Classic Deep RL. Furthermore, the cost limit introduced by Safe RL proved to be an effective controllability tool for the trade-off between profit and constraint satisfaction. The modeling of the current-dependent three-phase charging efficiencies resulted in no added value in this thesis and may be omitted in further research.

CVPO showed a profit improvement of 22% in Experiment 1, the scenario where transformer overloads cannot happen. It is thus concluded that in such a setting, an EV aggregator could improve its profits compared to AFAP charging by applying a CVPO agent as a real-time charging controller. As the CVPO agents in Experiment 1 had a mean user satisfaction of 99% compared to 100% with AFAP, the EV aggregator would, on average, still have happy customers. However, in 11.6% of the 100 evaluation days of Experiment 1, at least one EV user had a departure SOC below 70%. Therefore, aggregators would have to implement a shield to guarantee high user satisfaction for every EV user in practice.

In Experiment 2, the problem became more complex not only because of an extra constraint, but also because transformer overloading events were sparse. CVPO proved more effective in learning to respect the transformer load from the sparse costs than SAC-L. Still, the optimization problem easily became too complicated for CVPO to learn rewarding and constraint-satisfying behavior. Therefore, more research should be done to conclude whether CVPO could improve real-time charging in scenarios with a possibility of transformer overloading.

5.2. Limitations and future work

The following limitations are recommended for consideration in future related work:

- **Modeling of the EV aggregator's profit**. The main limitation of this thesis is the modeling of the EV aggregator's profit. By using market electricity prices directly as charging and discharging prices, the real source of profit for the EV aggregator is excluded: the EV aggregator's margin on the charging cost or discharging profit. In future work, a pricing strategy would have to be considered to include the EV aggregator's profit margin on dynamic electricity prices in the problem formulation. For instance, a varying profit margin dependent on the electricity price could be formulated. The margin should be minimal if EVs are charged at a high price. The margin should increase for lower prices. Vice versa, when EVs are discharged at low electricity prices, the profit margin of the aggregator should be smaller than when EVs are discharged at high prices. Such a strategy would encourage algorithms to find charging schemes that benefit both the EV aggregator and the EV users. As a result of modeling the profit of EV aggregators more realistically, the interpretation of the results of numerical simulations would become more intuitive. Furthermore,

the profits would better translate to an EV aggregator's real profits.

- **Charging behavior of Safe RL**. Figure 4.6 showed that for at least one agent evaluated on one simulation day, CVPO and SAC-L did not correctly incorporate electricity price information in their policies. It should be investigated whether all Safe RL agents fail to process daily price signals, and merely learn one charging strategy–V2G in the morning, charging in the afternoon–that, on average, leads to good results.
- **Fairness of the algorithm**. The fairness of the algorithm is an important topic not mentioned in this thesis. EV aggregators may be interested in whether every EV user benefits equally from smart charging algorithms. Furthermore, the fairness of the profit distribution between aggregator and EV users could be interesting to examine.
- **One cost function**. As the Safe RL implementations of *fsrl* have one cost function, in this thesis, the two main constraints were addressed with one cost function. The performance of the Safe RL agents may increase when each constraint is handled by a distinct cost function.
- **EV Data**. A substantial part of the ElaadNL data comes from PHEVs. These vehicles' arrival and departure times are probably similar to those of BEVs. However, as these vehicles have smaller batteries, the SOC at arrival parameter is not representative of a parking lot where most charging sessions correspond to BEVs. This thesis may be a starting point for future work where a fleet of only BEVs is more accurately modeled. If the researchers of these works also fail to find more accurate BEV data, ElaadNL could be approached to determine how much of the workplace data is from PHEVs. Then, the PHEV data could be compensated by increasing the average energy demand with a scaling factor.
- **Battery degradation**. Also, battery degradation is not considered in this thesis. The impact on battery degradation should be researched before Safe RL agents can be applied in practice.
- **Different target SOC for departure**. In this thesis, it is assumed that every EV user desires a departure SOC of 100%, while in reality this may vary per EV user. The effect of different desired departure SOC values, sampled from a distribution between 80% and 100%, for instance, could be interesting to investigate.
- **Minimum V2G SOC**. The minimum V2G SOC prevents EV users from having near-empty batteries in case of unexpected emergencies. It should be at a balance between good V2G profit potential and low range anxiety. This thesis assumed 50% to be an effective minimum V2G SOC, but this was not based on other articles.
- **Modeling of the inflexible loads and PV power generation**. The modeling of the inflexible loads and PV power generation severely limited how realistic Experiment 2 was. The load data was based on households instead of office buildings, the forecast errors for both the PV generation and the loads are not derived from real data, and the PV data was based on the aggregated PV power of the Netherlands instead of a small local PV system. If possible, future work should base the modeling of the loads and PV on more suitable data.
- **Size of the case study**. In reality, EV aggregators regularly encounter large parking lots with more than ten charging points. While the results of Safe RL applied to this case study are promising, the scalability of CVPO to parking lots with 30 chargers seems limited. In future work, more experiments should be conducted in settings with a larger number of chargers to research CVPO's performance at parking lots with more chargers.

References

[1] IEA, *Renewables 2023*, 2024. [Online]. Available: https://www.iea.org/reports/renewables-2023.

[2] IEA, *Global ev outlook 2024*, 2024. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2024.

[3] IEA, *Integrating solar and wind*, 2024. [Online]. Available: https://www.iea.org/reports/integrating-solar-and-wind.

[4] P. Denholm and M. Hand, "Grid flexibility and storage required to achieve very high penetration of variable renewable electricity," *Energy Policy*, vol. 39, no. 3, pp. 1817–1830, 2011, ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2011.01.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421511000292.

[5] P. D. Lund, J. Lindgren, J. Mikkola, and J. Salpakari, "Review of energy system flexibility measures to enable high levels of variable renewable electricity," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 785–807, 2015, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.01.057. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115000672.

[6] A. Stawska, N. Romero, M. Weerdt, and R. Verzijlbergh, "Demand response: For congestion management or for grid balancing?" *Energy Policy*, vol. 148, p. 111 920, Jan. 2021. DOI: 10.1016/j.enpol.2020.111920.

[7] ElaadNL, *Analyse van laadprofielen voor ev's bij thuis-, publiek-, werk- en snelladen in 2030.* 2023. [Online]. Available: https://elaad.nl/onderzoek-naar-laadprofielen-geeft-inzicht-in-belasting-stroomnet-door-laden-elektrische-autos-en-effect-van-slim-laden/.

[8] Tennet, *Market roles*. [Online]. Available: https://www.tennet.eu/market-roles.

[9] Tennet, *Frequently asked questions congestion studies*, 2022. [Online]. Available: https://www.tennet.eu/markets/dutch-market/studies-congestion-management.

[10] F. Tanrisever, K. Derinkuyu, and G. Jongen, "Organization and functioning of liberalized electricity markets: An overview of the dutch market," *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1363–1374, 2015, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.07.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032115006668.

[11] ENTSOE, *Transparency platform*. [Online]. Available: https://newtransparency.entsoe.eu/.

[12] J. Pecinovsky and F. Boerman, *entsoe-py*. [Online]. Available: https://github.com/EnergieID/entsoe-py.

[13] GOPACS, *Congestion management products*. [Online]. Available: https://en.gopacs.eu/congestion-management-products/.

[14] F. M. N. U. Khan, M. G. Rasul, A. Sayem, and N. K. Mandal, "Design and optimization of lithium-ion battery as an efficient energy storage device for electric vehicles: A comprehensive review," *Journal of Energy Storage*, vol. 71, p. 108 033, 2023, ISSN: 2352-152X. DOI: https://doi.org/10.1016/j.est.2023.108033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X23014305.

[15] M. S. E. Houache, C.-H. Yim, Z. Karkar, and Y. Abu-Lebdeh, "On the current and future outlook of battery chemistries for electric vehicles—mini review," *Batteries*, vol. 8, no. 7, 2022, ISSN: 2313-0105. [Online]. Available: https://www.mdpi.com/2313-0105/8/7/70.

[16] F. Mohammadi and M. Saif, "A comprehensive overview of electric vehicle batteries market," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 3, p. 100 127, 2023, ISSN: 2772-6711. DOI: https://doi.org/10.1016/j.prime.2023.100127. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772671123000220.

[17] ElaadNL, *Nationaal laadonderzoek 2023*, 2023. [Online]. Available: https://elaad.nl/elektrische-rijder-wil-bi-directioneel-kunnen-laden/.

[18] M. Kühnbach, M. Klobasa, A. Stephan, *et al.*, *Potential of a full ev-power-system-integration in europe and how to realise it. study on behalf of transport & environment (t&e) europe.* 2024. [Online]. Available: https://www.transportenvironment.org/articles/batteries-on-wheels-the-untapped-potential-of-ev-batteries.

[19] CBS, *Statline: Elektriciteitsbalans; aanbod en verbruik*. [Online]. Available: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/84575NED/table?dl=92781.

[20] H. M. Abdullah, A. Gastli, and L. Ben-Brahim, "Reinforcement learning based ev charging management systems–a review," *IEEE Access*, vol. 9, pp. 41 506–41 531, 2021. DOI: 10.1109/ACCESS.2021.3064354.

[21] N. I. Nimalsiri, C. P. Mediwaththe, E. L. Ratnam, M. Shaw, D. B. Smith, and S. K. Halgamuge, "A survey of algorithms for distributed charging control of electric vehicles in smart grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4497–4515, 2020. DOI: 10.1109/TITS.2019.2943620.

[22] D. Qiu, Y. Wang, W. Hua, and G. Strbac, "Reinforcement learning for electric vehicle applications in power systems: A critical review," *Renewable and Sustainable Energy Reviews*, vol. 173, p. 113 052, 2023, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2022.113052. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032122009339.

[23] S. S. A. Salam, V. Raj, M. I. Petra, A. K. Azad, S. Mathew, and S. M. Sulthan, "Charge scheduling optimization of electric vehicles: A comprehensive review of essentiality, perspectives, techniques, and security," *IEEE Access*, vol. 12, pp. 121 010–121 034, 2024. DOI: 10.1109/ACCESS.2024.3433031.

[24] J. Álvarez, M. Á. González Fernández, C. Vela, and R. Arias, "Electric vehicle charging scheduling by an enhanced artificial bee colony algorithm," *Energies*, vol. 11, p. 2752, Oct. 2018. DOI: 10.3390/en11102752.

[25] S. Xu, D. Feng, Z. Yan, *et al.*, "Ant-based swarm algorithm for charging coordination of electric vehicles," *International Journal of Distributed Sensor Networks*, vol. 2013, Jan. 2013. DOI: 10.1155/2013/268942.

[26] J. Liu, G. Lin, S. Huang, Y. Zhou, Y. Li, and C. Rehtanz, "Optimal ev charging scheduling by considering the limited number of chargers," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 3, pp. 1112–1122, 2021. DOI: 10.1109/TTE.2020.3033995.

[27] B. Canol, C. Andic, M. Purlu, and B. E. Turkay, "Optimum energy management in electric vehicle parking lots using heuristic methods," in *2022 4th Global Power, Energy and Communication Conference (GPECOM)*, 2022, pp. 473–477. DOI: 10.1109/GPECOM55404.2022.9815675.

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning, Second Edition: An Introduction*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts, 2018, ISBN: 978-0-262-03924-6.

[29] S. Ayyadi, H. Bilil, and M. Maaroufi, "Optimal charging of electric vehicles in residential area," *Sustainable Energy, Grids and Networks*, vol. 19, p. 100 240, 2019, ISSN: 2352-4677. DOI: https://doi.org/10.1016/j.segan.2019.100240. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352467719300761.

[30] A.-M. Koufakis, E. S. Rigas, N. Bassiliades, and S. D. Ramchurn, "Offline and online electric vehicle charging scheduling with v2v energy transfer," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2128–2138, 2020. DOI: 10.1109/TITS.2019.2914087.

[31] A. Dukpa and B. Butrylo, “Milp-based profit maximization of electric vehicle charging station based on solar and ev arrival forecasts,” *Energies*, vol. 15, no. 15, 2022, ISSN: 1996-1073. DOI: 10.3390/en15155760. [Online]. Available: https://www.mdpi.com/1996-1073/15/15/5760.

[32] Q. Chen, F. Wang, B.-M. Hodge, *et al.*, “Dynamic price vector formation model-based automatic demand response strategy for pv-assisted ev charging stations,” *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2903–2915, 2017. DOI: 10.1109/TSG.2017.2693121.

[33] S. Orfanoudakis, C. Diaz-Londono, Y. E. Yılmaz, P. Palensky, and P. P. Vergara, *Ev2gym: A flexible v2g simulator for ev smart charging research and benchmarking*, 2024. arXiv: 2404.01849 [cs.SE]. [Online]. Available: https://arxiv.org/abs/2404.01849.

[34] C. Diaz-Londono, S. Orfanoudakis, P. P. Vergara, P. Palensky, F. Ruiz, and G. Gruosso, *A simulation tool for v2g enabled demand response based on model predictive control*, 2024. arXiv: 2405.11963 [eess.SY]. [Online]. Available: https://arxiv.org/abs/2405.11963.

[35] M. Saleem, S. Saha, U. Izhar, and L. Ang, “A stochastic mpc-based energy management system for integrating solar pv, battery storage, and ev charging in residential complexes,” *Energy and Buildings*, vol. 325, p. 114 993, 2024, ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2024.114993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778824011095.

[36] Y. Yang, H.-G. Yeh, and R. Nguyen, “A robust model predictive control-based scheduling approach for electric vehicle charging with photovoltaic systems,” *IEEE Systems Journal*, vol. 17, no. 1, pp. 111–121, 2023. DOI: 10.1109/JSYST.2022.3183626.

[37] M. Dorokhova, Y. Martinson, C. Ballif, and N. Wyrsch, “Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation,” *Applied Energy*, vol. 301, p. 117 504, 2021, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.117504. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921008874.

[38] Z. Wan, H. Li, H. He, and D. Prokhorov, “Model-free real-time ev charging scheduling based on deep reinforcement learning,” *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, 2019. DOI: 10.1109/TSG.2018.2879572.

[39] F. Zhang, Q. Yang, and D. An, “Cddpg: A deep reinforcement learning-based approach for electric vehicle charging control,” *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, Aug. 2020. DOI: 10.1109/JIOT.2020.3015204.

[40] Y. Jiang, Q. Ye, B. Sun, Y. Wu, and D. H. Tsang, “Data-driven coordinated charging for electric vehicles with continuous charging rates: A deep policy gradient approach,” *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12 395–12 412, 2022. DOI: 10.1109/JIOT.2021.3135977.

[41] F. L. D. Silva, C. E. H. Nishida, D. M. Roijers, and A. H. R. Costa, “Coordination of electric vehicle charging through multiagent reinforcement learning,” *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2347–2356, 2020. DOI: 10.1109/TSG.2019.2952331.

[42] Z. Zhang, Y. Wan, J. Qin, W. Fu, and Y. Kang, “A deep rl-based algorithm for coordinated charging of electric vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 774–18 784, 2022. DOI: 10.1109/TITS.2022.3170000.

[43] J. Fan, H. Wang, and A. Liebman, “Marl for decentralized electric vehicle charging coordination with v2v energy exchange,” in *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023, pp. 1–6. DOI: 10.1109/IECON51785.2023.10312315.

[44] Y. Guan, J. Zhang, W. Ma, and L. Che, “Rule-based shields embedded safe reinforcement learning approach for electric vehicle charging control,” *International Journal of Electrical Power & Energy Systems*, vol. 157, p. 109 863, 2024, ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2024.109863. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014206152400084X.

[45] J. Achiam, D. Held, A. Tamar, and P. Abbeel, *Constrained policy optimization*, 2017. arXiv: 1705.10528 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1705.10528.

[46] H. Li, Z. Wan, and H. He, "Constrained ev charging scheduling based on safe deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2427–2439, 2020. DOI: 10.1109/TSG.2019.2955437.

[47] G. Chen and X. Shi, *A deep reinforcement learning-based charging scheduling approach with augmented lagrangian for electric vehicle*, 2022. arXiv: 2209.09772 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2209.09772.

[48] J. Zhang, Y. Guan, L. Che, and M. Shahidehpour, "Ev charging command fast allocation approach based on deep reinforcement learning with safety modules," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 757–769, 2024. DOI: 10.1109/TSG.2023.3281782.

[49] ElaadNL, *Elaadnl open datasets for electric mobility research*. [Online]. Available: https://platform.elaad.io/analyses/ElaadNL_opendata.php.

[50] RVO, *Elektrische personenauto's in nederland*. [Online]. Available: https://duurzamemobiliteit.databank.nl/mosaic/nl-nl/elektrisch-vervoer/personenauto-s.

[51] K. Sevdari, L. Calearo, B. H. Bakken, P. B. Andersen, and M. Marinelli, "Experimental validation of onboard electric vehicle chargers to improve the efficiency of smart charging operation," *Sustainable Energy Technologies and Assessments*, vol. 60, p. 103 512, 2023, ISSN: 2213-1388. DOI: https://doi.org/10.1016/j.seta.2023.103512. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138823005052.

[52] M. Esser, S. Orfanoudakis, O. Homaee, V. Vahidinasab, P. P. Vergara, and A. Spina, "Paving the way for electric vehicle mass deployment: A dataset of unidirectional, bidirectional, and dynamic charging profiles," 2024. DOI: 10.36227/techrxiv.173202789.91594134/v1.

[53] *Renewables-ninja*. [Online]. Available: https://github.com/renewables-ninja.

[54] *Pecan street data portal*. [Online]. Available: https://www.pecanstreet.org/dataport/.

[55] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html.

[56] Z. Liu, Z. Cen, V. Isenbaev, *et al.*, *Constrained variational policy optimization for safe reinforcement learning*, 2022. arXiv: 2201.11927 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2201.11927.

[57] Z. Liu and Z. Guo, *Fast safe reinforcement learning (fsrl)*. [Online]. Available: https://github.com/liuzuxin/FSRL.

[58] A. Stooke, J. Achiam, and P. Abbeel, *Responsive safety in reinforcement learning by pid lagrangian methods*, 2020. arXiv: 2007.03964 [math.OC]. [Online]. Available: https://arxiv.org/abs/2007.03964.

[59] D. H. P. C. C. (DHPC), *DelftBlue Supercomputer (Phase 2)*, https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

A

Appendix: python scripts

Some Python scripts used to generate figures or values for this thesis are shown here.

A.1. entsoe_loader.py

This script loads all 2024 electricity prices from the ENTSOE-E dataset.

```python
from entsoe import EntsoePandasClient
import pandas as pd

client = EntsoePandasClient(api_key='INSERT␣API␣KEY')

start = pd.Timestamp('20240101', tz='Europe/Brussels')
end = pd.Timestamp('20250101', tz='Europe/Brussels')
country_code = 'NL'
type_marketagreement_type = 'A01'
contract_marketagreement_type = "A01"

ts = client.query_day_ahead_prices(country_code, start=start, end=end)
ts.to_csv('prices_2024.csv')

df = pd.read_csv('prices_2024.csv')
print(df.head())

mean_price = df['0'].mean()
print('mean␣price␣2024:', mean_price)
```

A.2. boxplotter.py

This script generates Figure 1.1.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def boxplots(
        file_path,
        datetime_col='Datetime␣(Local)',
        price_col='Price␣(EUR/MWhe)'
    ):

    df = pd.read_csv(file_path)

    # Convert the datetime column to a proper datetime object
    df[datetime_col] = pd.to_datetime(df[datetime_col], format='mixed')

    # Drop rows with invalid or missing dates
    df.dropna(subset=[datetime_col], inplace=True)

```

```
    df['Year'] = df[datetime_col].dt.year
    df = df[(df['Year'] > 2016) & (df['Year'] < 2025)]


    # Group by year and hour, then calculate the mean price per hour
    df['Hour'] = df[datetime_col].dt.hour
    grouped = df.groupby(['Year', 'Hour'])[price_col].mean().reset_index()

    plt.figure(figsize=(10, 6))

    sns.boxplot(data=grouped, x='Year', y=price_col, palette='Set2')

    plt.title('Average electricity prices in the Netherlands', fontsize=16, family='Arial')
    plt.xlabel('Year', fontsize=12, family='Arial')
    plt.ylabel('Day-Ahead Price (EUR/MWh)', fontsize=12, family='Arial')
    plt.tight_layout()
    plt.show()

boxplots(file_path='Netherlands_day-ahead-2015-2024.csv')
```

A.3. weighted_mean_EV_battery.py

This script computes the weighted mean battery size of the EVs used in this thesis.

```
import numpy as np

registrations = np.array([47783, 39216, 28028, 23033, 21186, 19815, 19307, 17785, 16449,
    14545])
capacities = np.array([57.5, 57.5, 64.8, 58, 58, 64, 66, 46.3, 77, 52])

weighted_mean = np.sum(registrations * capacities) / np.sum(registrations)
print(weighted_mean)
```