Efficient Procurement and Trading of Electric Vehicle Charging Flexibility

by

Javier Sales-Ortiz

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

This thesis studies a virtual power plant (VPP) that trades the bidirectional charging flexibility of privately owned plug-in electric vehicles (EVs) in a real-time electricity market to maximize its profit. The main contribution of this thesis is the development of scalable and efficient algorithms for the procurement and scheduling of this flexibility. Specifically, to incentivize EVs to allow bidirectional charging, we design incentive-compatible, variable-term contracts between the VPP and EVs. Through deliberate aggregation of the energy storage capacity of individual EVs, we construct an abstraction of the aggregate flexibility that can be provided by the connected EVs. This abstraction is called a virtual battery and its operation is scheduled in real-time by learning a reinforcement learning (RL) policy. This policy efficiently trades the available flexibility, independent of the number of accepted contracts and connected EVs. The proposed aggregation method ensures the satisfaction of individual EV charging requirements by constraining the optimal action returned by the RL policy within certain bounds. We then develop a disaggregation scheme to allocate power to bidirectional chargers. We formulate this as a resource allocation problem, in which the total amount of energy traded in the market is distributed in a proportionally fair manner among the connected EVs. Evaluation on a real-world dataset demonstrates robust performance of the proposed method despite high variability of electricity prices and shifts in the distribution of EV mobility.

Preface

This thesis is based on two conference papers. The first one was authored by myself, and former M.Sc. student Saidur Rahman (both of us with equal contribution), along with my supervisor. This was published in ACM e-Energy 2023 [70]. The second was authored by myself and my supervisor, and it is recently accepted for publication in the upcoming ACM e-Energy 2024 [77].

In the first paper [69], I contributed the analysis and interpretation of one-dimensional V2G contracts, considerable work on the implementation, and the final experimentation process. In this thesis, I do not include the contributions made by my co-author, Saidur Rahman. These were the initial design of one-dimensional contracts, proofs of equivalence for the simplified formulation of one-dimensional contract design problem, and trading strategies in the day-ahead market.

I was primarily responsible for the contributions of the second conference paper [77], developed under the guidance of Prof. Ardakanian. These were: the development of two-dimensional V2G contracts (Chapter 3), revised VPP and EV charging models (Chapter 4), an efficient and scalable reinforcement learning agent (Chapter 5), experimental validation on two real-world datasets (Chapter 6), and a new streamlined codebase for the implementation of our methods.

Furthermore, this thesis expands on the second paper [77] by developing proofs that simplify the formulation of two-dimensional contracts (Chapter 3). These are based on the ones that Saidur presented for one-dimensional contracts in his thesis [69]. Moreover, it expands on the literature review (Chapter 2), presents additional experiments (Chapter 6), and provides a brief overview of the accompanying codebase (Appendix A).

All of the experiments and the development of the accompanying code repository for this thesis were done by myself. The code repository for the second paper [77] and this thesis is completely independent of the one that was released together with the first paper [70].

"In theory there is no difference between theory and practice.
In practice there is."

– Yale Literary Magazine, 1882.

Acknowledgements

First and foremost, I express my gratitude to my supervisor, Dr. Omid Ardakanian, for his exceptional guidance and invaluable advice throughout this journey. My Master's thesis and other publications would not have come to fruition without his support. Under his mentorship, I have gained essential insights into critical problem-solving, clarity in writing and dedication to my work, lessons that will undoubtedly shape my approach to future challenges.

My research was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) through the Alliance Program, and the Canada First Research Excellence Fund through the University of Alberta's Future Energy Systems (FES).

I also thank to my fellow students at CAB-385, namely Aakash, Hossein, Sanku, Afia, Tianyu, Xin, Sanju, Vidushi, Ioanna, Stephan and Jihoon. Working side by side with each of you has been truly inspiring and has fueled my dedication to my projects. I have greatly valued your friendship throughout this process.

Finally, I would like to express my heartfelt gratitude to my family–Carlos, Cordelia, Constanza, Julieta, and Laura, and to my girlfriend, Alex, for their unwavering love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern electrical grids face several pressing challenges. The rise of renewable energy sources has made power generation more intermittent and less predictable [44]. This intermittency forces the grid operator to rapidly adjust the output of their generating plants to match demand, which may require burning more fossil fuels (e.g. natural gas) to ramp up quickly [94]. Conversely, during periods of excess energy production, that extra energy goes to waste if it cannot be exported. A notable example of these challenges is illustrated by the "duck curve" phenomenon [19]. The example shown in Figure 1.1 is taken from California's grid operator (CAISO). The graph shows the net load, i.e. the demand that the operator needs to cover after subtracting solar generation. We observe reduced load, even negative, during the midday due to increased solar production. However, in the morning there is a sudden dip, and there is a precipitous increase followed by a peak in the evening.

Implementing storage solutions in the grid can address these issues [29]. In particular, short-term storage can balance supply and demand, improving overall grid reliability. It can also be used to shave peaks in demand [14] and smooth out fluctuations in renewable energy generation. Leveraging the storage capacity of electric vehicles (EVs) while they are connected to charging stations has emerged as a viable grid storage solution, especially considering the significant decline in the cost of EV lithium-ion batteries. These have dropped by over 97% since 1991 [72]. The need to control EV charging sessions becomes even more pressing when we consider that most EVs are plugged in

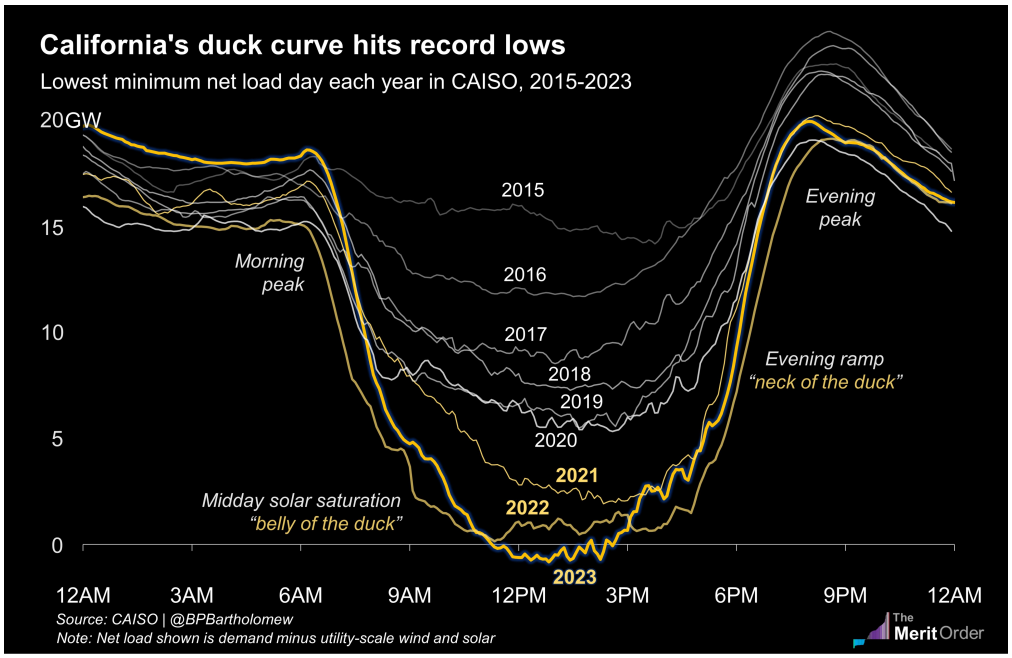

Figure 1.1: Evolution of California's duck curve 2015-2023, source [10]

when people come home after work during peak demand times [7].

EVs are positioned to serve as crucial assets for short-term grid storage. By the end of 2025, the USA is expected to have over 111 GWh of grid-scale energy storage capacity [79]. At the same time, there are 2.5 million EVs already operating in that country [40]. With a back-of-the-envelope calculation and assuming each EV has an 80 kWh battery capacity[1], this amounts to 200 GWh of potential storage. Indeed, not all of this can be accessed by the grid, and EVs are not suitable for long-term storage. Nonetheless, this highlights the significant role that EVs can play. Factors such as monetary and non-monetary incentives, consumer participation, guaranteeing charging deadlines, and intelligent (and possibly bidirectional) charging strategies will determine their impact. This thesis directly tackles these pressing challenges by studying how to procure and schedule the flexibility that can be offered by EVs.

[1]This assumption is explained in Section 6.1

1.1 Virtual Power Plants

Climate change has increased the frequency and intensity of extreme weather events, such as heat waves, wildfires, hurricanes, ice storms, and floods. In recent years, some of these events caused a grid emergency and left millions without power for several hours [89] due to loss of generation capacity, spiking demand for electricity, or weather-related impacts on transmission and distribution systems. Virtual power plants (VPP) – networks of distributed energy resources that are aggregated and controlled to serve the grid – could strengthen the resilience and reliability of the grid in the face of extreme weather events [24]. When these resources are owned by individual customers rather than power utilities, the VPP offers a low-cost alternative to infrastructure upgrades that would otherwise be necessary to improve grid reliability.

An emerging type of VPP aggregates privately owned plug-in EVs connecting to residential or public chargers to replenish their battery. This VPP can take advantage of three kinds of flexibility offered by EV chargers to participate in one or multiple electricity markets. First, chargers can regulate their power within certain bounds allowing the charging demand to be shaped. Second, EVs typically remain connected to the charger longer than is needed for their battery to fully charge. This makes it possible to shift the charging demand in time. Finally, as bidirectional chargers with vehicle-to-grid (V2G) functionality become available on the market [12], the EV battery can be discharged for some time before it is charged to the desired state-of-charge (SOC). Compared to VPPs that control a fleet of EVs owned by a company or city [68], e.g. electric taxis or buses, this VPP is capable of offering sizable flexibility to the grid because it can potentially control thousands of privately owned EVs in a large area. An illustration of this type of VPP, including our proposed flexibility abstraction layer, is shown in Figure 2.1.

Interest in V2G technology is at an all-time high. EVs that support bidirectional charging, such as the Ford F-150 Lighting and the Nissan Leaf are already available in the market [9]. GM has announced that all its electric models will support this feature by 2026 [34], with a similar commitment from

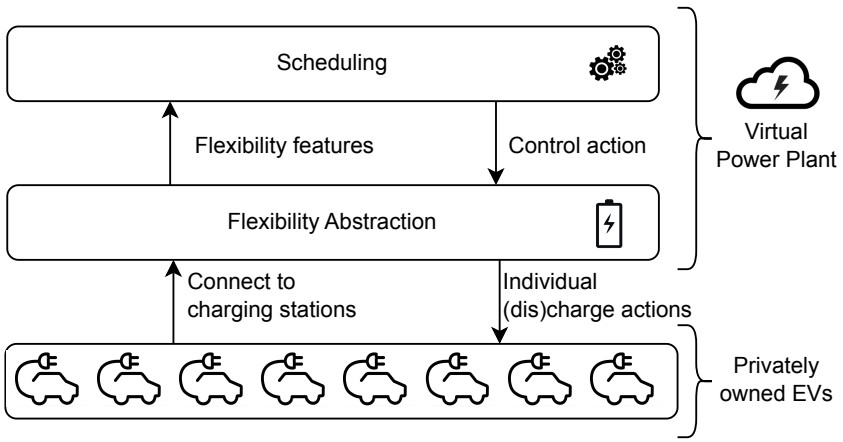

Figure 1.2: Multiple EV charging sessions can be coordinated by a VPP

Tesla [57]. Hyundai initiated a pilot project with 25 V2G enabled IONIQ 5 vehicles in Utrecht, Netherlands [43], while Toyota has ongoing V2G research projects with major utility companies in Texas and California [21], [82]. In 2023, Ford, Honda and BMW unveiled plans to establish a V2G company aimed at connecting electric utilities and private EV owners [39]. Numerous other companies like Nissan, Volvo and Polestar are also involved in V2G projects [17], [61], [95]. Octopus Energy, a utilities company in the UK, already offers a V2G tariff for customers charging at home, estimating potential savings of up to £880 (1,500 CAD) per year for each user. Customer interest is evident, with reports of EV owners supporting local grids during contingencies, as illustrated by a Sydney resident who earned $100 in just two hours during a heatwave in early 2024 [98]. Additionally, algorithms that control charging to support the grid might already exist in your pocket. In late 2023, Apple introduced "Clean Energy Charging" to selectively charge iPhones in a way that reduces their carbon footprint[2] [2].

1.2 Challenges

Despite the vast potential of V2G VPPs, ensuring its efficient operation is extremely challenging, especially as the number and diversity of pooled resources increases. This is mainly due to two reasons. First, incentivizing privately

[2]To check, go to Settings > Battery > Battery Health & Charging

owned EVs to allow their battery to be charged and discharged at variable rates is difficult because of individual differences in appraising flexibility and battery degradation costs, and concerns about whether their energy demand would be fulfilled before departure (feasibility concern) and their battery would be discharged by the same amount as others who received the same incentive (fairness concern). Second, volatile prices and stochastic EV mobility make it difficult to guarantee that the available flexibility can be traded efficiently in an electricity market. To address these challenges, in our previous work [70] we designed fixed-term V2G contracts that are offered to EVs upon arrival at the charging station, and develop an online scheduling algorithm to trade their charging flexibility in the imbalance market given knowledge of the accepted contracts and price forecasts. However, fixed-term contracts are too strict, either preventing many electric vehicles from participating in the VPP (when the contract term is too long) or failing to fully utilize their charging flexibility (when the contract term is too short). Additionally, in the real world, price forecasts for several hours in the future are inaccurate, so flexibility cannot be optimally managed by solving a deterministic optimization problem.

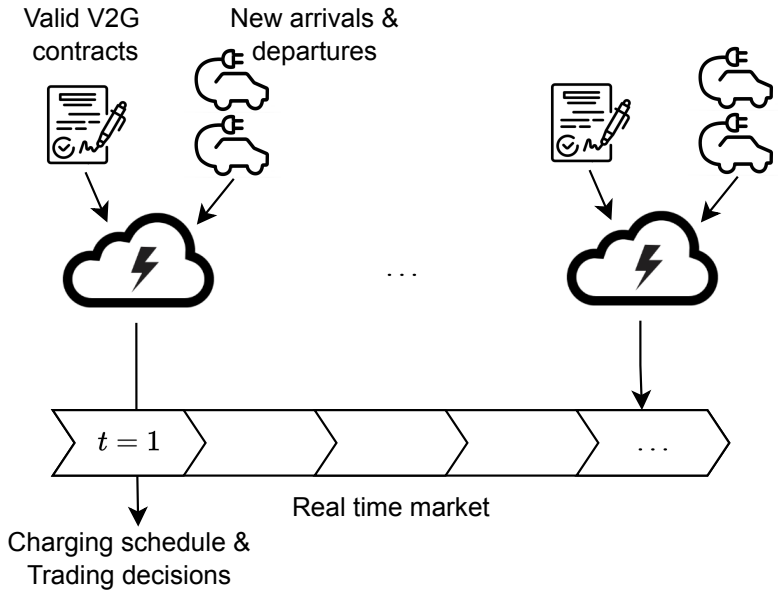

Figure 1.3: Our VPP takes in EVs, V2G contracts, and trades in the market.

1.3 Contributions

We design a set of incentive-compatible, variable-term V2G contracts that optimize the expected utility of the VPP. These contracts are offered to each EV as soon as it connects to the charger, allowing it to choose the contract that is best for it. Each contract signifies the VPP's commitment to EV owners that it will pay them a certain amount of money for discharging a certain amount of energy from their battery over the contract term. The VPP then uses a scalable and efficient reinforcement learning (RL) policy to buy electricity to supply the charging demand of all EVs and effectively trade the flexibility in real-time.[3] Since the available flexibility depends on the number of EVs that have accepted a contract, making decisions for each EV independently increases the dimension of the action space, making it difficult to (a) learn a good policy when the number of EVs varies in the dataset, and (b) ensure fairness among EVs that accepted the same V2G contract. To overcome these challenges, our flexibility management approach involves aggregating the energy storage capacity of all EVs into a *virtual battery*, real-time scheduling of the virtual battery operation and accordingly trading flexibility in aggregate in a real-time market, and finally disaggregating the (dis)charge power of the virtual battery into the (dis)charge power of individual EVs, while ensuring fairness and feasibility of these schedules. A diagram showing the overall structure of the VPP can be seen in Figure 1.3. We make five specific contributions in this work:

- To incentivize privately-owned EVs to join the coalition, i.e. the VPP, we design self-revealing and variable-term V2G contracts based on the principal-agent model [13] and by extending the agent type to two dimensions. Using realistic parameters for the utility function of the VPP and EV owners, we get nine distinct contracts with three different terms.

- We learn a reinforcement learning policy for trading energy every hour in an electricity market using the single-agent soft actor-critic algorithm

[3] By buying or selling more/less energy, the VPP offers bidirectional flexibility to the grid.

that respects time-varying constraints on the action. We show that these constraints can be efficiently calculated by aggregating the laxity and contract-related constraints of individual EV owners. As a result, the learned policy makes reasonable trading decisions irrespective of the number of EVs controlled by the VPP and enforces charging deadlines and terms of accepted contracts.

- We draw a parallel between flexibility disaggregation and resource allocation, as both of them involve self-interested parties, and borrow the notion of proportional fairness from resource allocation to design a fair and efficient flexibility disaggregation algorithm. We compare this algorithm with priority-based disaggregation algorithms that incorporate the notion of laxity.

- We evaluate our real-time scheduling approach using real data from a network of public charging stations and prices from an imbalance market. Our result reveals that, despite distribution shifts, noisy forecasts, and changes in the number of connected EVs, the learned policy achieves comparable performance with offline optimization problems that receive the same forecasts.

- We develop a gym-like environment, dubbed *EvChargeGym*, that simulates EV arrivals, charging and departures. We use this to run our experiments and validate our methods. This environment is open source and publicly available, more details are given on Chapter 5.

1.4 Outline

The remainder of this thesis is structured as follows. In Chapter 2, we conduct a literature review, identifying gaps in the literature, and pointing out the works that formed the basis of this work. Chapter 3 explains the design of incentive-compatible V2G contracts, accomplished by constructing and solving an optimization problem. We scrutinize and simplify the constraints, and analyze the resulting optimal contracts. Subsequently, Chapter 4 details the operation of the VPP and the dynamics of EV charging. Specifically, we explore optimal VPP operation while adhering to constraints from the EV charging dynamics and V2G contracts. In Chapter 5, we present the scalable reinforcement learning agent that trades flexibility in aggregate, along with its respective aggregation and disaggregation algorithms. Next, Chapter 6 empirically validates our claims by comparing our proposed methods against other baselines using real-world datasets under various scenarios. Finally, we present our conclusions, discussion about limitations, and ideas for future work in Chapter 7.

Chapter 2

Related Work

In this chapter, we conduct a literature review. We identify the strengths and weaknesses of existing solutions and analyze how our approach contributes to the existing body of knowledge. The chapter is organized as follows. First, we introduce and define the concept of a Virtual Power Plant (VPP) to set the context. Next, we examine the various types of electricity markets in which a VPP can participate. Then, we delve into general approaches for flexibility management in the power grid, focusing on those specifically designed for EVs. We explore how some of these approaches utilize an abstraction to model flexibility, and review existing literature on disaggregation, drawing parallels to the field of resource allocation in computer networks. Subsequently, we shift our attention to Reinforcement Learning (RL) for EV charging, proposing a taxonomy to categorize different approaches. We also discuss methods for ensuring safety in RL, i.e. how agents can adhere to hard constraints. Lastly, we explore approaches to incentivize private users to engage in V2G, with a specific emphasis on contract theory.

2.1 Virtual Power Plants

Virtual Power Plants (VPPs) aggregate multiple distributed energy resources (DERs) into a single system [73]. This provides many advantages, among them, each DER will be able to access energy markets that support dealing in larger volumes than those available to individual consumers. This enables them to maximize their revenue by optimizing their position in the market

with the help of the VPP's market intelligence. Furthermore, this can simplify the grid operator's task as it can treat the VPP as a single entity instead of managing each DER individually. Lastly, as all DERs can be utilized and optimized, the power grid as a whole will benefit [66].

The VPPs that combine the flexibility of multiple resources of the same or different types and subsequently trade in the electricity market(s) are extensively studied in the literature [25], [68], [70], [93], [96]. This includes VPPs that aggregate wind and solar generating plants, battery energy storage systems, and EVs.

2.1.1 Participation of VPPs in Electricity Markets

VPPs offer a significant benefit to distributed energy resource owners by accessing the wholesale electricity markets on their behalf [59]. Electricity markets can be categorized based on their operating timescale, which can range from years to minutes. The definition of these markets varies across regions and evolves over time. There is a specific phenomenon known as energy market liberalization, this entails making regulations and systems that reduce barriers to entry, and promote fair competition among market players [32]. It aims to enhance operational efficiency and incentivize broader goals such as reducing electricity costs and meeting decarbonization targets. Energy market liberalization has been a longstanding trend since before the turn of the century [46]. Below, we provide a brief overview of electricity markets based on the survey paper [59].

Before Dispatch These markets operate in a scale ranging from weeks to years. An example is bilateral contracts, where buyers and sellers agree on a specified amount of energy to be used at a future date. Additionally, energy can be traded in the form of financial products in the futures and forward markets which helps avoid electricity market price uncertainty.

Day Before Dispatch In this category we find the day-ahead market, which enables the sale and purchase of electricity for each hour of the following day

through energy bids. Additionally, ancillary services markets ensure security and reliability by maintaining the balance between generation and demand. Moreover, the reserve market handles additional generation reserves to guarantee that demand is met, a role becoming increasingly critical due to the expansion of non-dispatchable renewable energy sources such as wind and solar.

Day of Dispatch During the operation day, intraday markets aim to adjust the energy transactions of the day-ahead market, as they have more information about electricity production and consumption. Typically, these markets deal with smaller trading volumes compared to the day-ahead market. The final market responsible for balancing supply and demand is the real-time or imbalance market, operating on a timescale ranging from a few minutes to an hour. Real-time imbalance markets may adopt dual-price mechanisms, where different prices per kWh are set for buying and selling. However, there has been a recent transition towards single-price markets, where unit prices remain consistent for both buying and selling, with the purpose of streamlining market operations. To name a few examples, both the European (ENTSO-E) and Nordic Transmission System Operators have opted for single-price imbalance markets [30], [31]. Likewise, our study focuses on real-time markets with single-pricing scheme. These short-term electricity markets have an increasingly important role to play due to the higher penetration of renewable generation and its unpredictable nature.

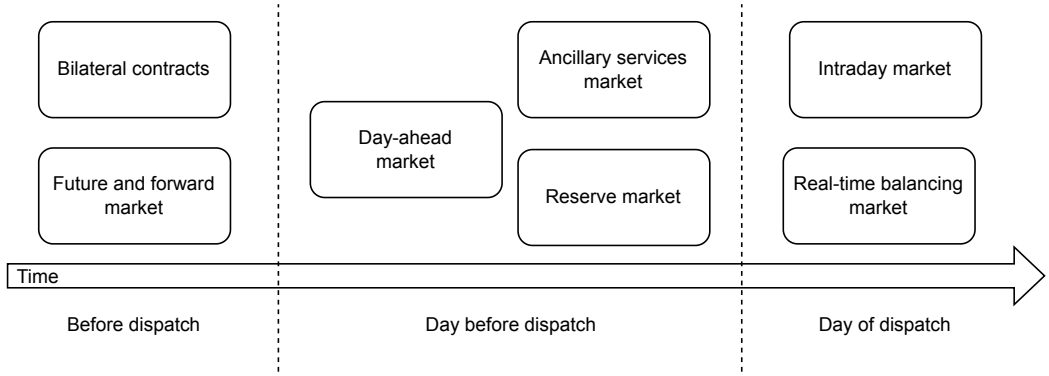

Figure 2.1: Different electricity markets sorted by timescale, adapted from [59]

2.1.2 Flexibility Management

In the context of the power grid, *flexibility* is defined as the ability of a resource, or a set of resources, to adjust their consumption or production over time in accordance with an external signal [50]. In the same work [50], the authors attempt to formally classify different definitions of flexibility based on their mathematical characteristics. The precise definition of flexibility greatly depends on the resources available, the participants that are considered, and the specific project objectives. A versatile way of modeling flexibility is using FlexOffers (FOs) [60], which define a framework for describing and managing flexibility from heterogeneous resources. The work presented in [85] describes the interaction of different market players using FOs, and in [90] methods for aggregating and disaggregating flexibility while trading in an electricity market are proposed. Moreover, Lilliu et al.[54] expand the definition of FOs by incorporating additional constraints and bidirectional charging. While an individual EV's bidirectional flexibility presented in this work can be expressed as FOs, our aggregation schemes are incompatible with FOs as we are interested in preserving the exact flexibility limits through aggregation and disaggregation, while FOs deal with approximate methods.

2.1.3 Leveraging Flexibility of Electric Vehicles

An approach specific to utilizing the flexibility in EV charging is presented in [26]. The same authors implement scheduling algorithms in their follow-up work [75], where unidirectional flexibility is defined in terms of connection time and energy delivered. This flexibility can be managed to flatten the load on the grid or to maximize the utilization of renewables. Furthermore, Schlund et al. [78] propose a methodology to aggregate EV flexibility with unidirectional charging to guarantee the availability of bidirectional flexibility over a fixed time horizon. Vandael et al.[92] propose trading EV flexibility over a finite horizon using dynamic programming. Danner et al. [23] solve the optimal scheduling of a stationary battery and EVs (without V2G) under various forecast conditions. The flexibility requests are then accommodated

by adjusting charging schedules using an iterative algorithm.

2.1.4 Aggregation and Disaggregation of Flexibility

Aggregation A VPP serves as a *de facto* aggregator from the perspective of a grid operator. However, the manner in which this is handled internally varies. Some VPPs consider individual DER requirements to calculate the power flow in each of them, as seen in [14], [25], [93], among others. Conversely, others create an aggregate abstraction and either base their decisions on it or expose it to the grid operator. Various techniques exist for aggregation. For instance, in the case of FlexOffers, multiple offers from individual devices can be merged into a single aggregate FlexOffer, approximating the total available flexibility [86]. Moreover, Martin et al. propose a software system in [58] capable of controlling a network of batteries as a single unit, managing capacity, charge, and discharge collectively. In [65], Pertl et al. aggregate EV batteries into a virtual storage model, predicting the evolution of its characteristics through time using autoregressive methods, with the purpose of trading in the day-ahead market. Our approach shares similarities with this, aggregating multiple car batteries connected to chargers into a single virtual battery. However, a key distinction is our emphasis on modeling and enforcing charging deadlines, ensuring feasibility in scheduling.

Disaggregation Disaggregation refers to the process of taking a decision made for the regarding the total flexibility that is available, and turning it into control signals sent to individual DERs. We are concerned with designing a disaggregation scheme that considers *fairness*. We base our approach on seminal work in resource allocation [48], [101], which introduces the notion of *proportional fairness* and elucidate its properties for elastic traffic in computer networks. We compare our *fair* disaggregation approach with a priority-based allocation that was previously adopted in [91]. Other disaggregation approaches exist in the literature, such as [22], where Danner et al. design a genetic algorithm that considers a *fairness index*, among other criteria. However, running a genetic algorithm at each timestep might be prohibitively computationally costly. In contrast, in our work we perform disaggregation

through a single call to a convex optimization solver.

2.2 Reinforcement Learning for EV Charging

A large number of studies use RL to find an optimal charging schedule for EVs. Some focus on a decision-making problem involving a *single* EV [38], [52], [84]. However, we focus our literature review on the papers that control the charging of multiple EVs as it aligns with the objective of this thesis. We categorize the approaches in the literature into three categories: multi-agent, single agent taking individual actions, and single agent taking aggregate actions.

Multi-Agent Reinforcement Learning In multi-agent reinforcement learning (MARL), each EV charger is a decision-making agent. For example, Zishan et al. propose an adaptive distributed control algorithm aimed at preventing grid congestion in [5]. They tune each EV's controller parameters using a MARL framework. In [83], Shi et al. design a multi-agent system for managing a fleet of ride-sharing EVs, where each EV can decide to idle, charge or accept a trip request from a user. Similarly, Wang et al. [97] address charging and routing problems jointly by formulating a Partially Observable Markov Game for discrete and continuous actions. Their focus is on enhancing resilience in electrical grids after extreme events. Furthermore, Yan et al. [102] propose a multi-agent strategy that controls the charge for each vehicle in order to minimize the cost for each individual and prevent transformers from overloading.

The MARL framework is widely acknowledged by recent reviews [1], [67] as the most popular for managing the charge of multiple EVs that do not have the same information about the environment, including the distribution grid. There are also opportunities to integrate the MARL framework with distributed algorithms. However, there remain disparities in the application of this framework. For instance, valid MARL approaches include centralized training with centralized execution (CTCE), centralized training with distributed execution (CTDE), and distributed training with distributed

execution (DTDE) [67]. Each approach has its trade-offs. In CTCE, decision-making is centralized, facilitating the modeling of interactions between agents, but imposing significant computational load on a single player. Conversely, DTDE faces challenges in learning optimal policies and enforcing coupling constraints since agents only interact with each other through the environment [67]. Additionally, for all of the MARL approaches, the computational complexity can increase as more EVs join the systems and more agents need to be executed.

Single Agent Taking Individual Actions A second approach involves a single RL agent, typically with a Neural Network (NN) policy, directly managing a group of EVs by assigning the charge rate for each EV through its output nodes. Rahman et al. design a SAC agent capable of V2G that controls a fixed number of EVs based on imbalance market prices and day-ahead commitments in [68]. Similarly, Karatzinis et al. [47] train PPO and DDPG agents to control 10 electric vehicles, aiming to minimize electricity costs while penalizing missed charging deadlines. Cording et al. [20] propose a PPO agent to charge five EVs while minimizing costs and avoiding grid overloads. This is a simple approach to controlling a fleet of EVs, as it does not require a multi-agent system or separate aggregation and disaggregation algorithms. Additionally, the agent has full knowledge of each of the EVs' individual features. However, it relies on knowing the maximum number of concurrent arrivals in advance to set the number of output nodes in the NN. Moreover, uneven use of charging stations during training can skew the agent's learning, and there's no assurance of consistent treatment for vehicles with similar features. Consequently, coupling constraints are overlooked, such as *fairness*. Moreover, any changes in the total number of charging spots require architecture adjustments, leading to a need to re-train the agent.

Single Agent Taking Aggregate Action In this third approach, a single agent operates on an aggregate level, making decisions based on the aggregate flexibility provided by all connected EVs. Individual features are combined

using a separate aggregation algorithm, and aggregate actions are then broken down into individual decisions using a disaggregation algorithm. Vandael et al. [91] use a single RL agent to lay out a day-ahead consumption plan, then use priority-based dispatch to disaggregate the action. However, the disaggregated action is not guaranteed to be feasible, so it is clipped at the individual level. Li et al. [53] develop an aggregator-operator scheme that combines RL with predictive control for discrete binary charging actions by calculating multiple trajectories. Sadeghianpour et al. [74] propose a streamlined binning approach for unidirectional charging only. Another binning approach is taken by Alshehhi et al. [6] to create a scalable representation of EV fleets in order to train a deep neural network to solve a combinatorial optimization problem. Our approach handles charging and V2G contract constraints by aggregating them from an individual level and learning an aggregate action that is guaranteed to be feasible at the individual level.

This category offers improved scalability compared to "Single Agent Taking Individual Actions", and eliminates the need for information sharing or collaboration among multiple agents. It also allows for the enforcement of coupling constraints (e.g. *fairness*) during the disaggregation stage. However, it requires the design of an additional aggregation-disaggregation scheme, potentially leading to the loss of some nuances present in the individual data once aggregated.

We argue that our "Single Agent Taking Aggregate Action" approach combines the strengths of the two other approaches, namely the ability to scale like MARL and the simplicity of training and deploying a single agent like "Single Agent Taking Individual Action". At the same time, it offers some additional benefits, such as being able to incorporate coupling constraints during disaggregation. A summary of the strengths and weaknesses of each approach is provided in Table 2.1.

2.2.1 Incorporating Constraints in RL

Learning policies that are guaranteed to produce actions within a safe or feasible set is an active area of research. In 2019, OpenAI published a report

Multiple Agents	Single Agent Taking Individual Actions	Single Agent Taking Aggregate Action
✓ Direct to set up for multiple EVs.	✓ Agent has knowledge of every EV's features.	✓ Can handle arbitrary amount of connected EVs.
✓ Potential for distributed decision making.	✓ No need to design (dis)aggregation methods.	✓ Disaggregation can be designed for properties like *fairness*.
× Difficult to incorporate coupling constraints.	× Number of connected EVs is limited by NN architecture.	× Need to design separate (dis)aggregation methods.
× Increased complexity of running multiple agents.	× Behaviour might change for EVs in different output nodes.	× Some information may be lost during aggregation.

Table 2.1: Comparison of different approaches.

benchmarking various RL methods for safety during exploration, i.e. while training a policy [71]. They considered Lagrangian penalized [11] versions of policy gradient methods, specifically Trust Region Policy Optimization [80] and Proximal Policy Optimization [81], and Constrained Policy Optimization (CPO) [3] – which sets to analytically include constraints at each policy update. However, none of these methods satisfy our requirements as the Lagrangian approaches merely penalize the reward function without guarantees, and large penalty factors may sacrifice optimality. Moreover, the CPO performs poorly in the benchmarks [71] due to approximation errors.

We seek methods that ensure the satisfaction of hard constraints during training and after deployment. To achieve this, we review architectures that constrain the output of the neural network directly. Chen et al. [18] proposed embedding a differentiable convex optimization layer [4] into the policy network (as the last layer) to ensure that the action complies with the constraints that form a convex set. Another approach uses the activation function in the last layer to constrain the action to a predefined range, then the action is projected to a sample-specific feasible set using a gauge function [15]. In our work, we also use an activation function to limit the range of the output. However, we use linear interpolation to deduce the action value between aggregate bounds. We prefer this approach because it has lower computational overhead than differentiable optimization layers and works well when there are

only simple range constraints for the action. Despite its simplicity, we warn that linear interpolation of the action may be problematic when the upper and lower bounds of the action change over time. This is because you may never see a certain range for the action during training. This is one drawback of this approach.

2.3 Monetary Incentives for V2G

There are various approaches for incentivizing private EV owners to participate in V2G. For instance, a mechanism is designed in [99] where an aggregator dynamically changes the price to incentivize EVs to provide frequency regulation service. In [103], a two-level reverse auction is used to achieve demand response management in V2G systems.

We incentivize V2G using *contract theory* [76], which is concerned with designing a set of contracts between two self-interested parties. These contracts represent a commitment where one party agrees to pay the other upon successful fulfillment of the terms of the contract. In our setting, a contract signifies the VPP's promise to pay a certain amount to EV owners for discharging a specific amount of energy from their battery for an agreed period of time. The study conducted in [41] surveys EV owners about their preferences regarding their participation in V2G contracts. It inquires about parameters such as monthly payoff, required connected time, guaranteed minimum battery level and number of discharge cycles. Jember et al. [45] propose a two-tiered approach where the aggregator finds the optimal energy price through a game theoretic approach, and subsequently provides V2G contracts to motivate EV owners to participate. But the aggregator's interactions with the electricity markets are not studied in that work. To design V2G contracts without any prior knowledge of the EV owners' willingness to participate, Gao et al. [33] develop an algorithm that learns the optimal unit price based on its ongoing interactions with EVs. We base our formulation of contracts on [70], where the authors design fixed-term contracts that specify the maximum discharge energy that is allowed and the corresponding payoff. In Chapter 3, we extend

this formulation to design variable-term contracts. This requires considering two dimensions for the EV owner type. As a result, more of the owners' preferences are captured in the resulting contracts. Multidimensional contracts have been studied in diverse areas, such as mobile networks [100], federated learning [55], and radio communications [16].

2.4 Summary

In this chapter, we explored the relevant literature to highlight the novelty of our work, and compared our approach with the approaches taken in the literature to address a similar problem. In the next chapters, we will present our methodology, providing detailed explanations, and present empirical validation of our ideas.

Chapter 3

Contract Design

In this chapter, we focus on designing variable-term contracts aimed at incentivizing private EV owners to let the VPP control their charging to take advantage of their bidirectional flexibility. We adopt the principal-agent model from contract theory [13] to design V2G contracts. According to this model, the VPP (principal) offers a set of V2G contracts to EV owners (agents) who decide to accept one of them or opt out based on their private information that determines their *type*. This relationship is shown in Figure 3.1.

We start by delineating the two-dimensional types employed to categorize EV owners. Then, we will define the contracts as a 3-tuple found by solving an optimization problem. This optimization involves defining utility functions for the agents, parameterized by their type, and another one for the VPP. Subsequently, we present the optimization problem to maximize the expected utility of the VPP, noting that the number of constraints can grow rapidly. We then look at how to simplify the constraints without altering the solution of the optimization problem, justifying this process with a proof. We will empirically compare the performance of the simplified version against the original version of the optimization problem. Finally, we will solve the optimization problem to derive the contract values, which will be visualized and analyzed to ensure they comply with their desired properties.

Two-dimensional Types We assume that the agent type is two-dimensional. The first dimension, called *energy type*, indicates how much they are willing to discharge their vehicle battery. This mainly depends on how they perceive the

battery degradation cost which could be influenced by various factors, from the characteristics of their vehicle battery and the difficulty of replacing it to the climate in which they live. We assume the energy type can take a finite number of values that belong to $\Theta^w = \{\theta_1^w, \ldots, \theta_I^w\}$. Note that the types are listed in ascending order, e.g. $\theta_1^w < \theta_2^w$. The second dimension, called *persistence type*, indicates how long they are willing to allow their vehicle battery to be discharged. This depends on how they perceive the cost of staying longer or idling at the charging station. We assume the persistence type can take a finite number of values that belong to $\Theta^\ell = \{\theta_1^\ell, \ldots, \theta_J^\ell\}$, also in ascending order. The VPP operator does not know the type of a specific EV since it depends on their private information. This condition is known as *information asymmetry*. However, it knows the probability distribution over the two-dimensional types, i.e. it knows that an arbitrary EV may be of type $(\theta_i^w, \theta_j^\ell)$ with probability $\rho_{i,j}$; $(\sum_{i,j} \rho_{i,j}{=}1)$.

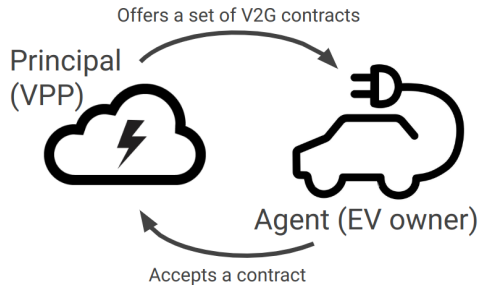

Figure 3.1: Principal-Agent Model

Contract Definition A variable-term V2G contract is characterized by a 3-tuple $(g_{i,j}, w_i, \ell_j)$, indicating respectively the payoff to agent (in €), the maximum amount of energy that can be discharged from their vehicle battery (in kWh), and the contract duration (in hours). If this contract is accepted, the VPP can withdraw up to w_i kWh from the vehicle battery during the first ℓ_j hours after accepting the contract. We seek contracts that possess two properties: *individual rationality* (IR) and *incentive compatibility* (IC). Individual rationality means that an EV owner will only accept a contract if it provides non-negative utility, that is, the payoff $g_{i,j}$ is enough to outweigh

the perceived battery degradation and idling costs. Incentive compatibility guarantees that an EV gains the highest utility by choosing the contract that was specifically designed for its type. This way the contracts will be *self-revealing* [76], addressing the information asymmetry between the principal and agents.

Agent's Utility The utility of an EV owner that accepts V2G contract $(g_{i,j}, w_i, \ell_j)$ is defined as:

$$U_{EV} = g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \tag{3.1}$$

where $g_{i,j}$ is the payoff for accepting the contract and the next two terms are the cost incurred by discharging w_i from the vehicle battery over a set duration of ℓ_j. The coefficient c_1 represents the *actual* battery degradation cost measured in €/kWh. This is multiplied by the amount of discharged energy w_i and gets divided by θ_i^w, implying that this cost will be higher for lower energy types. Similarly, c_2 represents the cost of having the vehicle battery available for discharge for the first ℓ_j hours of the charging session, measured in €/hr. This is multiplied by the contract duration ℓ_j and gets divided by θ_j^ℓ, implying that this cost will be higher for lower persistence types. As θ_i^w and θ_j^ℓ appear in the denominator, higher values indicate that the EV owner is more willing to participate.

Principal's Utility The utility function for the VPP is defined as:

$$U_{VPP} = \sum_{i=1}^{I} \sum_{j=1}^{J} \rho_{i,j}(\kappa_1 \log(w_i + 1) + \kappa_2 \log(\ell_j + 1) - g_{i,j}). \tag{3.2}$$

This is the VPP's expected utility, as it shows the sum of utilities over the EV types multiplied by their probability. For each EV type, the VPP appraises the amount of energy withdrawn from the battery (w_i) as well as the time window during which it can be used (ℓ_j). As the VPP is *risk-averse*, these two values are inside the concave log function. The relative importance of these two terms can be adjusted by tuning hyper-parameters κ_1 and κ_2. Finally, the

payoff $g_{i,j}$ for each type must be subtracted as this corresponds to the money that will be transferred to the EV owner.

Optimal Contract Mechanism The variable-term V2G contracts that will be offered by the VPP are the solution to an optimization problem that maximizes its expected utility. We overload the notation for sets Θ^w and Θ^ℓ to also denote the corresponding index sets, so we can write $i \in \Theta^w$, and $j \in \Theta^\ell$.

$$\underset{\substack{\{(g_{i,j},w_i,\ell_j)\}\\ i=1,\dots,I;\, j=1,\dots,J}}{\text{maximize}} \quad U_{VPP} \tag{3.3}$$

subject to:

(**IR**) $$g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq 0; \quad \forall (i,j) \in \Theta^w \times \Theta^\ell$$

(**IC**) $$g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq g_{i',j'} - \frac{c_1 \cdot w_{i'}}{\theta_i^w} - \frac{c_2 \cdot \ell_{j'}}{\theta_j^\ell};$$
$$\forall i, i' \in \Theta^w,\ \forall j, j' \in \Theta^\ell;\ i \neq i' \vee j \neq j'$$

(**PC**) $$w_I \leq \alpha_d \cdot \ell_J$$

(**MO**)

$(w_{(\cdot)})$ $$0 \leq w_1 \leq w_2 \leq \cdots \leq w_I$$

$(\ell_{(\cdot)})$ $$0 \leq \ell_1 \leq \ell_2 \leq \cdots \leq \ell_J$$

$(g_{(i,\cdot)})$ $$0 \leq g_{i,1} \leq g_{i,2} \leq \cdots \leq g_{i,J} \quad \forall i \in \Theta^w$$

$(g_{(\cdot,j)})$ $$0 \leq g_{1,j} \leq g_{2,j} \leq \cdots \leq g_{I,j} \quad \forall j \in \Theta^\ell$$

$(g_{(\cdot,\cdot)})$ $$g_{i,j} \leq g_{i',j'};\ \forall (i,j),(i',j') \in \Theta^w \times \Theta^\ell;\ i \leq i' \wedge j \leq j'$$

The first constraint in Problem (3.3), **(IR)**, ensures that for every type $(i,j) \in \Theta^w \times \Theta^\ell$, the contract offers non-negative utility. The second constraint, **(IC)**, ensures that an EV owner of type $(\theta_i^w, \theta_j^\ell)$ gets lower utility from accepting contract $(g_{i',j'}, w_{i'}, \ell_{j'})$ than contract $(g_{i,j}, w_i, \ell_j)$ where $i \neq i'$ or

$j \neq j'$. The third constraint, **(PC)**, reflects the physical constraint of bidirectional chargers. It ensures that the maximum discharge energy w_I is less than or equal to the charger's rated discharge power α_d (measured in kWh) multiplied by the maximum contract duration ℓ_J. Thus, the VPP can effectively discharge w_I while the contract remains active. The last group of constraints, **(MO)**, ensures the monotonicity of each contract parameter.

3.1 Simplifying V2G Contracts

The number of constraints in Problem (3.3) is of $O((I \times J)^2)$. This is particularly because of constraint **(IC)**, which compares every type against every other type. By extending the proofs in a manner similar to [70] with methods used in multidimensional auctions [56], this problem can be simplified to have $O(I \times J)$ constraints. Both formulations will still have the same number of decision variables, namely $I + J + I \times J$. However, the smaller number of constraints will result in faster solve times.

3.1.1 Preliminaries

For this simplification, let us first define a shorthand to express the EV's utility in Eqn. (3.1), but separating the contract parameters from the owner's true type.

$$V_{EV}(\underbrace{\theta^w_{i'}, \theta^\ell_{j'};}_{Contract\ params.} \underbrace{\theta^w_i, \theta^\ell_j}_{True\ types}) = g_{i',j'} - \frac{c_1 \cdot w_{i'}}{\theta^w_i} - \frac{c_2 \cdot \ell_{j'}}{\theta^\ell_j} \tag{3.4}$$

Note how the contract parameters $(g_{i',j'}, w_{i'}, \ell_{j'})$ have different indices than the true types $(\theta^w_i, \theta^\ell_j)$ on the right hand side of Eqn. (3.4) Now, we can express *individual rationality* **(IR)** as:

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq 0 \quad \forall i \times j \in \Theta^w \times \Theta^\ell$$

Similarly, *incentive compatibility* **(IC)** can be expressed as:

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_{i'}, \theta^\ell_{j'}; \theta^w_i, \theta^\ell_j);\ \forall \theta^w_i, \theta^w_{i'} \in \Theta^w, \forall \theta^\ell_j, \theta^\ell_{j'} \in \Theta^\ell$$

As we know, the **(IC)** constraints relate every type to every other type. We categorize these constraints into the following groups in order to talk precisely about these relationships.

- **Vertical IC Constraints:** For $i \neq i' \wedge j = j'$
- **Horizontal IC Constraints:** For $i = i' \wedge j \neq j'$
- **Other IC Constraints:** For $i \neq i' \wedge j \neq j'$
- **Downward IC Constraints (DIC):** For $i > i' \vee j > j'$
- **Upward IC Constraints (UIC):** For $i < i' \vee j < j'$
- **Local IC Constraints:** When $|i - i'| = 1 \vee |j - j'| = 1$

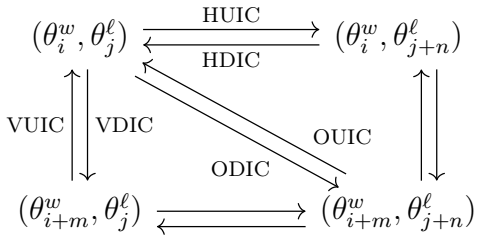

Figure 3.2: Relationships between **IC** constraints

Furthermore, these categories of **IC** constraints can be combined in such a way that a *Vertical Downward* **IC** is $V_{EV}(\theta_i^w, \theta_j^\ell; \theta_i^w, \theta_j^\ell) \geq V_{EV}(\boxed{\theta_{i'}^w}, \theta_j^\ell; \theta_i^w, \theta_j^\ell)$ where $i > i'$. Another example is a *Horizontal Local Upward* **IC**: $V_{EV}(\theta_i^w, \theta_j^\ell; \theta_i^w, \theta_j^\ell) \geq V_{EV}(\theta_i^w, \boxed{\theta_{j+1}^\ell}; \theta_i^w, \theta_j^\ell)$. This nomenclature will be used throughout the remainder of this chapter. Additionally, as a shorthand of two-dimensional types we define $\theta_{i,j}^{w,\ell} = (\theta_i^w, \theta_j^\ell)$ and $\Theta^{w,\ell} = \Theta^w \times \Theta^\ell$, such that $\theta_{i,j}^{w,\ell} \in \Theta^{w,\ell}$.

3.1.2 Simplified Problem

The simplified problem is presented here as Problem (3.5). The first constraint **(C1)** sets the utility to zero for the first type of EV owner $(\theta_1^w, \theta_1^\ell)$. Then, constraint **(C2)** only deals with *Vertical Local Downward IC's*, and **(C3)** deals with *Horizontal Local Downward IC's*. Note that these constraints are taken to be binding, that is, the "less or equals" sign is substituted by an "equals" sign. Notice that all of the monotonicity constraints are omitted. Lastly, the physical constraint **(PC)** remains unchanged.

$$\underset{\substack{\{(g_{i,j},w_i,\ell_j)\}\\ i=1,\dots,I;\ j=1,\dots,J}}{\text{maximize}} \quad U_{VPP} \tag{3.5}$$

subject to:

(**C1**) $$g_{1,1} - \frac{c_1 \cdot w_1}{\theta_1^w} - \frac{c_2 \cdot \ell_1}{\theta_1^\ell} = 0$$

(**C2**) $$g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} = g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell};$$
$$\forall i \in \Theta^w \setminus \{1\},\ \forall j \in \Theta^\ell$$

(**C3**) $$g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} = g_{i,j-1} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_{j-1}}{\theta_j^\ell};$$
$$\forall i \in \Theta^w,\ \forall j \in \Theta^\ell \setminus \{1\}$$

(**PC**) $$w_I \leq \alpha_d \cdot \ell_J$$

3.1.3 Proof of Equivalence

This proof is structured as a series of lemmas that simplify certain aspects of the constraints. After all the lemmas are presented, they will be strung together to form the final proof that Problems (3.3) and (3.5) are equivalent.

Lemma 3.1.1. *At the solution, the **IR** constraint is active for the lowest type, $(\theta_1^w, \theta_1^\ell)$. Further, the **IR** constraints for higher types can be derived from the **IC** constraints.*

Proof. Take $\{(g_{i,j}, w_i, \ell_j)\}_{i=1,\dots,I;\ j=1,\dots,J}$ to be a feasible point for the optimization problem, and $\{(g^*_{i,j}, w^*_i, \ell^*_j)\}_{i=1,\dots,I;\ j=1,\dots,J}$ to be its solution. We know that every feasible point satisfies the **IC** constraints, therefore:

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_i, \theta^\ell_j) \tag{3.6}$$

Now, if contract $\{(g_{1,1}, w_1, \ell_1)\}$ is offered to both type $(\theta^w_1, \theta^\ell_1)$ and type $(\theta^w_i, \theta^\ell_j)$, with $i, j > 1$. The owner of latter type will gain higher utility since $(\theta^w_1, \theta^\ell_1) \preceq (\theta^w_i, \theta^\ell_j)$, because these variables appear in the denominator, and we have $c_1, w_1, c_2, \ell_1 \geq 0$. Therefore:

$$\begin{aligned} g_{1,1} - \frac{c_1 \cdot w_1}{\theta^w_i} - \frac{c_2 \cdot \ell_1}{\theta^\ell_j} &\geq g_{1,1} - \frac{c_1 \cdot w_1}{\theta^w_1} - \frac{c_2 \cdot \ell_1}{\theta^\ell_1} \\ \implies V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_i, \theta^\ell_j) &\geq V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_1, \theta^\ell_1) \end{aligned}$$

Combining this with (3.6) and the **IR** constraint for type $(\theta^w_1, \theta^\ell_1)$ we have that:

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_1, \theta^\ell_1) \geq 0$$

Thus, given the **IC** constraints, the **IR** constraints for higher types are naturally satisfied for every feasible point in Problem (3.3).

Next, we establish by contradiction that the **IR** constraint for the lowest type, $(\theta^w_1, \theta^\ell_1)$, is binding at the solution. First, suppose that the **IR** constraint is *not* binding, then it would stand that:

$$\begin{aligned} V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_1, \theta^\ell_1) &> 0 \\ \implies g^*_{1,1} - \frac{c_1 \cdot w^*_1}{\theta^w_1} - \frac{c_2 \cdot \ell^*_1}{\theta^\ell_1} &> 0 \end{aligned}$$

Next, consider a new contract $(g'_{1,1}, w^*_i, \ell^*_j)$, such that $g'_{1,1} = g^*_{1,1} - \epsilon$ where $0 < \epsilon \leq g^*_{1,1} - (c_1 \cdot w^*_1)/\theta^w_1 - (c_2 \cdot \ell^*_1)/\theta^\ell_1$. Then the utility for that user would be:

$$\begin{aligned} g'_{1,1} - \frac{c_1 \cdot w^*_1}{\theta^w_1} - \frac{c_2 \cdot \ell^*_1}{\theta^\ell_1} &= (g^*_{1,1} - \epsilon) - \frac{c_1 \cdot w^*_1}{\theta^w_1} - \frac{c_2 \cdot \ell^*_1}{\theta^\ell_1} \\ \implies V_{EV}(\theta^w_1, \theta^\ell_1; \theta^w_1, \theta^\ell_1) - \epsilon &> 0 \end{aligned}$$

Since the new contract, $\{(g'_{1,1}, w^*_1, \ell^*_1)\}$, satisfies the **IR** as well as the rest of constraints, and it increases the expected utility of the VPP as $g'_{1,1} < g^*_{1,1}$,

then it follows that $\{(g^*_{1,1}, w^*_1, \ell^*_1)\}_{i=1,\dots,I;\, j=1,\dots,J}$ cannot be the solution. So, by contradiction, it is proved that the **IR** constraint for type $(\theta^w_1, \theta^\ell_1)$ is binding. □

Lemma 3.1.2. *All "other" monotonicity constraints in $(g_{(\cdot,\cdot)})$ can be derived from vertical $(g_{(\cdot,j)})$ and horizontal $(g_{(i,\cdot)})$ monotonicity constraints for g.*

Proof. Take an instance of $g_{i,j}$ and $g_{i+m,j+n}$ where $i \leq i+m \wedge j \leq j+n$. We can simply follow the *vertical* monotonicity constraints in order to find that $g_{i,j} \leq g_{i+m,j}$. From there, we follow the *horizontal* monotonicity constraints to find that $g_{i+m,j} \leq g_{i+m,j+n}$. From transitivity, it follows that $g_{i,j} \leq g_{i+m,j+n}$ without explicitly employing the *"other"* monotonicty constraints. □

Lemma 3.1.3. *Given the vertical and horizontal **IC** constraints, monotonicity in $g_{(\cdot,j)}$ follows from monotonicity in $w_{(\cdot)}$, and monotonicity in $g_{(i,\cdot)}$ follows from monotonicity in $\ell_{(\cdot)}$. Therefore, we only need to keep monotonicity in $w_{(\cdot)}$ and $\ell_{(\cdot)}$.*

Proof. We begin by showing that if $w_i \geq w_{i'}$ then $g_{i,j} \geq g_{i',j'}$. For this, we take the *vertical* **IC** constraint for some type $(\theta^w_i, \theta^\ell_j) \in \Theta^w \times \Theta^\ell$.

$$
\begin{aligned}
& V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_{i'}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq \boxed{g_{i',j}} - \frac{c_1 \cdot \boxed{w_{i'}}}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} \geq g_{i',j} - \frac{c_1 \cdot w_{i'}}{\theta^w_i} \\
\implies & g_{i,j} - g_{i',j} \geq \frac{c_1 \cdot (w_i - w_{i'})}{\theta^w_i}
\end{aligned}
$$

From this inequality we can observe that if $w_i \geq w_{i'}$ then the right side of the inequality must be positive. Hence, the left side must be positive too, so $g_{i,j} \geq g_{i',j}$.

The second part of this proof is to show that if $\ell_j \geq \ell_{j'}$ then $g_{i,j} \geq g_{i,j'}$. In

this case, we take the *horizontal* **IC** constraint:

$$
\begin{aligned}
& V_{EV}(\theta_i^w, \theta_j^\ell; \theta_i^w, \theta_j^\ell) \geq V_{EV}(\theta_i^w, \boxed{\theta_{j'}^\ell}; \theta_i^w, \theta_j^\ell) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq \boxed{g_{i,j'}} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \boxed{\ell_{j'}}}{\theta_j^\ell} \\
\implies & g_{i,j} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq g_{i,j'} - \frac{c_2 \cdot \ell_j'}{\theta_j^\ell} \\
\implies & g_{i,j} - g_{i,j'} \geq \frac{c_2 \cdot (\ell_j - \ell_{j'})}{\theta_j^\ell}
\end{aligned}
$$

As before, we can see from this inequality that if $\ell_j \geq \ell_{j'}$ then the right side of the inequality is positive. Then, the left side is also positive, such that $g_{i,j} \geq g_{i,j'}$. □

Lemma 3.1.4. *Given vertical IC constraints, we can derive monotonicity in* $(w_{(\cdot)})$ *from monotonicity in* θ^w*. Additionally, given horizontal IC constraints, we can derive monotonicity in* $(\ell_{(\cdot)})$ *from monotonicity in* θ^ℓ*.*

Proof. Let us start by deriving monotonicity in w from monotonicity in θ^w. For this, we consider two types $(\theta_i^w, \theta_j^w), (\theta_{i+m}^w, \theta_j^\ell) \in \Theta^w \times \Theta^\ell$, such that $\theta_i^w < \theta_{i+m}^w$. We write and reorganize the *vertical* IC constraint for type $(\theta_{i+m}^w, \theta_j^\ell)$ with respect to type (θ_i^w, θ_j^w), like so:

$$
\begin{aligned}
& V_{EV}(\theta_{i+m}^w, \theta_j^\ell;\ \theta_{i+m}^w, \theta_j^\ell) \geq V_{EV}(\boxed{\theta_i^w}, \theta_j^\ell;\ \theta_{i+m}^w, \theta_j^\ell) \\
\implies & g_{i+m,j} - \frac{c_1 \cdot w_{i+m}}{\theta_{i+m}^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq \boxed{g_{i,j}} - \frac{c_1 \cdot \boxed{w_i}}{\theta_{i+m}^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \\
\implies & g_{i+m,j} - g_{i,j} \geq \frac{c_1 \cdot w_{i+m}}{\theta_{i+m}^w} - \frac{c_1 \cdot w_i}{\theta_{i+m}^w}
\end{aligned} \tag{3.7}
$$

Subsequently, we write the *vertical* IC constraint for type (θ_i^w, θ_j^w) with respect to type $(\theta_{i+m}^w, \theta_j^\ell)$ and reorganize in the same manner.

$$
\begin{aligned}
& V_{EV}(\theta_i^w, \theta_j^\ell;\ \theta_i^w, \theta_j^\ell) \geq V_{EV}(\boxed{\theta_{i+m}^w}, \theta_j^\ell;\ \theta_i^w, \theta_j^\ell) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \geq \boxed{g_{i+m,j}} - \frac{c_1 \cdot \boxed{w_{i+m}}}{\theta_i^w} - \frac{c_2 \cdot \ell_j}{\theta_j^\ell} \\
\implies & g_{i,j} - g_{i+m,j} \geq \frac{c_1 \cdot w_i}{\theta_i^w} - \frac{c_1 \cdot w_{i+m}}{\theta_i^w}
\end{aligned} \tag{3.8}
$$

By adding Inequalities (3.7) and (3.8), we have that:

$$
\begin{aligned}
0 &\geq \frac{c_1 \cdot w_{i+m}}{\theta^w_{i+m}} - \frac{c_1 \cdot w_i}{\theta^w_{i+m}} + \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_1 \cdot w_{i+m}}{\theta^w_i} \\
\implies 0 &\geq \frac{w_{i+m}}{\theta^w_{i+m}} - \frac{w_i}{\theta^w_{i+m}} + \frac{w_i}{\theta^w_i} - \frac{w_{i+m}}{\theta^w_i}; \text{ Since } c_1 > 0 \\
\implies 0 &\geq \frac{1}{\theta^w_{i+m}} \cdot (w_{i+m} - w_i) - \frac{1}{\theta^w_i} \cdot (w_{i+m} - w_i) \\
\implies 0 &\geq \left(\frac{1}{\theta^w_{i+m}} - \frac{1}{\theta^w_i} \right) \cdot (w_{i+m} - w_i)
\end{aligned} \tag{3.9}
$$

Since we know that $\theta^w_{i+m} > \theta^w_i$ then $\left(\frac{1}{\theta^w_{i+m}} - \frac{1}{\theta^w_i} \right) < 0$. Hence, $(w_{i+m} - w_i) \geq 0$ in order for Inequality (3.9) to be satisfied. Thus, we have derived monotonicity in $(w_{(\cdot)})$ from monotonicity in θ^w. A similar process can be followed to derive monotonicity in $(\ell_{(\cdot)})$ from monotonicity in θ^ℓ by using horizontal IC constraints. For conciseness, we skip these derivations.

□

Lemma 3.1.5. *"Other" (non-vertical or non-horizontal)* ***IC*** *constraints, where* $\theta^w_i \neq \theta^w_i \wedge \theta^\ell_j \neq \theta^\ell_{j'}$*, can be derived from vertical and horizontal* ***IC****s, where* $\theta^w_i \neq \theta^w_{i'}$ XOR $\theta^\ell_j \neq \theta^\ell_{j'}$.

Proof. We start by considering three types: $(\theta^w_i, \theta^\ell_j), (\theta^w_{i'}, \theta^\ell_j), (\theta^w_{i'}, \theta^\ell_{j'}) \in \Theta^w \times \Theta^\ell$. Note that in this proof we do not make any assumption about the relationship between i and i', or j and j'. We will follow the vertical IC constraint $(\theta^w_i, \theta^\ell_j) \rightarrow (\theta^w_{i'}, \theta^\ell_j)$ and the horizontal IC constraint $(\theta^w_{i'}, \theta^\ell_j) \rightarrow (\theta^w_{i'}, \theta^\ell_{j'})$ to arrive to the *"other"* IC constraint $(\theta^w_i, \theta^\ell_j) \rightarrow (\theta^w_{i'}, \theta^\ell_{j'})$.

We start by writing the vertical IC constraint $(\theta^w_i, \theta^\ell_j) \rightarrow (\theta^w_{i'}, \theta^\ell_j)$, and reorganizing it:

$$
\begin{aligned}
& V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_{i'}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i',j} - \frac{c_1 \cdot w_{i'}}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & g_{i,j} - g_{i',j} \geq \frac{c_1}{\theta^w_i}(w_i - w_{i'})
\end{aligned} \tag{3.10}
$$

Then we write the horizontal IC constraint for $(\theta^w_{i'}, \theta^\ell_j) \rightarrow (\theta^w_{i'}, \theta^\ell_{j'})$, and reorganize:

$$
\begin{aligned}
& V_{EV}(\theta^w_{i'}, \theta^\ell_j; \theta^w_{i'}, \theta^\ell_j) \geq V_{EV}(\theta^w_{i'}, \boxed{\theta^\ell_{j'}}; \theta^w_{i'}, \theta^\ell_j) \\
\implies & g_{i',j} - \frac{c_1 \cdot w_{i'}}{\theta^w_{i'}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i',j'} - \frac{c_1 \cdot w_{i'}}{\theta^w_{i'}} - \frac{c_2 \cdot \ell_{j'}}{\theta^\ell_j} \\
\implies & g_{i',j} - g_{i',j'} \geq \frac{c_2}{\theta^\ell_j}(\ell_j - \ell_{j'}) \qquad (3.11)
\end{aligned}
$$

We then add Inequalities (3.10) and (3.11) and reorginize them in order to construct an *"other"* IC constraint, like so:

$$
\begin{aligned}
& g_{i,j} - g_{i',j'} \geq \frac{c_1}{\theta^w_i}(w_i - w_{i'}) + \frac{c_2}{\theta^\ell_j}(\ell_j - \ell_{j'}) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i',j'} - \frac{c_1 \cdot w_{i'}}{\theta^w_i} - \frac{c_2 \cdot \ell_{j'}}{\theta^\ell_j} \\
\implies & V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_{i'}, \theta^\ell_{j'}; \theta^w_i, \theta^\ell_j)
\end{aligned}
$$

Therefore, we have shown that all *"other"* **IC** constraints are redundant since we can derive them from vertical and horizontal **IC** constraints.

□

Lemma 3.1.6. *The vertical and horizontal **DICs** and **UICs** can be derived from Local DICs (**LDICs**) and Local UICs (**LUICs**) respectively.*

Proof. For the vertical case consider three types of owners:
$(\theta^w_{i-1}, \theta^\ell_j),\ (\theta^w_i, \theta^\ell_j),\ (\theta^w_{i+1}, \theta^\ell_j)$; with $\theta^w_{i-1} < \theta^w_i < \theta^w_{i+1}$.

Step 1. First consider the vertical LDIC of type $(\theta^w_i, \theta^\ell_j)$, expressed as:

$$
\begin{aligned}
& V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_{i-1}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} \geq g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_i} \\
\implies & g_{i,j} - g_{i-1,j} \geq \frac{c_1 \cdot (w_i - w_{i-1})}{\theta^w_i}
\end{aligned}
$$

Now, since $\theta^w_{i+1} > \theta^w_i$ we can write the following:

$$g_{i,j} - g_{i-1,j} \geq \frac{c_1 \cdot (w_i - w_{i-1})}{\boxed{\theta^w_{i+1}}}$$

This can be reorganized as:

$$\begin{aligned}
& g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i+1}} \geq g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_{i+1}} \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i+1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_{i+1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & V_{EV}(\theta^w_i, \theta^\ell_j, \theta^w_{i+1}, \theta^\ell_j) \geq V_{EV}(\theta^w_{i-1}, \theta^\ell_j, \theta^w_{i+1}, \theta^\ell_j)
\end{aligned}$$

Next, let us write the LDIC constraint for the EV owner type $(\theta^w_{i+1}, \theta^\ell_j)$:

$$V_{EV}(\theta^w_{i+1}, \theta^\ell_j; \theta^w_{i+1}, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_i}, \theta^\ell_j; \theta^w_{i+1}, \theta^\ell_j)$$

We can combine the two previous inequalities to get a *non-local* Downward Incentive Constraint, as such:

$$V_{EV}(\theta^w_{i+1}, \theta^\ell_j; \theta^w_{i+1}, \theta^\ell_j) \geq V_{EV}(\theta^w_{i-1}, \theta^\ell_j; \theta^w_{i+1}, \theta^\ell_j) \tag{3.12}$$

Therefore, we can employ the same approach for any type θ^w_i, with a constant θ^ℓ_j, to show that if its vertical Local DIC is satisfied, then the rest of vertical DICs will be satisfied as well.

Step 2. Next, we prove that vertical UICs can be derived from LUICs. To start, take the LUIC for an EV owner of type $(\theta^w_i, \theta^\ell_j)$:

$$\begin{aligned}
& V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_{i+1}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i+1,j} - \frac{c_1 \cdot w_{i+1}}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} \geq g_{i+1,j} - \frac{c_1 \cdot w_{i+1}}{\theta^w_i} \\
\implies & g_{i+1,j} - g_{i,j} \leq \frac{c_1 \cdot (w_{i+1} - w_i)}{\theta^w_i}
\end{aligned}$$

As $\theta_i > \theta_{i-1}$ and $w_{i+1} \geq w_i$, we can say that:

$$g_{i+1,j} - g_{i,j} \leq \frac{c_1 \cdot (w_{i+1} - w_i)}{\boxed{\theta^w_{i-1}}}$$

Then, we reorganize the inequality as such:

$$\begin{aligned} & g_{i+1,j} - \frac{c_1 \cdot w_{i+1}}{\theta^w_{i-1}} \leq g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i-1}} \\ \implies & g_{i+1,j} - \frac{c_1 \cdot w_{i+1}}{\theta^w_{i-1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \leq g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i-1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\ \implies & V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \geq V_{EV}(\theta^w_{i+1}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \end{aligned}$$

As before, let us write the LUIC constraint for type $(\theta^w_{i-1}, \theta^\ell_j)$

$$V_{EV}(\theta^w_{i-1}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \geq V_{EV}(\boxed{\theta^w_i}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j)$$

If we combine the two previous inequalities, we obtain a non-local Upward Incentive Constraint:

$$V_{EV}(\theta^w_{i-1}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \geq V_{EV}(\theta^w_{i+1}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \tag{3.13}$$

Parallel to the reasoning in the previous step, we can make use of the same approach for each type θ^w_i, with a constant θ^ℓ_j, to prove that if its vertical LUIC is satisfied then the rest of vertical UICs will be satisfied too.

Furthermore, we can repeat **Steps 1** and **2** for constant type θ^w_i and variable type θ^ℓ_j, e.g. $(\theta^w_i, \theta^\ell_{j-1}), (\theta^w_i, \theta^\ell_j), (\theta^w_i, \theta^\ell_{j+1})$; with $\theta^\ell_{j-1} < \theta^\ell_j < \theta^\ell_{j+1}$. Then we can prove that horizontal DICs and UICs can be derived from horizontal LDICs and LUICs, respectively. For conciseness, we skip that analysis as it is almost identical to that of vertical ICs. □

Lemma 3.1.7. *Vertical and horizontal LDICs are active (binding) at the solution.*

Proof. Firstly, we denote the solution as $\{(g^*_{i,j}, w^*_i, \ell^*_j)\}_{i=1,\dots,I;\ j=1,\dots,J}$, and then proceed to prove this by contradiction for the *vertical* LDICs. Suppose that for type $(\theta^w_i, \theta^\ell_j)$ the vertical LDIC is not active. This would be written as:

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \boxed{>} V_{EV}(\boxed{\theta^w_{i-1}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j)$$

$$\implies g^*_{i,j} - \frac{c_1 \cdot w^*_i}{\theta^w_i} - \frac{c_2 \cdot \ell^*_j}{\theta^\ell_j} > g^*_{i-1,j} - \frac{c_1 \cdot w^*_{i-1}}{\theta^w_i} - \frac{c_2 \cdot \ell^*_j}{\theta^\ell_j} \tag{3.14}$$

Then, we introduce $(g'_{i,j}, w^*_i, \ell^*_j)$ as a replacement for $(g^*_{i,j}, w^*_i, \ell^*_j)$, with $g'_{i,j} = g^*_{i,j} - \epsilon$ such that $\epsilon > 0$ and it is less than the difference between the two sides of (3.14). This new contract will increase the VPP's profit by lowering the payoff to the EV owner while still satisfying (3.14). This is a contradiction and thus $(g^*_{i,j}, w^*_i, \ell^*_j)$ cannot be the solution. Therefore, vertical LDICs must be active at the solution.

The same reasoning can be applied to the horizontal LDICs to prove how

$$V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \geq V_{EV}(\theta^w_i, \boxed{\theta^\ell_{j-1}}; \theta^w_i, \theta^\ell_j)$$

will be active at the solution.

□

Lemma 3.1.8. *LUICs can be relaxed if LDICs are active.*

Proof. First, we prove this for the vertical **IC**s. Take the binding vertical LDIC for type $(\theta^w_i, \theta^\ell_j)$:

$$\begin{aligned}
& V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_i, \theta^\ell_j) = V_{EV}(\boxed{\theta^w_{i-1}}, \theta^\ell_j; \theta^w_i, \theta^\ell_j) \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} = g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_i} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_i} = g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_i} \\
\implies & g_{i,j} - g_{i-1,j} = \frac{c_1 \cdot (w_i - w_{i-1})}{\theta^w_i}
\end{aligned}$$

Because we know that $\theta^w_{i-1} < \theta^w_i$, then we can substitute and rearrange such that:

$$\begin{aligned}
& \frac{c_1 \cdot (w_i - w_{i-j})}{\boxed{\theta^w_{i-1}}} \geq g_{i,j} - g_{i-1,j} \\
\implies & g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_{i-1}} \geq g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i-1}} \\
\implies & g_{i-1,j} - \frac{c_1 \cdot w_{i-1}}{\theta^w_{i-1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \geq g_{i,j} - \frac{c_1 \cdot w_i}{\theta^w_{i-1}} - \frac{c_2 \cdot \ell_j}{\theta^\ell_j} \\
\implies & V_{EV}(\theta^w_{i-1}, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j) \geq V_{EV}(\theta^w_i, \theta^\ell_j; \theta^w_{i-1}, \theta^\ell_j)
\end{aligned}$$

This is a vertical LUIC. Using a similar approach we can prove how vertical LUICs for different types can be derived from vertical LDICs when they are binding.

Moreover, we can follow the same steps to show how *horizontal* LUICs can be derived from *horizontal* LDICs. □

Theorem 3.1.9. *Problem* (3.3) *and Problem* (3.5) *are equivalent*

Proof. For this we make use of the previous lemmas and go through a series of steps as follows:

1. Use Lemma 3.1.1 to replace (**IR**) with (**C1**).
2. Then omit monotonicity for $(g_{(\cdot,\cdot)})$, as a result of Lemma 3.1.2.
3. With the help of Lemma 3.1.3, remove monotonicity for $(g_{(\cdot,j)})$ and $(g_{(i,\cdot)})$.
4. Lemma 3.1.4, leads to removing monotonicity for $(w_{(\cdot)})$ and $\ell_{(\cdot)}$.
5. Remove all *"other"* **IC**s and just keep the vertical and horizontal with Lemma 3.1.5.
6. All **UIC**s and **DIC**s can be reduced to **LUIC**s and **LDIC**s by using Lemma 3.1.6.
7. Change **LDICS** to an equality since they will be binding because of Lemma 3.1.7, these are separated into vertical (**C2**) and horizontal (**C3**).
8. Because of Lemma 3.1.8, remove **LUICs**.

Thus, we have proven that Problem (3.3) and Problem (3.5) are equivalent. □

Figure 3.3 shows the solver runtime for varying numbers of types, $I \times J$ of Problem (3.3) and (3.5), non-tractable and tractable formulations. Results are averaged over 5 runs and the error bars indicate the standard deviation. Both x and y axes follow a linear scale. Notice how the tractable formulation shows dramatically lower computation time as the number of types increases.

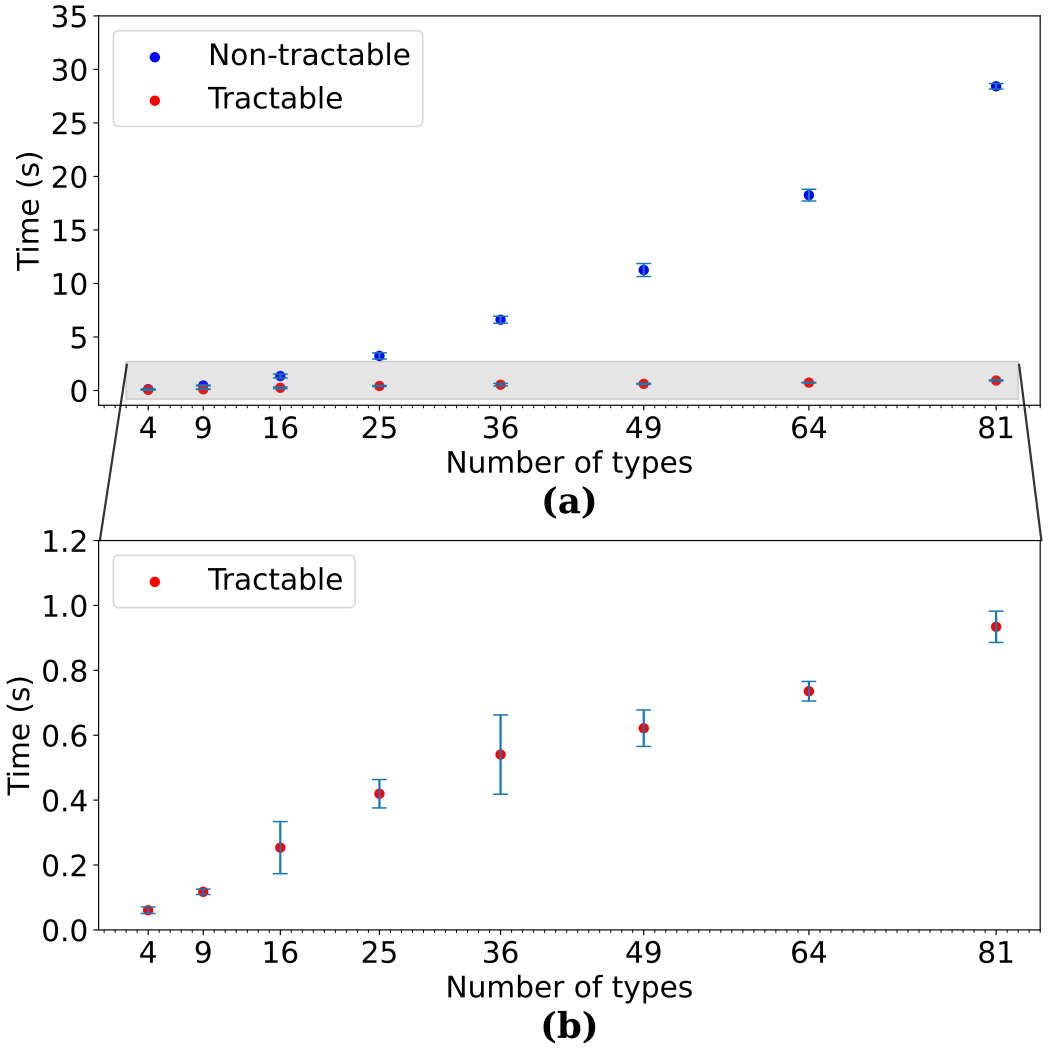

Figure 3.3: Solver runtime for a varying number of types. Plot **(a)** shows the comparison between the non-tractable and tractable formulations, while plot **(b)** shows the detail for the tractable case.

3.2 Optimal V2G Contracts

We solve the optimization problem with κ_1=0.4, κ_2=0.6, c_1=0.01 €/kWh, and c_2=0.05 €/hr. These parameters are chosen based on medium-term predictions of battery prices [35] and an analysis of rates in the electricity market similar to [70]. For the EV owner types, we are interested in three cases for each dimension, representing individuals that have low, average, and high value functions: Θ_i^w=Θ_j^ℓ=$\{0.75, 1, 1.25\}$. This results in 9 distinct types. We assume a uniform distribution of types and set $\rho_{i,j}$=1/9 for all i and j values.

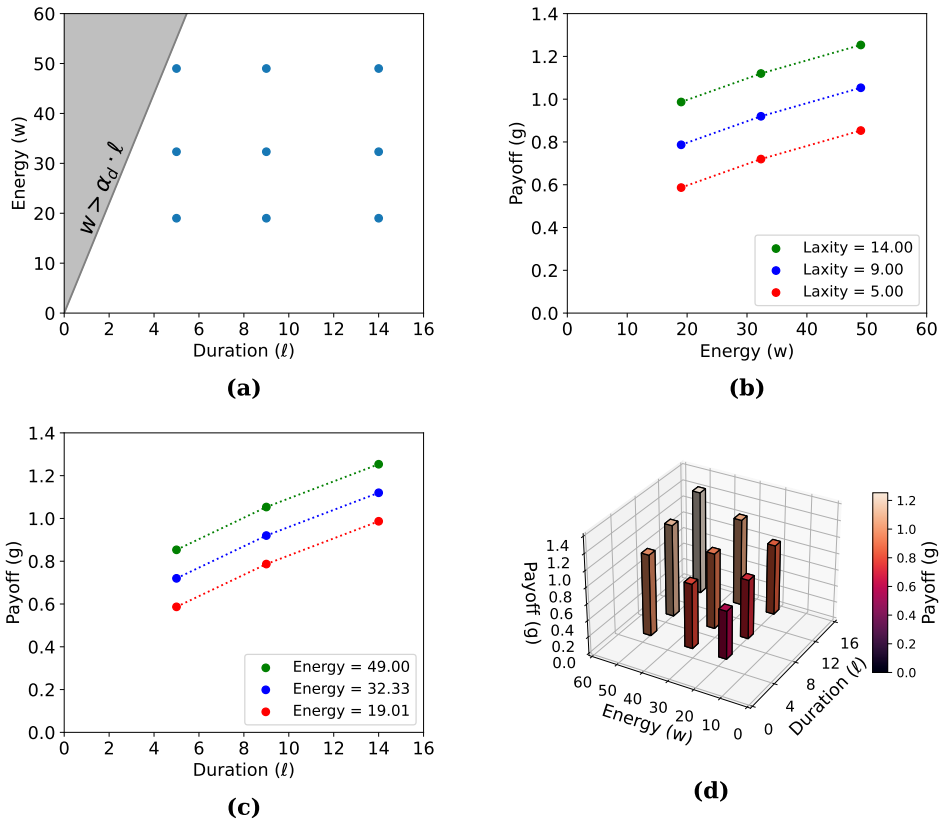

Figure 3.4: Visualizing the resulting contracts

We obtain the variable-term contracts that are shown in Tbl. 3.1.

	$\ell_1 = 5$	$\ell_2 = 9$	$\ell_3 = 14$
$w_1 = 19.01$	$g_{1,1} = 0.59$	$g_{1,2} = 0.79$	$g_{1,3} = 0.99$
$w_2 = 32.33$	$g_{2,1} = 0.72$	$g_{2,2} = 0.92$	$g_{2,3} = 1.12$
$w_3 = 49.00$	$g_{3,1} = 0.85$	$g_{3,2} = 1.05$	$g_{3,3} = 1.25$

Table 3.1: Pivot table of the contract values. Row headers indicate the amount of discharge energy in kWh, column headers indicate the contract duration in hours. The value in each cell is the payoff in €.

Figure 3.4 visualizes the values in Table 3.1. First, subplot **(a)** shows the contract duration vs. the discharge energy. All of the plotted contracts are below the shaded gray area because of the physical constraint **(PC)**. Also, note how they are not equally spaced. Since the VPP is risk-averse, and values energy and duration *logarithmically*, recall Eqn. (3.2), the gap between w_2 and w_3 is larger than the one between w_1 and w_2. A similar observation can be

made about the payoffs in subplots **(b)** and **(c)**. Looking closely at a group of three points on either of these subplots, we can see that the payoff does not increase linearly with the amount of resources offered by the EV. Instead, its slope decreases slightly. Again, this is because of the choice of the VPP's utility function. Lastly, subplot **(d)** shows a 3D barplot of the values of the contracts.

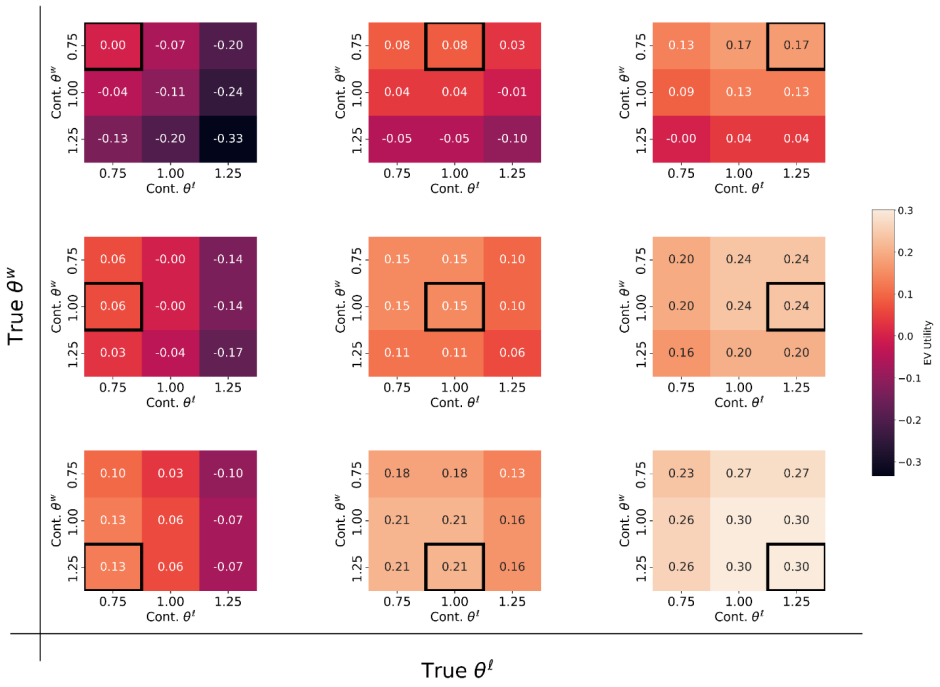

Figure 3.5: EV utility from the perspective of different owner types

From Figure 3.5 we can observe the desired properties of *individual rationality* and *incentive compatibility*. This figure shows 9 subplots. Each subplot is the point of view of a particular type of EV owner as indicated by the "True θ^w" and "True θ^ℓ" axis. Within each subplot, we observe the valuation of all the feasible contracts, as seen by that type of EV owner. The value is given by Eqn. (3.1). The contract designed for their specific type is enclosed by a black rectangle. The property of *individual rationality* is confirmed, as the value inside the black rectangle is non-negative in each of the subplots. Furthermore, these contracts show *incentive compatibility*. This is because the enclosed value is greater or equal to all other values inside that subplot.

3.3 Summary

In this chapter, we developed variable-term V2G contracts tailored to EV owners with two-dimensional types. Additionally, we introduced a simplified optimization problem that drastically reduces the computational time required for finding the solution. We then presented the V2G contracts that will be used in the rest of this work, confirming their adherence to the properties of *individual rationality* and *incentive compatibility*. In the next chapter, we will explain the VPP's approach to offering and managing the contracts, alongside its participation in the real-time electricity market, and how the EVs get (dis)charged while they are connected to the VPP.

Chapter 4

Optimal VPP Operation

This chapter explains the VPP's operation including processing arriving/departing EVs, (dis)charging them, and trading their available flexibility in the real-time market. We use a discrete-time model to optimize the VPP operation. This is motivated by the fact that the VPP participates in a real-time market that has one-hour resolution. In each time step (hour), the proposed algorithm goes through a loop that starts with receiving new arrivals and offering them a subset of the variable-term V2G contracts, then scheduling their charging based on the accepted contracts, stay times, energy demands, and price forecasts. Finally, the required amount of energy will be traded in the electricity market in that hour. This process is depicted in Fig. 4.1. In this chapter, we will model the optimal flexibility trading without discussing aggregation and disaggregation of flexibility of individual EVs. We will introduce trading aggregate flexibility in the next chapter.

The rest of this chapter is organized as follows. Initially, we explain the participation of the VPP in the real-time market. Subsequently, we delve into the formulation of equations modeling the EV charging dynamics. Following this, we outline the protocol through which the VPP offers V2G contracts and EVs accept them. Lastly, we design an optimization problem that aims to maximize the VPP's profitability within the constraints dictated by contract parameters and EV charging dynamics.

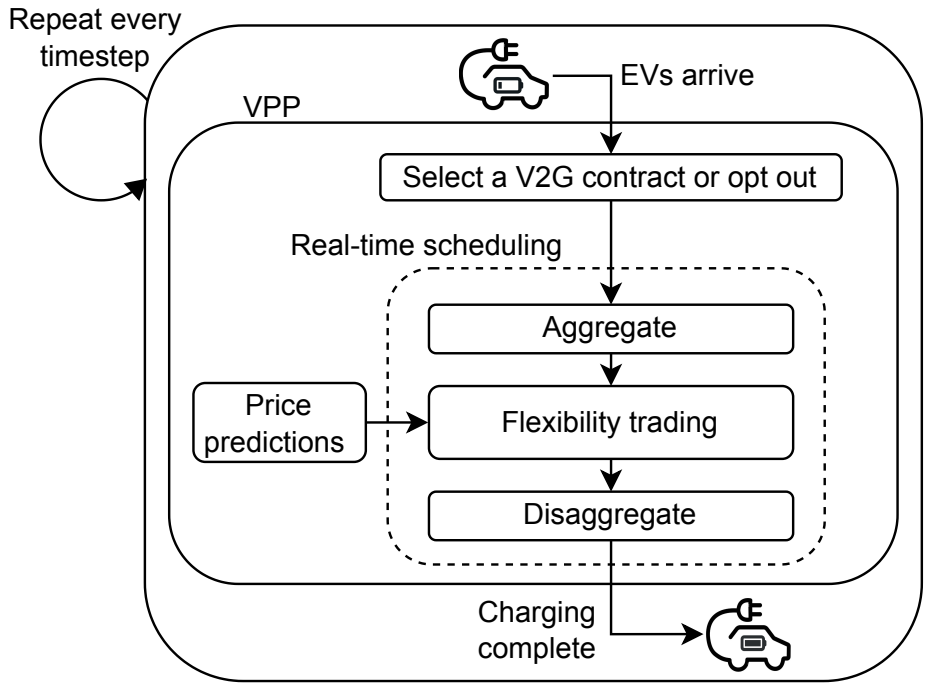

Figure 4.1: Overview of our methodology

4.1 Market Participation

The VPP is assumed to participate in an electricity market that runs once every hour so that electricity can be traded in real-time and uses a single price for buy and sell in each hour. The hourly price reflects market demand and is treated as an exogenous random variable. We denote the next $h+1$ hourly prices starting from hour t in vector form as $\mathbf{p}_{t:t+h} = (p_t, \ldots, p_{t+h})$. Note that every element of this vector is a random variable.

Although the trading strategy developed in this paper is suitable for participation in a single-stage electricity market, it can be used as part of a multi-stage optimization problem if the VPP participates in other electricity markets, such as the day-ahead market [70].

4.2 EV Charging Dynamics

The VPP must charge EVs in a way that ensures their energy demand will be satisfied by their specified deadline while abiding by the physical limitations of charging equipment. This is referred to as a *feasible* charging schedule. Given

the hourly timescale of the market, the VPP readjusts the (dis)charge power of chargers every hour. We denote the set of EVs that are plugged in at time t as $\mathcal{N}_t$ and the set of EVs that are plugged in and have accepted a V2G contract as $\mathcal{N}_t^D \subseteq \mathcal{N}_t$. Hence, $\mathcal{N}_t \setminus \mathcal{N}_t^D$ contains all EVs that have opted out. An EV $n \in \mathcal{N}_t$ arrives in the charging station at t_{arr}^n and departs at t_{dep}^n. At any t between the arrival and departure time, its remaining connection time is given by $\tau_t^n = t_{dep}^n - t$.

The amount of energy charged or discharged from each vehicle's battery at time t forms a vector $\mathbf{y}_t = (y_t^1, \ldots, y_t^{|\mathcal{N}_t|})$. To apply charge and discharge efficiencies, we divide each element y_t^n into a positive component (charged energy), denoted AC_t^n, and a negative component (discharged energy), denoted AD_t^n. We assume that all chargers are of the same type, with $\alpha_c > 0$ being their maximum charge power and $\alpha_d > 0$ being their maximum discharge power. Since the length of each timeslot is 1 hour, we reuse α_c and α_d to express the maximum amount of energy (in kWh) that can be charged or discharged from the battery in one timeslot. The constraints for EV charge and discharge energy are given below:

$$y_t^n = AC_t^n + AD_t^n \qquad \forall n \in \mathcal{N}_t \tag{4.1a}$$

$$0 \leq AC_t^n \leq \alpha_c \qquad \forall n \in \mathcal{N}_t \tag{4.1b}$$

$$AD_t^n = 0 \qquad \forall n \in \mathcal{N}_t \setminus \mathcal{N}_t^D \tag{4.1c}$$

$$-\alpha_d \leq AD_t^n \leq 0 \qquad \forall n \in \mathcal{N}_t^D \tag{4.1d}$$

Note that all of these constraints are defined only at times when EV n is connected to a charger, i.e. for $t \in \{t_{arr}^n, \ldots, t_{dep}^n\}$. We denote the energy capacity of its battery as B^n and its SOC at time t as soc_t^n. At t_{arr}^n, soc_t^n is initialized with the observed SOC at arrival, soc_{arr}^n. For the schedule to be feasible, by t_{dep}^n, soc_t^n must reach the SOC specified by the EV owner, denoted as soc_{dep}^n. Furthermore, soc_t^n must be maintained between minimum and maximum levels, δ_{min} and δ_{max}, at all times. While connected, soc_t^n is calculated using a recursive formula after accounting for the energy charged or discharged by the VPP and incorporating the battery charge and discharge efficiencies, denoted as η_c and η_d respectively. We assume that the battery

self-discharge is negligible at this timescale and can be ignored. The SOC constraints are given below:

$$\delta_{min} \leq soc_t^n \leq \delta_{max} \qquad \forall n \in \mathcal{N}_t \tag{4.2a}$$

$$soc_t^n = soc_{arr}^n \qquad \text{when } t = t_{arr}^n \tag{4.2b}$$

$$soc_t^n = soc_{dep}^n \qquad \text{when } t = t_{dep}^n \tag{4.2c}$$

$$soc_{t+1}^n = soc_t^n + \frac{\eta_c \, AC_t^n}{B^n} + \frac{AD_t^n}{\eta_d \, B^n} \qquad \forall n \in \mathcal{N}_t \tag{4.2d}$$

Lastly, the accepted V2G contracts pose additional constraints on the charging schedule. Let us use $\bar{w}_t^n$ and $\bar{\ell}_t^n$ to track the remaining energy that can be discharged and the remaining time to discharge this energy from the battery of the n^{th} EV. If this EV refuses to accept a contract, i.e. $n \in \mathcal{N}_t \setminus \mathcal{N}_t^D$, then both $\bar{w}_t^n$ and $\bar{\ell}_t^n$ are set to zero. Recall that w_i and ℓ_j are the initial parameters of the V2G contract. Thus, when EV n arrives at t_{arr}^n and accepts contract $(g_{i,j}, w_i, \ell_j)$, then we initialize $\bar{w}_t^n = w_i$ and $\bar{\ell}_t^n = \ell_j$. It follows from these definitions that every time the battery of this EV is discharged, we have to update $\bar{w}_t^n$ by subtracting the amount of energy that is discharged. Moreover, $\bar{\ell}_t^n$ is reduced by one in every timeslot regardless of whether the battery is discharged. An EV can be discharged only if $\bar{w}_t^n > 0$ and $\bar{\ell}_t^n > 0$. If one of them reaches zero, the EV is removed from $\mathcal{N}_t^D$ and put into $\mathcal{N}_t \setminus \mathcal{N}_t^D$, then both $\bar{w}_t^n$ and ℓ_t^n are set to zero.

$$\bar{w}_{t+1}^n = \bar{w}_t^n + AD_t^n/\eta_d \qquad \forall n \in \mathcal{N}_t^D \tag{4.3a}$$

$$\bar{\ell}_{t+1}^n = \bar{\ell}_t^n - 1 \qquad \forall n \in \mathcal{N}_t^D \tag{4.3b}$$

$$-AD_t^n \leq \bar{w}_t^n \eta_d \qquad \forall n \in \mathcal{N}_t^D \tag{4.3c}$$

Note how the discharge efficiency η_d is taken into account in Eqn.(4.3c). This is because the contract specifies how much energy can be withdrawn directly from the battery. But due to discharge inefficiency, the VPP will receive slightly less energy.

In summary, for EV n at time t, the variables that are updated at every timestep are: (i) state-of-charge, soc_t^n; (ii) time left until departure, τ_t^n; (iii) remaining energy available for discharge, $\bar{w}_t^n$; (iv) remaining term of the contract, $\bar{\ell}_t^n$.

4.3 Pruning V2G Contracts

At the start of each timeslot t, the VPP receives the list of newly connected EVs along with their battery capacity B^n, initial SOC soc^n_{arr}, arrival time t^n_{arr}=t, specified departure time t^n_{dep}, and desired SOC at departure soc^n_{dep}. With this information, the VPP calculates the *laxity* of each EV which is defined as the difference between the time left until departure and the minimum amount of time that is required to bring the battery SOC to their desired SOC. Calculating laxity is helpful because it must remain nonnegative at all times for the charging schedule to be feasible [68].

$$lax^n_t = \tau^n_t - \frac{(soc^n_{dep} - soc^n_t)B^n}{\alpha_c \eta_c} \qquad \forall n \in \mathcal{N}_t \qquad (4.4a)$$

$$lax^n_t \geq 0 \qquad \forall n \in \mathcal{N}_t \qquad (4.4b)$$

We assume that EVs arrive with non-negative laxity, so initially there is at least one feasible charging schedule. Note that the laxity at arrival can be calculated by substituting t^n_{arr} for t in Eqn. (4.4a).

Once the laxity is calculated for each EV, the VPP offers a subset of V2G contracts presented in Table 3.1 by pruning the V2G contracts that, if accepted, there will be no feasible charging schedule for the respective EV. To find this subset, the VPP performs three entry checks: (i) The EV does not depart before the contract term is over, $t^n_{dep} - t^n_{arr} \geq \ell_j$. (ii) The energy content of its battery at arrival is not less than the contract discharge energy, $B^n soc^n_{arr} \geq w_i$. (iii) The time required to discharge w_i from the battery and then charge it by the same amount does not exceed its laxity at arrival: $w_i \cdot \eta_d/\alpha_d + w_i/(\alpha_c \cdot \eta_c) \leq lax^n_{t_{arr}}$. A diagram explaining how contracts go through these checks and are added to the menu is shown in Figure 4.2.

If there are no feasible contracts or none of them provides positive utility to the EV owner, then they will opt out of V2G. In that case, their battery is still charged to the desired SOC, and their laxity is used to offer flexibility, which is less than the flexibility they could offer if they permitted bidirectional charging. Otherwise, the EV owner will choose the contract that corresponds to their type $(\theta^w_i, \theta^\ell_j) \in \Theta^w \times \Theta^\ell$ as the contracts are incentive compatible. If

the contract designed specifically for their type is not feasible, the EV owner will choose one with a lower discharge energy or a shorter term (whichever maximizes their utility), as long as it is feasible and provides positive utility as outlined in Eqn. (3.1).

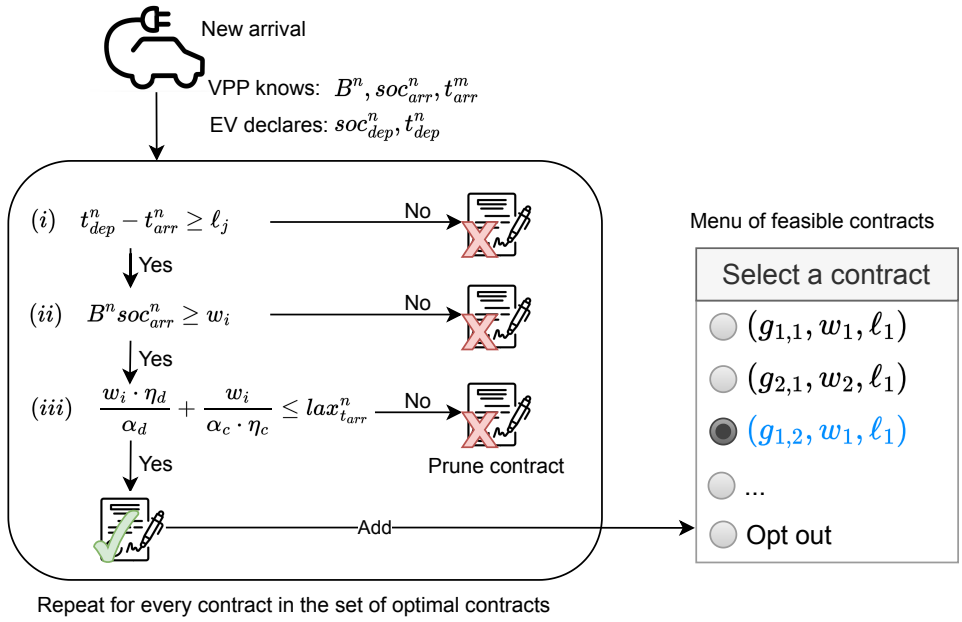

Figure 4.2: A demonstration of how a VPP contract menu is constructed for each EV.

4.4 Optimal VPP Scheduling

Once the set of newly arrived EVs that accepted a V2G contract is determined, the VPP decides how much energy it should trade in in the real-time market by solving an optimization problem to maximize its total profit by minimizing the cost of trading in the market. Indeed, this problem is a stochastic optimization problem due to the uncertainty of EV mobility and hourly prices. A fast and robust approach to trading flexibility under uncertainty will be discussed in the next section. For now, we assume perfect information about the future and solve this problem in an offline fashion. This serves as a baseline enabling us to quantify the optimality gap due to uncertainty in Chapter 6.

The cost of trading in the market depends on the net energy delivered to

EVs in that timeslot:

$$y_t^{agg} = \sum_{n \in \mathcal{N}_t} y_t^n \tag{4.5}$$

Note that y_t^{agg} would be negative if the amount of energy discharged from EVs surpassed the amount of energy used to charge them.

Problem (4.6) finds the charging schedule that minimizes the cost of trading in the market from the current time $t_\emptyset$ up to the next h hours. For the solution to be the true optimal, the optimization horizon should be the maximum departure time of all the cars currently connected. That is $h = \max(\tau_t^n,\ \forall n \in \mathcal{N}_t)$. The cost is calculated by multiplying the predicted market price for that hour by the net energy delivered to EVs in that hour, y_t^{agg}. For the solution to be optimal, price predictions are assumed to be perfect. The main optimization variable is a matrix containing the charging amount for every currently connected EV $n \in \mathcal{N}_{t_\emptyset}$ from the current time $t_\emptyset$ up to $t_\emptyset + h$, that is $\mathbf{Y}_{t_\emptyset:t_\emptyset+h} = \left[\mathbf{y}_{t_\emptyset}, \ldots, \mathbf{y}_{t_{\emptyset+h}}\right]$, with each $\mathbf{y}_t = \left[y_t^1, \ldots, y_t^{|\mathcal{N}_{t_\emptyset}|}\right]$. Note that $\mathbf{AC}_{t_\emptyset:t_\emptyset+h}$ and $\mathbf{AD}_{t_\emptyset:t_\emptyset+h}$ are auxiliary variables that determine $\mathbf{Y}_{t_\emptyset:t_\emptyset+h}$.

$$\underset{\mathbf{Y}_{t_\emptyset:t_\emptyset+h};\, \mathbf{AC}_{t_\emptyset:t_\emptyset+h};\, \mathbf{AD}_{t_\emptyset:t_\emptyset+h}}{\text{minimize}} \quad \sum_{t=t_\emptyset}^{t_\emptyset+h} \hat{p}_t \cdot y_t^{agg} \tag{4.6}$$

subject to:

Charging constraints Eqns. (4.1)

State-of-charge constraints Eqns. (4.2)

V2G contract constraints Eqns. (4.3)

Remark 4.4.1. *In the above problem, we do not need a constraint to explicitly prevent a battery from being charged and discharged at the same time, i.e. forcing* $AC_t^n AD_t^n = 0$. *Due to battery imperfections, such opposing actions would waste energy and are therefore suboptimal.*

Remark 4.4.2. *Since there are no coupling constraints between EVs in Problem* (4.6)*, the VPP does not need predictions for arrivals that will occur between* $t_\emptyset$ *and* $t_\emptyset + h$ *in order to find the optimal solution for the current time* $t_\emptyset$.

Observe that in Problem (4.6), the objective function is linear and all constraints are affine. Thus, it is a linear program that can be solved in polynomial time using interior point or simplex-based methods. Nevertheless, when the VPP aggregates a large number of EVs and some of them remain connected to a charger for an extended period of time, solving this problem with noisy or perfect predictions takes a considerable amount of time because of the optimization horizon and the number of decision variables and constraints.

Let us denote the solution of the above optimization problem as $\mathbf{Y}^{\star}_{t_\emptyset:t_\emptyset+h}$. The first row of this matrix, $\mathbf{y}^{\star}_{t_\emptyset}$, denotes the optimal decision that must be implemented at $t = t_\emptyset$.

4.5 Summary

In this chapter, we explained the VPP's participation in the market, its management of V2G contracts, and its approach to EV charging while ensuring compliance with all the constraints. However, the optimization problem outlined here, Problem (4.6), exhibits two primary limitations. Firstly, it relies on the assumption of perfect electricity price predictions, making it potentially brittle to noisy forecasts, a common occurrence in real-world scenarios. Secondly, the number of constraints depends on the number of connected EVs at any given timestep, consequently prolonging the time needed to find the solution. In the worst-case scenario, where a considerable number of EVs is connected, the optimization problem might not be solved within the allotted time resolution of 1 hour. To mitigate these challenges, the next chapter will introduce a Reinforcement Learning agent capable of taking an aggregate action within the region defined by aggregate constraints. We will derive the aggregate constraints, detail how the agent takes an aggregate action, and illustrate the transformation of this aggregate action into individual charging or discharging actions. We will keep Problem (4.6) as a baseline for our experiments in Chapter 6 to evaluate these claims.

Chapter 5

Trading Flexibility in Aggregate

In this chapter, we present our methodology for solving the stochastic optimization problem presented in Chapter 4. Our goal is to design a scalable and efficient algorithm for flexibility trading that does not rely on accurate forecasts, and its running time does not change drastically as more EVs are controlled by the VPP. Moreover, it should guarantee that there is a feasible charging schedule for all EVs regardless of whether they accepted a V2G contract, and that EVs are treated fairly. To achieve this goal, we attempt to trade flexibility in aggregate using an RL agent that, once trained, can be quickly executed.

The proposed methodology has three steps which are shown inside the dashed box in Fig. 4.1. First, we create an aggregate representation of the state, denoted as $\mathbf{s}_t^{agg}$, by aggregating individual EVs into a virtual battery. In this process, we also aggregate constraints defined for individual EVs to ensure that the action taken for the virtual battery will not violate a constraint defined for an EV. In the second step, we pass the aggregate state representation to a safe RL agent that outputs an action to operate the virtual battery. This action is denoted as y_t^{agg} and minimizes the expected cost of trading in the market. Due to the way that we aggregate constraints and the design of our RL agent, this action is guaranteed to satisfy constraints defined for individual EVs after it is disaggregated to their respective actions. In the last step, we disaggregate y_t^{agg} into $(y_t^1, \ldots, y_t^{|\mathcal{N}_t|})$ by incorporating a fairness criterion.

In Figure 5.1, we illustrate the concepts of aggregation and disaggregation.

Subplot **(a)** depicts the upper and lower bounds for the (dis)charging power that each EV can receive at time t to maintain a feasible charging session. These bounds are summed to create aggregate upper and lower bounds, labeled as *Agg.* In subplot **(b)**, we show the process of disaggregation. Initially, an aggregate action, indicated by the grey bar, is taken within the aggregate bounds. This action is then broken down into individual actions, represented by the blue bars, ensuring each falls within its respective individual bounds. The sum of these individual actions must match the aggregate action.

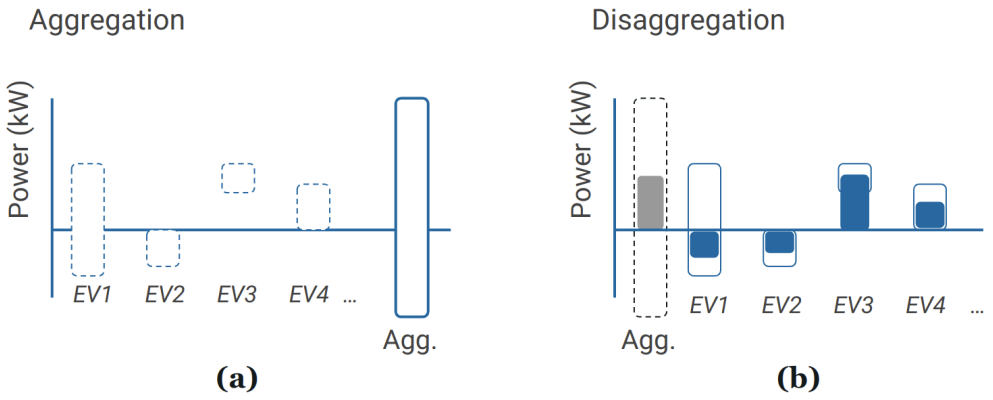

Figure 5.1: Illustration of aggregation and disaggregation schemes, shown in subplots **(a)** and **(b)** respectively.

5.1 Aggregation

Aggregating the energy storage capacity of individual EVs into a virtual battery is essential for using a single-agent reinforcement learning framework for real-time scheduling of flexibility, independent of the number of connected EVs. The main challenge in learning an action for (dis)charging this virtual battery is ensuring that it does not violate the feasibility requirement of individual charging schedules. In this section, we show that this can be achieved by carefully aggregating the constraints defined for the (dis)charge action of each individual EV, i.e. y_t^n, in Problem (4.6). Specifically, we inspect the constraints defined for individual EVs and identify those that can be translated to upper and lower bounds for y_t^n. This allows us to aggregate lower and upper

bounds separately to obtain bounds for y_t^{agg}.

We first examine Eqn. (4.1b) which states that $0 \leq AC_t^n \leq \alpha_c$. Since we know from Eqn. (4.1a) that $y_t^n = AC_t^n + AD_t^n$ and that AC_t^n and AD_t^n cannot be nonzero at the same time at the optimal point (see Remark 4.4.1), we can rewrite (4.1b) as $0 \leq y_t^n \leq \alpha_c$ when the battery is charging. From this, we get an upper bound on y_t^n:

$$y_{t,upper1}^n = \alpha_c \quad \forall n \in \mathcal{N}_t \tag{5.1}$$

Using the same arguments, we get a lower bound on y_t^n from combining Eqns. (4.1c) and (4.1d):

$$y_{t,\,lower1}^n = -\alpha_d \quad n \in \mathcal{N}_t^D \tag{5.2}$$

For EVs that accepted a V2G contract, Eqn. (4.3c) forces the VPP to respect the discharge energy in the contract. This gives the following lower bound:

$$y_{t,\,lower2}^n = -\bar{w}_t^n \eta_d \quad n \in \mathcal{N}_t^D \tag{5.3}$$

Next, we consider the SOC constraints in (4.2a) and (4.2d). By substituting (4.2d) in (4.2a) for the two cases where the battery charges and discharges, we get an upper bound and a lower bound on y_t^n:

$$y_{t,upper2}^n = B^n(\delta_{max} - soc_t^n)/\eta_c \qquad \forall n \in \mathcal{N}_t \tag{5.4}$$

$$y_{t,lower3}^n = B^n \eta_d(\delta_{min} - soc_t^n) \qquad \forall n \in \mathcal{N}_t^D \tag{5.5}$$

Finally, we turn our attention to constraints (4.4a) and (4.4b) which deal with laxity. Notice that y_t^n should maintain the non-negativity of laxity in the next step, as $lax_{t+1}^n \geq 0$ indicates that it is possible for the SOC of this EV to reach soc_{dep}^n by t_{dep}^n. To get the definition of lax_{t+1}^n, we write (4.4a) for $t+1$. We then substitute (4.2d) in (4.4a) to relate laxity at $t+1$ to the action taken at t, and replace τ_{t+1}^n with $\tau_t^n - 1$. To obtain a lower bound for y_t^n, we set lax_{t+1}^n to zero. Finally, we consider the two cases where $y_t^n = AC_t^n > 0$ and $y_t^n = AD_t^n < 0$ separately as the efficiencies are handled differently in Eqn. (4.2d). Solving each case for y_t^n, we get Eqns. (5.5) and (5.6):

$$y_{t,lower4}^n = B^n(soc_{dep}^n - soc_t^n)/\eta_c - \alpha_c(\tau_t^n - 1) \qquad \forall n \in \mathcal{N}_t \tag{5.6}$$

$$y_{t,lower5}^n = B^n \eta_d(soc_{dep}^n - soc_t^n) - \alpha_c \eta_c \eta_d(\tau_t^n - 1) \qquad \forall n \in \mathcal{N}_t^D \tag{5.7}$$

Now that we have all constraints posed on y_t^n, we can write the overall individual upper and lower bounds as follows:

$$\begin{aligned} y^n_{t,\,upper} &= \min\left(y^n_{t,\,upper1}, y^n_{t,\,upper2}\right) && \forall n \in \mathcal{N}_t \\ y^n_{t,\,lower} &= y^n_{t,\,lower4} && \forall n \in \mathcal{N}_t \setminus \mathcal{N}_t^D \\ y^n_{t,\,lower} &= \max\left(y^n_{t,\,lower1}, \ldots, y^n_{t,\,lower5}\right) && \forall n \in \mathcal{N}_t^D \end{aligned}$$

Finally, we get the upper (lower) bound for the action of the virtual battery, y_t^{agg}, by aggregating the upper (lower) bounds of individual EVs:

$$\sum_n y^n_{t,\,lower} \leq y_t^{agg} \leq \sum_n y^n_{t,\,upper} \tag{5.8}$$

It follows from the above derivations that if the y_t^{agg} returned by the RL policy satisfies (5.8), there exists a disaggregation of y_t^{agg} into $(y_t^1, \ldots, y_t^{|\mathcal{N}_t|})$ such that every element of this vector satisfies the feasibility requirement of the respective EV charging schedule. We denote the aggregate lower and upper bounds in Eqn. (5.8) as $y^{agg}_{t,\,lower}$ and $y^{agg}_{t,\,upper}$, respectively.

5.2 Soft Actor-Critic Reinforcement Learning

To schedule the charge and discharge of the virtual battery, we use the Soft Actor-Critic (SAC) algorithm which trains a stochastic policy with entropy regularization [37]. Our SAC agent, called *Aggregate SAC*, receives the aggregate state representation (defined below) and outputs the (dis)charge action of the virtual battery. This action determines the amount of energy that will be traded in the market in real-time. The three main components of Aggregate SAC are the actor, critic, and replay buffer, as shown in Fig. 5.2.

5.2.1 Aggregate State

To make the state of the SAC agent truly independent of the number of connected EVs, we argue that the number of state variables (dimension of the state space) and the scale of each state variable should be independent of the number of connected EVs. This leads us to design an aggregate *state*

representation denoted as $\mathbf{s}^{agg}$. Specifically, we take the average of features characterizing individual EVs, such as SOC, upper and lower bounds of the action, etc. Additionally, price predictions for the current time step and the next h time steps, $\hat{\mathbf{p}}_{t:t+h}$, and its discrete derivative, $\Delta\hat{\mathbf{p}}_{t:t+h}/\Delta t$, are included in the state. Below, we provide the full list of state variables. All of these variables are concatenated and the resulting 1D vector is fed to the agent.

Aggregate State Definition

The following features constitute the state of our RL agent. Firstly, we consider the features that describe EVs:

- Average upper bound; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y^n_{t,upper}$.
- Average lower bound; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y^n_{t,lower}/N_t$.
- Average minimum power to maintain non-negative laxity; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} y^n_{t,lower4}$
- Average state of charge of the EVs; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} soc^n_t$
- Average energy demand of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t}\left(soc^n_{dep}-soc^n_t\right)$
- Average remaining time connected of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t}\tau^n_t$
- Average laxity of the EVs.; $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} lax^n_t$
- Proportion of connected cars with contracts; $|\mathcal{N}^D_t|/|\mathcal{N}_t|$;
- Average energy available for discharge in EVs with contracts; $\frac{1}{|\mathcal{N}^D_t|}\sum_{n\in\mathcal{N}^D_t}\bar{w}^n_t$
- Average remaining time of contracts; $\frac{1}{|\mathcal{N}^D_t|}\sum_{n\in\mathcal{N}^D_t}\bar{l}^n_t$

Note that for the features normalized by N_t, if $N_t = 0$ then the feature is set to 0. The same applies to the features normalized by N^D_t. In addition to these aggregate features, the agent also receives some additional features about the rest of the environment:

- One-hot encoded hour of the day
- One-hot encoded day of the week
- Price predictions over the next h hours; $\hat{\mathbf{p}}_{t:t+h}$. Experimentally, we used $h = 8$.
- Discrete derivative of the price predictions; $\frac{\Delta\,\hat{\mathbf{p}}_{t:t+h}}{\Delta t}$
- Average slope of the price over the next h hours; $\frac{1}{h-1}\sum_{i=0}^{h-1}(\hat{p}_{t+1+i}-\hat{p}_{t+i})$

5.2.2 Reward

The reward signal is defined as $r_t = -p_{t-1} \cdot y_{t-1}^{agg}$ which is the negated single-step version of the objective function of Problem (4.6). As the SAC agent maximizes the cumulative reward over an episode, the reward must be defined as the negative transfer to the imbalance market at a single timeslot t. Note that $r_t > 0$ implies that the VPP receives money from trading in the market, while $r_t < 0$ implies that the VPP pays for the trade.

5.2.3 Actor and Critic Networks

The actor is a neural network that takes the aggregate state $\mathbf{s}_t^{agg}$, and producing the mean and standard deviation, μ_t and σ_t, of a Gaussian distribution from which a coefficient $\bar{\beta}_t$ is sampled. We use the *sigmoid* activation function in the last layer before outputting μ_t to ensure that this coefficient falls in the range of $[0, 1]$. The sampled $\bar{\beta}_t$ is then clipped between 0 and 1 to obtain the final coefficient β_t. Subsequently, β_t is used for linear interpolation between the upper and lower bounds, $y_{t,\,upper}^{agg}$ and $y_{t,\,lower}^{agg}$ respectively:

$$y_t^{agg} = \beta_t \cdot y_{t,\,upper}^{agg} + (1 - \beta_t) \cdot y_{t,\,lower}^{agg} \tag{5.9}$$

The resulting y_t^{agg} represents the total amount of energy that must be charged or discharged from the virtual battery at t. This value is then passed to the disaggregation algorithm described in Section 5.3) to determine individual (dis)charging actions.

The other two components, the critic and replay buffer, play central roles in the model-free RL algorithm outlined in [37]. Essentially, the SAC algorithm seeks to find the policy that maximizes an objective function, combining the accumulated expected reward and the entropy of the policy at any given state. The entropy encourages the policy to act more randomly, promoting exploration. The extent to which the entropy is considered is tuned by a temperature term. The critic learns a value function, indicating the expected reward of selecting an action at a given state. This helps to provide learning updates to the actor to enhance the policy. In our implementation, tuples of

state, action, reward, and next state are saved to the buffer, and subsequently sampled to perform updates to the actor and the critic (labeled in Fig. 5.2 as $(\mathbf{s}_i^{agg}, \bar{\beta}_i, r_i, \mathbf{s}_{i+1}^{agg})$).

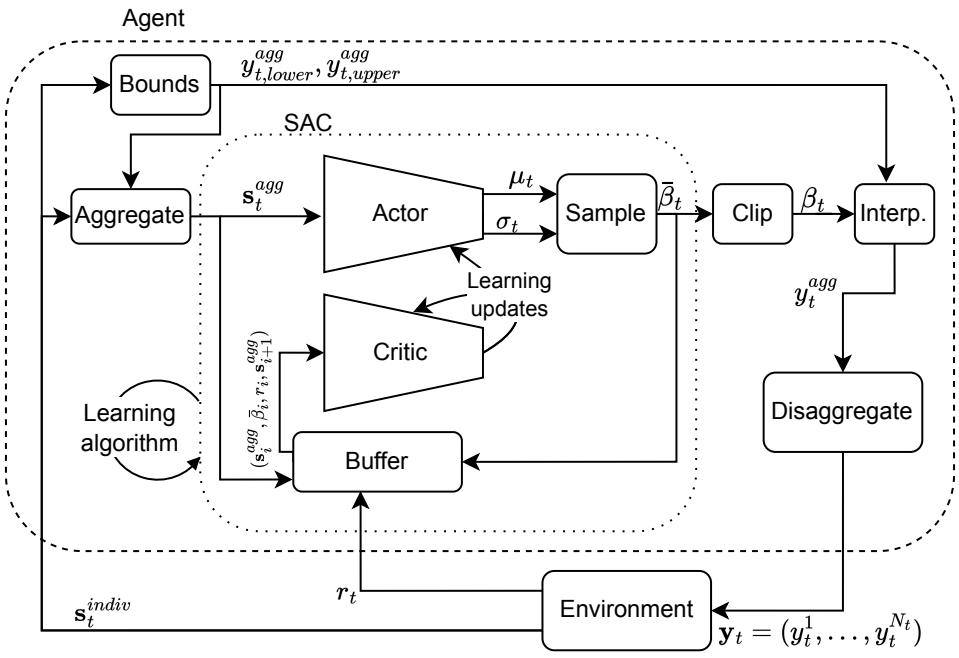

Figure 5.2: Block diagram of the agent, *Aggregate SAC*.

5.3 Proportionally Fair Disaggregation

The VPP employs a disaggregation algorithm to compute $\mathbf{y}_t = (y_t^1, \ldots, y_t^{|\mathcal{N}_t|})$, which is a feasible charging schedule for individual EVs, given y_t^{agg} returned by the RL agent. While there are various ways to disaggregate y_t^{agg}, we compute $\mathbf{y}_t$ by solving a convex optimization problem. In particular, we maximize the sum of the logarithm of every EV owner's utility function, where the utility function of EV owner n is given by:

$$U_t^n(y_t^n) = y_t^n - y_{t,lower}^n + 1 \tag{5.10}$$

This utility function is greater than or equal to 1 and increasing in y_t^n. We wish to emphasize that this function is different from (3.1) and is defined for all EVs regardless of whether they have accepted a V2G contract. The following

optimization problem is solved at every timeslot.

$$\underset{\mathbf{y}_t=\left(y_t^1,\ldots,y_t^{|\mathcal{N}_t|}\right)}{\text{maximize}} \quad \sum_{n\in\mathcal{N}_t} \log\left(U_t^n(y_t^n)\right) \tag{5.11a}$$

subject to:

$$\sum_{n\in\mathcal{N}_t} y_t^n = y_t^{agg} \tag{5.11b}$$

$$y_{t,\,lower}^n \leq y_t^n \leq y_{t,\,upper}^n \qquad \forall n \in \mathcal{N}_t \tag{5.11c}$$

The significance of this formulation is that the solution of this convex problem, which we call a *proportionally fair* disaggregation, satisfies fairness axioms from game theory [101]. We have borrowed the notion of proportional fairness from the resource allocation literature to ensure that all EVs are treated fairly.

5.4 Priority-Based Disaggregation

To evaluate the performance of the *proportionally fair* disaggregation algorithm, we borrow a priority-based resource allocation algorithm and use it for flexibility disaggregation. The basic idea of a priority-based algorithm is that we can assign an arbitrary scalar that signifies priority to each EV n at every timeslot t. Let us denote this priority as pr_t^n. The algorithm sorts the EVs from highest to lowest priority and they receive their share of y_t^{agg} in that order. Concretely, the algorithm receives the priority list $\mathbf{pr}_t$, the aggregate action of the virtual battery (y_t^{agg}) which is either charge or discharge this amount of energy depending on its sign, and the upper and lower bounds of the action of each EV (y_t^n). Next, it starts by fulfilling the lower bound of each EV. If there is surplus energy after that, the maximum possible energy is allocated to the EV with the highest priority. It continues down the list until y_t^{agg} is used up. The pseudocode for this algorithm is presented in Section 5.4. In Chapter 6, we define the priority of each EV based on its laxity. This results in two versions of this algorithm, namely *most laxity first* and *least laxity first*.

Algorithm for Priority-Based Disaggregation

Algorithm 1 starts by calculating the indices that would sort the priority array, $\mathbf{pr}_t$, in descending order. These indices will allow us to loop through the other

Algorithm 1: Priority-Based Disaggregation

Inputs:

Total energy available y_t^{agg}

Individual upper bounds $\mathbf{y}_{t,\,upper} = \{y_{t,\,upper}^1, \ldots, y_{t,\,upper}^{N_t}\}$

Individual lower bounds $\mathbf{y}_{t,\,lower} = \{y_{t,\,lower}^1, \ldots, y_{t,\,lower}^{N_t}\}$

Priority of each EV $\mathbf{pr}_t = \{pr_t^1, \ldots, pr_t^{N_t}\}$

Output:

Energy allocated to each car $\mathbf{y}_t = \{y_t^1, \ldots, y_t^{N_t}\}$

```
1  index_pr ← indices that would sort pr_t by descending;
2  y_t ← y_{t,lower};
3  y^agg_{t,surplus} ← y^agg_t − Σ(y_{t,lower});
4  for i ← 1 to |N_t| do
5  |   n ← index_pr[i];
6  |   y_temp ← min(y^agg_{t,surplus}, (y^n_{t,upper} − y^n_{t,lower}));
7  |   y^n_t ← y^n_t + y_temp;
8  |   y_{t,surplus} ← y_{t,surplus} − y_temp;
9  |   if y^agg_{t,surplus} == 0 then
10 |   |   break;
11 |   end
12 end
13 return y_t;
```

arrays from highest to lowest priority. Line 2 sets the output vector $\mathbf{y}_t$ to $\mathbf{y}_{t,\,lower}$, because each EV must have at least its lower bound met. Next, in Line 3, we subtract the energy we just allocated from the total available. The result is stored in the new variable $y_{t,\,surplus}$.

We then enter the main loop. During the first iteration, we select, in Line 5, the index of the EV with the highest priority, and the EVs with progressively lower priorities will be selected in later iterations. In Line 6, we check if we can assign all the surplus energy, $y_{t,\,surplus}$ to EV n. If not, we saturate it to its upper bound. In Line 7, that energy is allocated, and in Line 8 it is subtracted from the surplus. The loop breaks when all of the surplus energy has been allocated.

5.5 Implementation Considerations

The code, available at `https://github.com/J27avier/EvCharge`, is structured in a Gym-like fashion, following Farama's Gymnasium [88]. There are separate classes for the environment and the agent as shown in Figure 5.3. The different agents (Aggregate SAC and the baselines) can be swapped without changing the environment class. Furthermore, the settings for each run are passed as command-line arguments. This setup streamlines the experimentation and ensures consistent results across all the agents. An example of the code for this implementation is given in Appendix A.

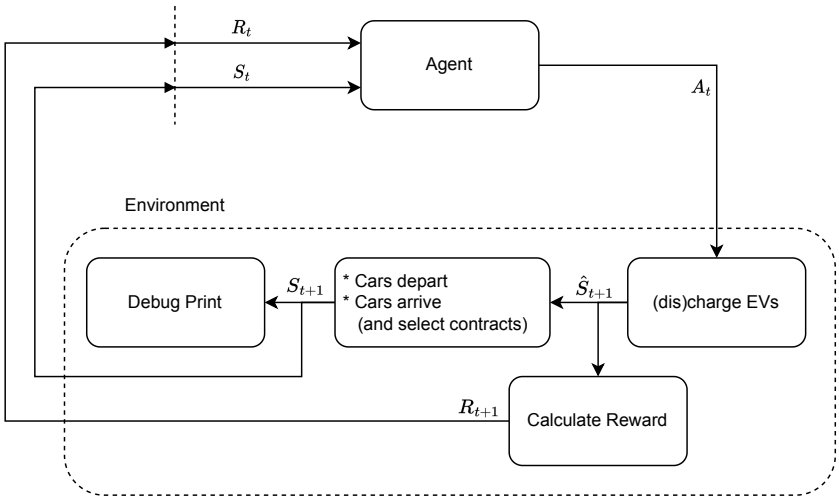

Figure 5.3: Diagram of the main blocks in the `EvCharge` Gym code.

5.6 Summary

In this chapter, we presented our design for a soft actor-critic reinforcement learning agent that trades flexibility in the real-time market. We presented the aggregation algorithm together with the aggregate state representation. Moreover, we covered our approach to guaranteeing that the agent operates within the specified constraints. We also presented two options for disaggregation. First is the *proportionally fair* disaggregation, which relies on solving an optimization problem. Second, we presented an algorithm for performing priority-based disaggregation. With this groundwork established, let us move

on to the experimental chapter. There we will explore the performance of our approach, comparing it against baselines, and analyzing how different factors can have an effect on its performance.

Chapter 6

Experimentation

In this chapter, we evaluate the proposed methods for energy trading in the real-time market through aggregation and disaggregation of flexibility provided by individual EVs. We begin by introducing the datasets used in our evaluation, detailing the data preparation procedures, and providing a brief overview through visualization. Following this, we establish the baselines, which are derived from the optimization problem outlined in Chapter 4, in particular Problem (4.6). We investigate the operation of the VPP through five experiments. Initially, we analyze the proportion of EVs accepted V2G contracts and which contract they chose. Next, we look at the performance of the *Aggregate SAC* agent during training and test under various forecasting conditions, and compare it against the baselines. Subsequently, we assess how this performance impacts the overall profitability of the VPP. Later, we evaluate the effect of different disaggregation algorithms on the agent's real-time market performance. Finally, we quantify the time required for the *Aggregate SAC* agent to generate a charging schedule for our dataset, comparing it against the baselines.

6.1 Datasets & Baselines

We use two real datasets that contain data collected between Jan. 1 and Dec. 31, 2019, from the Netherlands. The first dataset contains settlement prices in the imbalance market operated by TENNET [87]. This market has a 15-minute timescale, so to make it compatible with our 1-hour timeslots, we take

the price at the top of every hour. This results in a total of $365 \times 24 = 8,760$ data points.

The second dataset is the ElaadNL dataset [28]. It contains 10,000 charging sessions that took place in 2019 in a network of public charging stations in the Rotterdam region. Each session is characterized by a charger ID in addition to the arrival time t^n_{arr}, departure time t^n_{dep}, and amount of energy delivered to the EV that connected to this charger. To calculate the SOC at arrival, we assume each EV was charged to $soc^n_{dep} = 0.97$ before departure. Then, we calculate its soc^n_{arr} based on the amount of energy delivered in that charging session. We assume all chargers support bidirectional charging, their rated charge and discharge power is $\alpha_c, \alpha_d = 11$ kW, and the battery charge and discharge efficiency is $\eta_c, \eta_d = 0.98$. The SOC bounds are set to $\delta_{min} = 0$ and $\delta_{max} = 1$, and the energy capacity of the battery is set to $B = 80$ kWh.[1] While the methodology is designed to handle a heterogeneous set of EVs, we chose this to simplify the data pre-processing. As discussed in Chapter 4, we assume that all EVs have non-negative laxity upon arrival, so we discard all EVs with negative laxity in the dataset. That leaves us with $9,997$ sessions. Fig. 6.1 shows the box plot of hourly prices throughout the year. The number of daily sessions is depicted in Fig. 6.2a. Histograms of arrival and departure times on each day are shown in Fig. 6.2b. Finally, Fig. 6.2c shows the histogram of laxity at arrival time.

When the EVs arrive at charging stations operated by the VPP, their type with respect to energy and persistence is sampled from a discrete uniform distribution between 1 and 3. We build the contract menu for each EV based on their specified charging deadline and energy demand after applying the entry checks outlined in Section 4.3. As a result, a subset of the nine contracts presented in Table 3.1 will be offered to that EV.

As we need some of the data to train the RL agent, we split the price and EV charging datasets into training and test. The training dataset spans from January to June, while the test dataset spans from July to December.

[1] This is approximately the capacity of common EV models such as Hyundai IONIQ 5 (77.4 kWh), Tesla Model 3 Long Range (82 kWh), and Ford Mustang Mach E (88 kWh).

Thus, the training set contains 4,764 sessions, and the test set contains 5,233 sessions. Unless otherwise stated, the main disaggregation algorithm that is used in our experiments is the *proportionally fair* algorithm, and the real-time scheduling algorithm that is based on this disaggregation scheme is called *Aggregate SAC PF*.

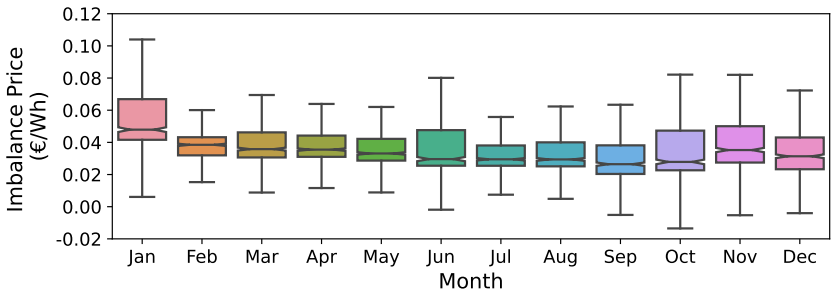

Figure 6.1: Distribution of prices in the imbalance market, whiskers show 1.5×IQR.

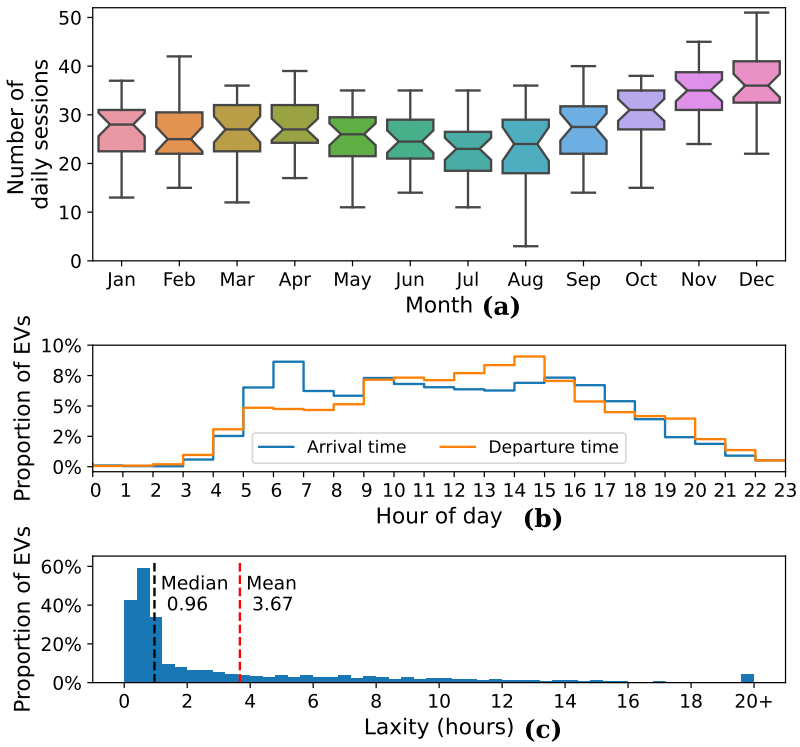

Figure 6.2: EV charging sessions dataset. Subplot (a) shows a boxplot of daily sessions per month, whiskers are 1.5×IQR. Subplot (b) shows a histogram of arrivals and departures. Subplot (c) shows a histogram of laxity at arrival.

Baselines We use three baselines to evaluate our approach. The first is *No-Control*, when an EV is charged at the maximum power supported by the charger as soon as it connects to the charger. This 'charging as soon as possible' policy minimizes the length of the charging session without taking advantage of any kind of flexibility. The second is *No-V2G* which solves Problem (4.6) with perfect information, but without offering any V2G contracts. In this case, the VPP only uses the flexibility allowed by the EVs' laxity at arrival. Lastly, we have *OPT-V2G* which solves Problem (4.6) in an offline fashion, taking into account the V2G contracts. As we solve this problem with perfect information, it gives the best solution that can be possibly achieved. For a fair comparison with *Aggregate SAC PF*, we also solve Problem (4.6) with noisy price predictions. In that case, we renamed it to *LP-V2G* since it no longer represents the optimal solution.

Implementation Details The VPP is implemented in an in-house environment developed in Python 3.10 on an Ubuntu 22 machine with an AMD EPYC 7313 CPU (64 cores), onboard ASPEED GPU, and 516 GB of RAM. We model the optimization problems (3.3), (4.6) and (5.11) in CVXPY [27] and solve them using Mosek [8]. The implementation of *Aggregate SAC* agent is based on the code provided by CleanRL's continuous-action SAC [42] for PyTorch [63].

6.2 Contract Acceptance

We first analyze how the EV owners accept or opt out of the variable-term V2G contracts. Fig. 6.3 depicts the distributions of V2G contracts that were accepted in our experiment. Interestingly, most EV owners opted out and only 21% of them accepted a V2G contract. This is because their energy or time type creates a contract menu without any feasible contracts or with contracts that provide negative utility. For example, many EVs do not have enough laxity at arrival to participate in V2G for 5 hours or more. This is evident from Fig. 6.2c. Focusing on the EV owners that accepted a contract, we see

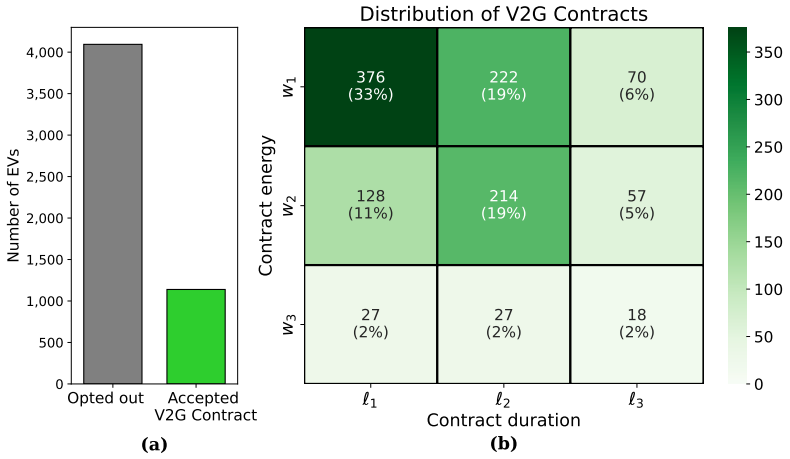

Figure 6.3: Plot (a) shows EVs that opted out or accepted a V2G contract; (b) shows the distribution of accepted contracts.

that the most popular contract is $(g_{1,1}, w_1, \ell_1) = (0.59\ €, 19.01\ \text{kWh}, 5\ \text{hr})$ with 33% of the total share of accepted contracts. As the contract increases in energy or duration, it is selected less frequently.

6.3 Evaluation of Aggregate Trading Agent

Training and Testing The *Aggregate SAC PF* agent is trained via interaction with the environment in discrete time steps, where each episode of interaction starts on January 1, 2019, and ends on June 30, 2019. We consider 200 episodes for training and deploy the policy obtained at the end of episode 200 to trade EV charging flexibility in the imbalance market during the second half of the year (July to December 2019). Thus, the training and testing episodes will have nearly the same number of 1-hour time steps, that is 4,344 and 4,416 time steps respectively. We continue making learning updates after the agent is deployed, provided they are done based on information we have seen from deployment till now.

Fig. 6.4a shows the learning curve of the Aggregate SAC PF agent. The agent is trained on perfect price predictions. In accordance with best practices [64], we perform five independent runs. The shaded area shows the cost (lower is better) of the best and worst performing agents in every episode (i.e. the tolerance bounds), and the solid line shows the average performance. The

horizontal dotted lines show the baseline performance for comparison. We can see that there are some variations in the agents during the early episodes (5-25), but the gap narrows as more episodes are used for training. Furthermore, this figure shows that during training, the *Aggregate SAC PF* settles on a performance around 40% higher than *OPT-V2G*.

When we evaluate the agent, we consider different qualities of price forecast. These are generated by adding synthetic noise to the prices: $\hat{p}_t = p_t + \epsilon_t$. Here, ϵ_t is sampled from a normal distribution with a mean of 0 and a standard deviation $\sigma = 0, 0.01, 0.02, 0.04, 0.06$. The case where $\sigma = 0$ corresponds to perfect predictions, and the other ones correspond to progressively worse predictions. To directly relate σ to the quality of price forecasts, we demonstrate in Table 6.1 how a certain σ affects the coefficient of determination, R^2, of the prediction. As explained in [51], the coefficient of determination is defined as $R^2 = 1 - \sum_{t=1}^{T}(p_t - \hat{p}_t)^2 / \sum_{t=1}^{T}(p_t - \bar{p}_t)^2$, with $\bar{p}_t$ being the mean price.

σ	0.00	0.01	0.02	0.04	0.06
R^2	1.000	0.974	0.897	0.588	0.075

Table 6.1: Relating synthetic noise ϵ_t to R^2

After letting the agent train for 200 episodes (using perfect information during training), we perform five runs on each of our forecast scenarios. These are shown in Fig. 6.4b, where the x-axis shows the amount of noise added to the predictions when the agent is deployed, and the y-axis shows the transfer to the imbalance market during the test episode. A small offset along the x-axis is added to the points to avoid overlaps. As before, the shaded area shows the best and worst performance. We compare the performance of *Aggregate SAC PF* to the two baselines, which were run five times as well. Recall that *LP-V2G* solves the same problem as *OPT-V2G*, but with noisy predictions.

When predictions are perfect ($\sigma = 0$), *Aggregate SAC PF* achieves a transfer to market that is lower (better) than *No-V2G* and, as expected, higher than *LP-V2G* which is equivalent to *OPT-V2G* in this case. The good performance of *Aggregate SAC PF* shows that it is able to translate what it learned during training to the new charging sessions and prices in the test set. There-

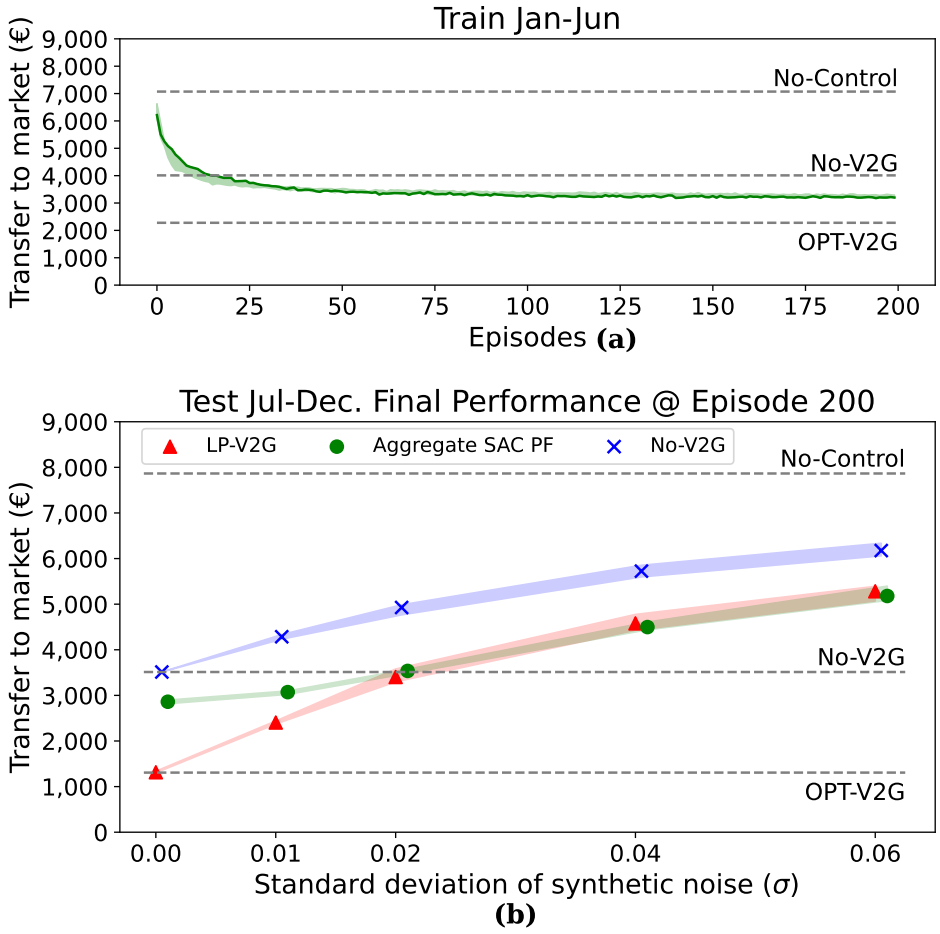

Figure 6.4: *Aggregate SAC PF* with perfect and noisy predictions. Subplot (a) shows the learning curve during training. Subplot (b) shows the final test performance compared with baselines under different forecasting scenarios.

fore, it is robust to distribution shifts.[2] Moreover, it can be seen that as the prediction quality decreases, the gap between *Aggregate SAC PF* and *LP-V2G* quickly narrows. When the noise is very high, $\sigma = 0.04, 0.06$, *Aggregate SAC PF* achieves marginally better performance than *LP-V2G*. This shows that *Aggregate SAC PF* is more robust to imperfect price forecasts than the baselines.

[2]Note the difference in prices from Jan.-Jun. to Jul.-Dec. in Fig. 6.1, and the difference in daily charging sessions between the first and second halves of the year in Fig. 6.2a.

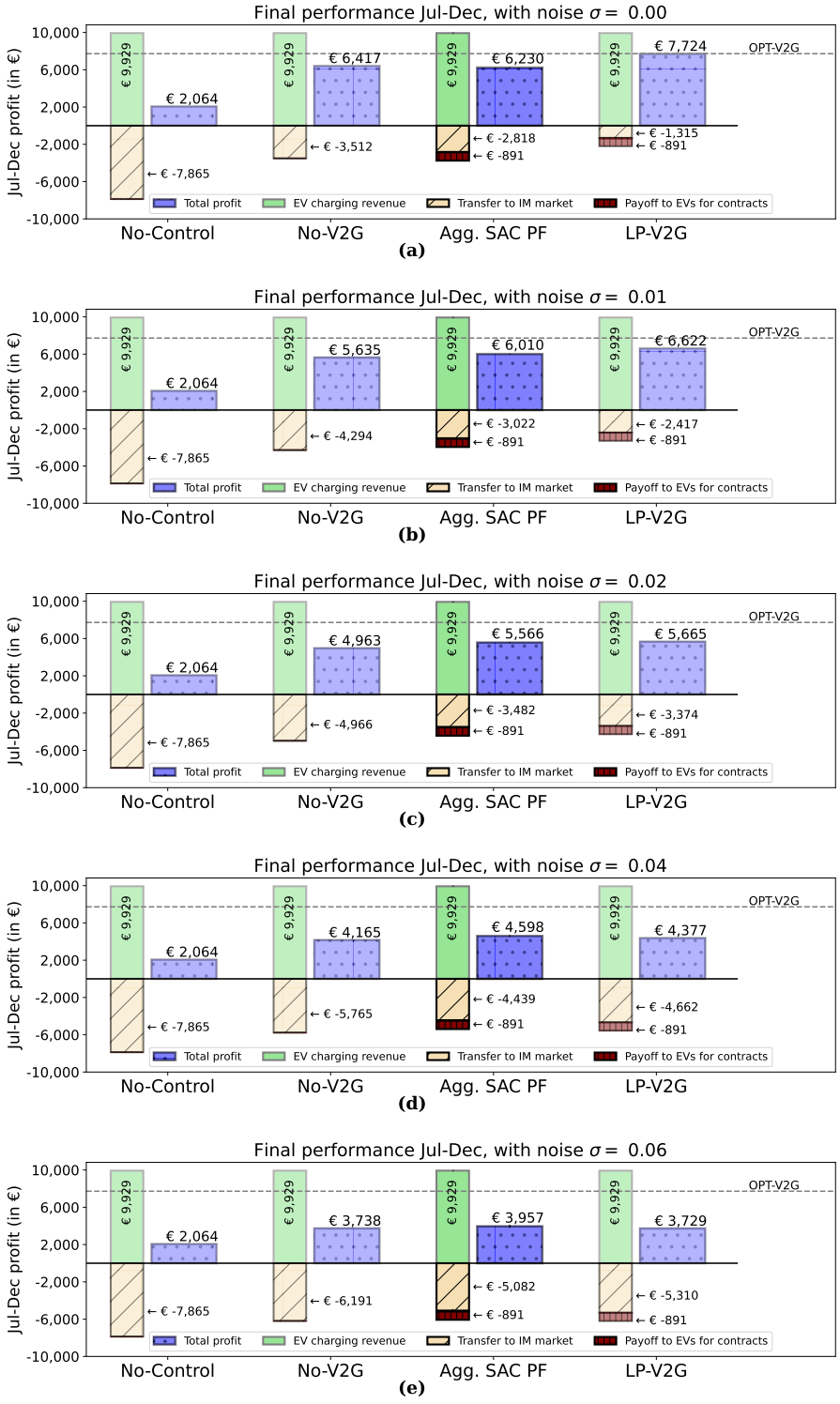

Figure 6.5: Performance for the baselines and top performing *Aggregate SAC PF* on the test dataset (Jul-Dec) with each subplot being a different noise level. Our analysis focuses on subplot **(b)**.

6.4 Profitability

The total VPP's profit is calculated by subtracting the transfer to the imbalance market and payoffs for the accepted V2G contracts from the revenue received from the EVs for supplying their charging demand. We assume EV owners pay a fixed retail price, p^R, for the energy delivered to them, which is $B^n(soc^n_{dep} - soc^n_{arr})$. The rate is $p^R = 0.064$ €/kWh which corresponds to the 85^{th} percentile of the imbalance price over the year.

In Fig. 6.5, we compare the profit made by our top performing *Aggregate SAC PF* on the test dataset, under imperfect price forecasts at every noise level. In this analysis, we will focus on the scenario with the smallest noise level, i.e. σ=0.01, corresponding to subplot **(b)**. The *No-Control* baseline has the lowest profit due to its inability to control EV charging. Next is the *No-V2G* baseline that achieves a higher profit by reducing the transfer to the market to approximately 45% of the previous baseline, underscoring the significance of unidirectional flexibility. Further reduction in the transfer to market is achieved by *Aggregate SAC PF*, precisely 29% compared to *No-V2G*, resulting in higher profitability. This suggests that, despite the payoffs given to EVs, V2G contracts lead to increased profitability. By using information about individual EVs and knowledge of the state evolution model, *LP-V2G* attains the highest profit. Again, the performance of *Aggregate SAC PF* is noteworthy, given its model-free approach and the fact that it utilizes aggregate state information instead of information about every individual EV and considers only 1-step ahead flexibility. Finally, the profitability of *OPT-V2G* is shown as a horizontal dotted line.

6.5 Performance of Disaggregation Algorithms

To evaluate the performance of the *Proportionally Fair* disaggregation algorithm, we utilize the five *Aggregate SAC PF* agents that we previously trained. The evaluation on the test dataset with imperfect price predictions, σ=0.01, is repeated once more, but this time, disaggregation is done using the *priority-based* algorithms described in Section 5.4.

Disaggregation Algorithm	Min.	Mean	Max.
Least Laxity First	**2,966**	**3,026**	**3,088**
Proportionally Fair	3,022	3,058	3,180
Most Laxity First	3,311	3,358	3,477

Table 6.2: One-shot performance (transfer to market in €) of different disaggregation algorithms in the imbalance market with noise level σ=0.01.

The results are presented in Table 6.2. Among these, the *Least Laxity First* is the most effective one in terms of the average performance, showcasing a slight improvement over *Proportionally Fair*. On the other hand, *Most Laxity First* performs the least favorably, accumulating around 9.8% higher cost than *Proportionally Fair*. These findings showcase how the choice of the disaggregation algorithm affects performance, as it decides which specific EVs receive energy, thereby changing the flexibility available in future time steps to the RL agent. Nonetheless, it can be argued that the fairness property offered by *Proportionally Fair* disaggregation is advantageous, considering its performance is on a par with *Least Laxity First*, which does not take fairness into account.

6.6 Computation Time

To verify the claim that our *Aggregate SAC* agent, once trained, is more computationally efficient than solving an optimization problem, we timed the agents on a single test episode. The results are shown in Fig. 6.6, averaged over 5 runs, with error bars indicating the range (max-min) of execution time.

The fastest scheduler is *No-Control* with a time of 0.85 minutes, as it does not make decisions, consistently charging at maximum power until reaching the desired State of Charge (SOC). Following is *Aggregate SAC*. When utilizing priority-based allocation, particularly Least Laxity First, the agent averages 2.31 minutes, represented as *Agg. SAC LL* in the figure. Conversely, with Proportionally Fair Allocation, it averages 2.91 minutes, denoted as *Agg. SAC PF*. This disparity arises because Proportionally Fair Allocation requires solving a

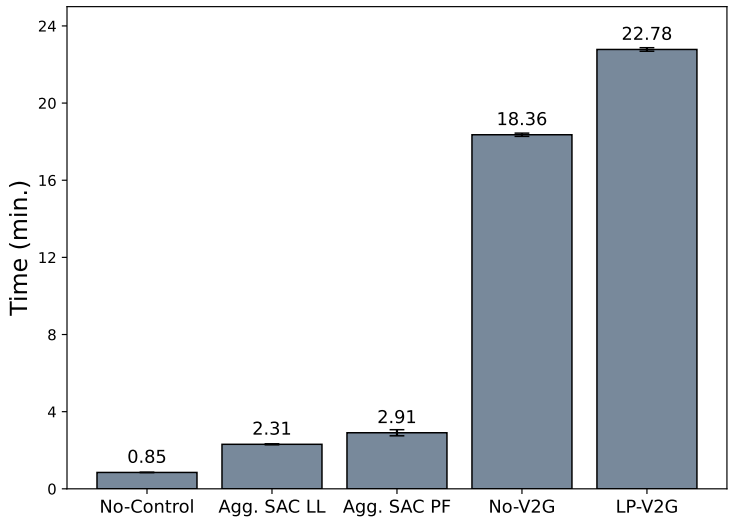

Figure 6.6: Time (minutes) for 1 test episode for the different agents. Each bar is the average of 5 runs and the error bars show (max-min) time.

convex optimization problem. However, this discrepancy is minor when compared to baselines that solve an optimization problem for the (dis)charging action: *No-V2G* and *LP-V2G*. Their running times are 18.36 and 22.78 minutes, respectively. The longer time for *LP-V2G* is attributed to additional constraints associated with V2G contracts. Thus, it is evident that *Aggregate SAC* agents can generate a charging schedule significantly faster than the *No-V2G* and *LP-V2G* baselines.

6.7 Summary

In this chapter, we evaluated our methods using two real-world datasets containing charging sessions and electricity imbalance prices from the Netherlands in 2019. In the first experiment, we investigated the acceptance of different V2G contracts. Next, we assessed the training of *Aggregate SAC*, and tested its performance on new data. The agent was compared against three baselines: a solution that does not control the charging process (*No-Control*), and two based on Linear Programming (*No-V2G*, *LP-V2G*). *Aggregate SAC* demonstrated robust performance in environments with noisy price predictions.

We then analyzed the detailed profitability of the agent and the baselines

across different scenarios. Subsequently, we evaluated the performance of various disaggregation algorithms on the pre-trained model. Finally, we compared the running time of *Aggregate SAC* and the baseline methods, finding that our method could generate feasible charging schedules faster than the Linear Programming methods.

Chapter 7

Conclusion

We proposed a real-time scheduling algorithm for a VPP that trades aggregate flexibility in a real-time electricity market with hourly resolution. The VPP procures V2G participation by offering contracts to EV owners. This increases its profitability despite offering V2G incentives. At the core of our scheduling algorithm is a model-free RL agent capable of making decisions on behalf of all EVs that are controlled by the VPP and trading their flexibility. This is accomplished by aggregating them into a virtual battery with specific constraints. This aggregation is performed in a way that ensures the aggregate action results in a feasible schedule for all EVs. The aggregate action is then broken into individual actions using a proportionally fair disaggregation scheme. Our evaluation shows remarkable performance of this real-time scheduling algorithm, especially when it receives noisy forecasts for future hours.

In this thesis, we have made five key contributions:

- We devised variable-term incentive-compatible V2G contracts that motivate private EV owners to allow the VPP to control their charging. These contracts consider two-dimensional EV owner types in order to cater to a wider range of users.

- We proposed a scalable and efficient reinforcement learning agent that takes in an aggregate abstraction of the flexibility present in the connected EVs, and outputs an aggregate action that is guaranteed to be

feasible at every timestep. The aggregate constraints are calculated by adding the upper and lower charging bounds of each individual EV.

- We constructed a disaggregation algorithm that draws on a popular resource allocation scheme to provide individual charging decisions that have *proportional fairness*. We compared this disaggregation method to others that use a priority-based disaggregation algorithm.

- We assessed the performance of our real-time scheduling approach using real data from public charging stations and prices in the imbalance market.

- We developed a charging simulation environment that uses some of the conventions of Farama's Gymnasium Library to test our agents.

7.1 Limitations and Discussion

In our work, we have made some simplifying assumptions. Firstly, we have not taken into account any grid constraints. This means that the VPP operates without knowledge of the underlying electricity distribution infrastructure. For instance, if there were an overloaded transformer near an EV charging facility, the VPP would not factor this into its decision-making process. While these grid constraints could be incorporated into the methods presented here, they are currently beyond the scope of this work.

Secondly, we assume that users always depart at the time they declare upon arrival. While we incentivize EV owners to truthfully reveal their preferences, in reality, uncertainties may arise, leading to unexpected departures either earlier or later than specified. However, such departures are unlikely to significantly impact the VPP's performance. Early departures automatically result in penalties as the desired SOC may not be reached, while delayed departures simply indicate underutilized flexibility by the VPP.

Additionally, the EV charging dynamics described in Chapter 4 allow for various max/min (dis)charging rates, desired SOC at departure and battery capacities. However, for simplicity in our evaluation, we opted for uniform

values for these variables across all cases. Furthermore, we standardized the timescale to one hour, whereas different real-time markets might have different time scales. To accommodate these (e.g. 15 minutes), the equations can be easily adapted by introducing the time scale, Δt, where necessary. Moreover, while less straightforward, one can extend the equations to allow the VPP to operate at a shorter timescale than the market. In this scenario, the VPP could make decisions more frequently than the price updates, potentially being more responsive as cars arrive and depart.

7.2 Future Work

The work presented in this thesis can be expanded in several directions. Firstly, the parameters for finding optimal V2G contracts, as outlined in Problem (3.5), could be dynamically adjusted based on fluctuations in electricity market prices. This adjustment intuitively aligns with market dynamics; in a stable price environment, the need for V2G would be minimal, whereas in volatile markets, V2G procurement becomes crucial for the VPP. Moreover, insights from financial products, whose value is often tied to volatility, could inform pricing strategies for V2G contracts while keeping their desired properties of *individual rationality* and *incentive compatibility*.

The V2G contracts can also be supplemented by providing additional incentive mechanisms tailored to different types of flexibility. For instance, offering No-V2G contracts for EVs that exclusively wish to provide unidirectional charging flexibility. Furthermore, this strategy could be integrated with participation in different markets, such as the day-ahead market. This could entail committing to energy transactions one day in advance based on predictions for EV energy demand and available V2G discharge energy while modifying the operation in the real-time market to take these commitments into account.

To address uncertainties stemming from electricity price predictions in the real-time market, we plan to compare our result with the solution found via robust optimization or risk-aware reinforcement learning as they are expected to better handle uncertainties arising from market prices and EV traffic patterns.

There are several emerging methods for risk-aware reinforcement learning, a suitable candidate for this would be the *Cross-Entropy Soft-Risk* method presented by Greenberg et. al. in [36].

Moreover, we intend to conduct tests on datasets featuring a higher number of charging sessions per day (10x-100x) to explore the scalability of the proposed algorithm and measure any potential performance degradation. Synthetic data generation methods, such as those outlined in [49], could facilitate the creation of additional charging sessions.

Expanding the heterogeneity of flexibility resources available to the VPP, such as incorporating stationary batteries or renewable energy sources like solar or wind, would enable more strategic energy management decisions. For example, when to use the generated energy to charge the battery or the EVs instead of selling it to the market. Moreover, incorporating grid constraints could help us understand the effect of the VPP on grid congestion. That could help us create new schemes for its operation, and further investigate where to place charging stations within the distribution grid.

Finally, making the in-house EV charging environment fully compatible with Gymnasium [88] could facilitate broader adoption by other research groups. This would enable easier benchmarking against clear and accessible baselines. Later on, this might lead to the establishment of competitions akin to AI Crowd's Citylearn [62], where researchers could submit their own algorithms for VPP-controlled EV charging. There are many substantial areas of opportunity in this field to promote the widespread adoption of VPPs managing bidirectional charging in electrical vehicles in the real world.

References

[1] 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.

[2] *About optimized battery charging on your iphone*, Jan. 2024. [Online]. Available: `https://support.apple.com/en-ca/108055`.

[3] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*, PMLR, 2017, pp. 22–31.

[4] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," *Advances in neural information processing systems*, vol. 32, 2019.

[5] A. Al Zishan, M. M. Haji, and O. Ardakanian, "Adaptive control of plug-in electric vehicle charging with reinforcement learning," in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, ser. e-Energy '20, Virtual Event, Australia: ACM, 2020, pp. 116–120.

[6] B. Alshehhi, A. Karapetyan, K. Elbassioni, S. C.-K. Chau, and M. Khonji, "Dclevernet: Deep combinatorial learning for efficient ev charging scheduling in large-scale networked facilities," in *Proceedings of the 14th ACM International Conference on Future Energy Systems*, 2023, pp. 287–298.

[7] M. B. Anwar, M. Muratori, P. Jadun, *et al.*, "Assessing the value of electric vehicle managed charging: A review of methodologies and results," *Energy & Environmental Science*, vol. 15, no. 2, pp. 466–498, 2022.

[8] M. ApS, "Mosek optimizer api for python," *Version*, vol. 9, no. 17, pp. 6–4, 2022.

[9] D. Avery, *Bidirectional charging and evs: How does it work and which cars have it?* Sep. 2023. [Online]. Available: `https://www.cnet.com/roadshow/news/bidirectional-charging-and-evs-how-does-it-work-and-which-cars-have-it/`.

[10] B. Bartholomew, *Coming soon: California duck curve, how low can you go?* Apr. 2023. [Online]. Available: https://themeritorder.substack.com/p/coming-soon-california-duck-curve.

[11] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[12] *Bidirectional EV chargers to finally materialize in 2024*. [Online]. Available: https://www.solarpowerworldonline.com/2024/01/bidirectional-ev-chargers-to-finally-materialize-in-2024/.

[13] P. Bolton and M. Dewatripont, *Contract theory*. MIT press, 2004.

[14] P. Bovornkeeratiroj, J. Wamburu, D. Irwin, and P. Shenoy, "VPeak: Exploiting volunteer energy resources for flexible peak shaving," in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '21, Coimbra, Portugal: ACM, 2021, pp. 121–130.

[15] M. Brosowsky, F. Keck, O. Dünkel, and M. Zöllner, "Sample-specific output constraints for neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 6812–6821.

[16] Y. Cao, Q. Shi, X. Wang, X. Tian, and Y. Cheng, "Two-dimensional contract theory in cognitive radio networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2012, pp. 1156–1161.

[17] P. Chadraba, *Polestar initiates v2g projects and develops virtual power plant to support large-scale energy transition*, Nov. 2023. [Online]. Available: https://media.polestar.com/global/en/media/pressreleases/675426/polestar-initiates-v2g-projects-and-develops-virtual-power-plant-to-support-large-scale-energy-trans.

[18] B. Chen, P. L. Donti, K. Baker, J. Z. Kolter, and M. Bergés, "Enforcing policy feasibility constraints through differentiable projection for energy optimization," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 199–210.

[19] *Confronting the duck curve: How to address over-generation of solar energy*, Oct. 2017. [Online]. Available: https://www.energy.gov/eere/articles/confronting-duck-curve-how-address-over-generation-solar-energy.

[20] E. Cording and J. Thakur, "Fleetrl: Realistic reinforcement learning environments for commercial vehicle fleets," *SoftwareX*, vol. 26, p. 101 671, 2024.

[21] N. Crisler, *Toyota expands vehicle-to-grid (v2g) research with san diego gas & electric company collaboration*, Feb. 2024. [Online]. Available: https://pressroom.toyota.com/toyota-expands-vehicle-to-grid-v2g-research-with-san-diego-gas-electric-company-collaboration/.

[22] D. Danner, R. Huwa, and H. De Meer, "Multi-objective flexibility disaggregation to distributed energy management systems," *ACM SIGENERGY Energy Informatics Review*, vol. 2, no. 2, pp. 1–12, 2022.

[23] D. Danner, J. Seidemann, M. Lechl, and H. de Meer, "Flexibility disaggregation under forecast conditions," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 27–38.

[24] M. Dehghan, M. Zadehbagheri, M. J. Kiani, and S. Nejatian, "Virtual power plants planning in the distribution network constrained to system resiliency under extreme weather events," *Energy Reports*, vol. 9, pp. 4243–4256, 2023.

[25] S. Y. Derakhshandeh, A. S. Masoum, S. Deilami, M. A. Masoum, and M. H. Golshan, "Coordination of generation scheduling with pevs charging in industrial microgrids," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3451–3461, 2013.

[26] C. Develder, N. Sadeghianpourhamami, M. Strobbe, and N. Refa, "Quantifying flexibility in ev charging as dr potential: Analysis of two real-world data sets," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2016, pp. 600–605.

[27] S. Diamond *et al.*, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[28] *Elaadnl open data*. [Online]. Available: `https://platform.elaad.io/download-data/`.

[29] *Electricity explained: Energy storage for electricity generation*, Aug. 2023. [Online]. Available: `https://www.eia.gov/energyexplained/electricity/energy-storage-for-electricity-generation.php`.

[30] ENTSO-E, *Explanatory document to all TSOs' proposal to further specify and harmonise imbalance settlement in accordance with Article 52(2) of Commission Regulation (EU) 2017/2195 of 23 November 2017, establishing a guideline on electricity balancing*, 2017. [Online]. Available: `https://eepublicdownloads.entsoe.eu/clean-documents/nc-tasks/EBGL/EBGL_A52.2_181218_ALL%20TSOs%20proposal_ISH_explanatory_document_for%20submission.pdf?Web=0`.

[31] EU, *Decisions of the agency for the cooperation of energy cooperation of energy regulators no 18-2020*, 2020. [Online]. Available: `https://nordicbalancingmodel.net/roadmap-and-projects/single-price-model/`.

[32] N. Fatras, Z. Ma, H. Duan, and B. N. Jørgensen, "A systematic review of electricity market liberalisation and its alignment with industrial consumer participation: A comparison between the nordics and china," *Renewable and Sustainable Energy Reviews*, vol. 167, p. 112 793, 2022.

[33] Y. Gao *et al.*, "A contract-based approach for ancillary services in v2g networks: Optimality and learning," in *2013 Proceedings IEEE INFOCOM*, IEEE, 2013, pp. 1151–1159.

[34] *Gm makes vehicle-to-home bidirectional charging technology available across portfolio of upcoming ultium-based evs*, Aug. 2023. [Online]. Available: https://news.gm.com/newsroom.detail.html/Pages/news/us/en/2023/aug/0808-v2h.html.

[35] L. Goldie-Scot, *A behind the scenes take on lithium-ion battery prices*, Mar. 2019. [Online]. Available: https://about.bnef.com/blog/behind-scenes-take-lithium-ion-battery-prices/.

[36] I. Greenberg, Y. Chow, M. Ghavamzadeh, and S. Mannor, "Efficient risk-averse reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 639–32 652, 2022.

[37] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[38] X. Hao, Y. Chen, H. Wang, H. Wang, Y. Meng, and Q. Gu, "A v2g-oriented reinforcement learning framework and empirical study for heterogeneous electric vehicle charging management," *Sustainable Cities and Society*, vol. 89, p. 104 345, 2023.

[39] A. J. Hawkins, *Ford, honda, and bmw create a new vehicle-to-grid company to help ev owners save money*, Sep. 2023. [Online]. Available: https://www.theverge.com/2023/9/12/23870267/ford-honda-bmw-ev-v2g-company-chargescape.

[40] J. Howarth, *How many electric vehicles are in the united states? (2024)*, Nov. 2023. [Online]. Available: https://explodingtopics.com/blog/electric-vehicles-stats.

[41] B. Huang, A. G. Meijssen, J. A. Annema, and Z. Lukszo, "Are electric vehicle drivers willing to participate in vehicle-to-grid contracts? a context-dependent stated choice experiment," *Energy Policy*, vol. 156, p. 112 410, 2021.

[42] S. Huang, R. F. J. Dossa, C. Ye, *et al.*, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html.

[43] *Hyundai and we drive solar launch energy system of the future in utrecht*, Apr. 2022. [Online]. Available: https://www.hyundai.news/eu/articles/press-releases/hyundai-and-we-drive-solar-launch-energy-system-of-the-future-in-utrecht.html.

[44] *It is harder for new electric grids to balance supply and demand*, Apr. 2023. [Online]. Available: https://www.economist.com/technology-quarterly/2023/04/05/it-is-harder-for-new-electric-grids-to-balance-supply-and-demand.

[45] A. G. Jember *et al.*, "Game and contract theory-based energy transaction management for internet of electric vehicle," *IEEE Access*, vol. 8, pp. 203 478–203 487, 2020.

[46] P. L. Joskow, "Lessons learned from electricity market liberalization," *The Energy Journal*, vol. 29, no. 2_suppl, pp. 9–42, 2008.

[47] G. Karatzinis, C. Korkas, M. Terzopoulos, *et al.*, "Chargym: An ev charging station model for controller benchmarking," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, 2022, pp. 241–252.

[48] F. Kelly, "Charging and rate control for elastic traffic," *European transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.

[49] M. Lahariya, D. F. Benoit, and C. Develder, "Synthetic data generator for electric vehicle charging sessions: Modeling and evaluation using real-world data," *Energies*, vol. 13, no. 16, p. 4211, 2020.

[50] M. Lechl, T. Fürmann, H. de Meer, and A. Weidlich, "A review of models for energy system flexibility requirements and potentials using the new flexblox taxonomy," *Renewable and Sustainable Energy Reviews*, vol. 184, p. 113 570, 2023.

[51] C. Lewis-Beck and M. Lewis-Beck, *Applied regression: An introduction*. Sage publications, 2015, vol. 22.

[52] S. Li, W. Hu, D. Cao, *et al.*, "Electric vehicle charging management based on deep reinforcement learning," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 3, pp. 719–730, 2021.

[53] T. Li, B. Sun, Y. Chen, Z. Ye, S. H. Low, and A. Wierman, "Learning-based predictive control via real-time aggregate flexibility," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 4897–4913, 2021.

[54] F. Lilliu, T. B. Pedersen, and L. Šikšnys, "Capturing battery flexibility in a general and scalable way using the flexoffer model," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, 2021, pp. 64–70.

[55] Y. Liu, M. Tian, Y. Chen, Z. Xiong, C. Leung, and C. Miao, "A contract theory based incentive mechanism for federated learning," in *Federated and Transfer Learning*, Springer, 2022, pp. 117–137.

[56] A. Malakhov and R. V. Vohra, "Single and multi-dimensional optimal auctions: A network approach," Discussion Paper, Tech. Rep., 2004.

[57] H. Man, *Tesla confirms bidirectional charging coming by 2025, but musk says it's "inconvenient"*, Mar. 2023. [Online]. Available: https://www.whichcar.com.au/news/tesla-confirms-bidirectional-charging-coming-by-2025-but-musk-says-its-inconvenient.

[58] S. Martin, N. Mosier, O. Nnorom Jr, *et al.*, "Software defined grid energy storage," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2022, pp. 218–227.

[59] N. Naval and J. M. Yusta, "Virtual power plant models and electricity markets-a review," *Renewable and Sustainable Energy Reviews*, vol. 149, p. 111 393, 2021.

[60] B. Neupane, L. Šikšnys, and T. B. Pedersen, "Generation and evaluation of flex-offers from flexible electrical devices," in *Proceedings of the Eighth International Conference on Future Energy Systems*, 2017, pp. 143–156.

[61] *Nissan works to power v2x bi-directional charging across the globe*, Feb. 2023. [Online]. Available: https://www.nissan-global.com/EN/STORIES/RELEASES/nissan-works-to-power-v2x/.

[62] K. Nweye, Z. Nagy, S. Mohanty, *et al.*, "The citylearn challenge 2022: Overview, results, and lessons learned," *NeurIPS 2022 Competition Track*, pp. 85–103, 2022.

[63] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[64] A. Patterson, S. Neumann, M. White, and A. White, "Empirical design in reinforcement learning," *arXiv preprint arXiv:2304.01315*, 2023.

[65] M. Pertl, F. Carducci, M. Tabone, M. Marinelli, S. Kiliccote, and E. C. Kara, "An equivalent time-variant storage model to harness ev flexibility: Forecast and aggregation," *IEEE transactions on industrial informatics*, vol. 15, no. 4, pp. 1899–1910, 2018.

[66] D. Pudjianto, C. Ramsay, and G. Strbac, "Virtual power plant and system integration of distributed energy resources," *IET Renewable power generation*, vol. 1, no. 1, pp. 10–16, 2007.

[67] 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.

[68] S. Rahman *et al.*, "On efficient operation of a V2G-enabled virtual power plant: When solar power meets bidirectional electric vehicle charging," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '22, ACM, 2022, pp. 119–128.

[69] S. Rahman, "Design and optimal operation of a virtual power plant with bidirectional electric vehicle chargers," 2023.

[70] S. Rahman, J. Sales-Ortiz, and O. Ardakanian, "Making a virtual power plant out of privately owned electric vehicles: From contract design to scheduling," in *Proceedings of the 14th ACM International Conference on Future Energy Systems*, 2023, pp. 459–472.

[71] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, vol. 7, no. 1, p. 2, 2019.

[72] H. Ritchie, "The price of batteries has declined by 97% in the last three decades," *Our World in Data*, 2021, https://ourworldindata.org/battery-price-decline.

[73] H. Saboori, M. Mohammadi, and R. Taghe, "Virtual power plant (vpp), definition, concept, components and types," in *2011 Asia-Pacific power and energy engineering conference*, IEEE, 2011, pp. 1–4.

[74] N. Sadeghianpourhamami, J. Deleu, and C. Develder, "Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 203–214, 2019.

[75] N. Sadeghianpourhamami, N. Refa, M. Strobbe, and C. Develder, "Quantitive analysis of electric vehicle flexibility: A data-driven approach," *International Journal of Electrical Power & Energy Systems*, vol. 95, pp. 451–462, 2018.

[76] B. Salanié, *The economics of contracts: a primer*. MIT press, 2005.

[77] J. Sales-Ortiz and O. Ardakanian, "Efficient trading of aggregate bidirectional ev charging flexibility with reinforcement learning," in *Proceedings of the 15th ACM International Conference on Future Energy Systems, to appear*, 2024.

[78] J. Schlund, M. Pruckner, and R. German, "Flexability-modeling and maximizing the bidirectional flexibility availability of unidirectional charging of large pools of electric vehicles," in *Proceedings of the eleventh ACM international conference on future energy systems*, 2020, pp. 121–132.

[79] M. Schoeck, *Us to deploy 30 gw/111 gwh of grid-scale energy storage by 2025*, Dec. 2022. [Online]. Available: `https://www.pv-magazine.com/2022/12/09/us-to-deploy-30-gw-111-gwh-of-grid-scale-energy-storage-by-2025/`.

[80] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[81] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[82] J. Sensiba, *Toyota to study v2g in texas in 2023*, Jan. 2023. [Online]. Available: https://cleantechnica.com/2023/01/01/toyota-to-study-v2g-in-texas-in-2023/.

[83] J. Shi, Y. Gao, W. Wang, N. Yu, and P. A. Ioannou, "Operating electric vehicle fleet for ride-hailing services with reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4822–4834, 2019.

[84] W. Shi and V. W. Wong, "Real-time vehicle-to-grid control algorithm under price uncertainty," in *2011 IEEE international conference on smart grid communications (SmartGridComm)*, IEEE, 2011, pp. 261–266.

[85] L. Šikšnys, T. B. Pedersen, M. Aftab, and B. Neupane, "Flexibility modeling, management, and trading in bottom-up cellular energy systems," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 170–180.

[86] L. Šikšnys, E. Valsomatzis, K. Hose, and T. B. Pedersen, "Aggregating and disaggregating flexibility objects," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 2893–2906, 2015.

[87] *Tennet export data*. [Online]. Available: https://www.tennet.org/english/operational_management/export_data.aspx.

[88] M. Towers, J. K. Terry, A. Kwiatkowski, *et al.*, *Gymnasium*, Mar. 2023. DOI: 10.5281/zenodo.8127026. [Online]. Available: https://zenodo.org/record/8127025 (visited on 07/08/2023).

[89] *U.S. electricity customers averaged five and one-half hours of power interruptions in 2022*. [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=61303.

[90] E. Valsomatzis, T. B. Pedersen, and A. Abelló, "Day-ahead trading of aggregated energy flexibility," in *Proceedings of the Ninth International Conference on Future Energy Systems*, ser. e-Energy '18, Karlsruhe, Germany: Association for Computing Machinery, 2018, pp. 134–138, ISBN: 9781450357678. DOI: 10.1145/3208903.3208936. [Online]. Available: https://doi.org/10.1145/3208903.3208936.

[91] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic ev fleet charging in a day-ahead electricity market," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1795–1805, 2015.

[92] S. Vandael, B. Claessens, M. Hommelberg, T. Holvoet, and G. Deconinck, "A scalable three-step approach for demand side management of plug-in hybrid vehicles," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 720–728, 2012.

[93] M. Vasirani *et al.*, "An agent-based approach to virtual power plants of wind power generators and electric vehicles," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1314–1322, 2013.

[94] E. Verdolini, F. Vona, and D. Popp, "Bridging the gap: Do fast-reacting fossil technologies facilitate renewable energy diffusion?" *Energy Policy*, vol. 116, pp. 242–256, 2018.

[95] *Volvo cars launches new energy solutions business, embracing wider climate potential of electric cars*, Nov. 2023. [Online]. Available: `https://www.media.volvocars.com/global/en-gb/media/pressreleases/318585/volvo-cars-launches-new-energy-solutions-business-embracing-wider-climate-potential-of-electric-cars`.

[96] J. Wang, C. Guo, C. Yu, and Y. Liang, "Virtual power plant containing electric vehicles scheduling strategies based on deep reinforcement learning," *Electric power systems research*, vol. 205, p. 107 714, 2022.

[97] Y. Wang, D. Qiu, and G. Strbac, "Multi-agent deep reinforcement learning for resilience-driven routing and scheduling of mobile energy storage systems," *Applied Energy*, vol. 310, p. 118 575, 2022.

[98] S. White, *V2g: My nissan leaf earned $100 in just two hours supporting grid in heatwave*, Mar. 2024. [Online]. Available: `https://thedriven.io/2024/03/14/v2g-my-nissan-leaf-earned-100-in-just-two-hours-supporting-grid-in-heatwave/`.

[99] C. Wu *et al.*, "Vehicle-to-aggregator interaction game," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 434–442, 2011.

[100] Z. Xiong, J. Kang, D. Niyato, P. Wang, H. V. Poor, and S. Xie, "A multi-dimensional contract approach for data rewarding in mobile networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5779–5793, 2020.

[101] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM transactions on networking*, vol. 8, no. 5, pp. 667–678, 2000.

[102] L. Yan, X. Chen, Y. Chen, and J. Wen, "A cooperative charging control strategy for electric vehicles based on multiagent deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8765–8775, 2022.

[103] M. Zeng *et al.*, "An incentivized auction-based group-selling approach for demand response management in v2g systems," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1554–1563, 2015.

Appendix A

Implementation Details

In this appendix we will go over the main parts of the code, the most important parameters for training, the architecture of the neural networks, and the commands used to run our program. Recall that we base our implementation of the agent on CleanRL's code of continuous-action SAC [42] for PyTorch [63].

A.1 Code

In Listing A.1 we show an example of how we train our implementation of *Aggregate SAC*. First, we import some general modules. Then we import the user-defined modules, mainly the environment (`ChargeWorldEnv`), the actor (`agentSAC_sagg`), and the critic (`SoftQNetwork`).

In the body of the program, we initialize `ChargeWorldEnv` with the dataset that contains the charging sessions `(df_sessions)`, the dataset that contains the real-time prices (`df_prices`), the contract parameters (`contract_info`), and a random number generator (`rng`).

For the agent, we initialize the actor (`agentSAC_agg`) with the price dataset (`df_price`), arguments read from the command line (`args`), and the device (`device`). This `device` is needed for certain PyTorch functionalities. The critic is composed of two Q networks (`SoftQNetwork`). Additionally, soft actor-critic uses a replay buffer (`rb`).

We can train the agent for many `episodes`, each with a predetermined number of `timesteps`. Similar to Farama's Gym, the environment is initialized with a `world.reset()`. During training, the agent receives an observation

from the environment, and it outputs an action with `agent.get_action()`. The environment receives the action and moves forward one timestep with `world.step()`. The loop keeps going on until all the timesteps are completed for all the episodes. At each iteration, the training of the agent is performed.

The charging sessions dataset, real-time prices dataset and state are implemented in Pandas DataFrames. Additionally, the environment also receives a Pandas DataFrame for the action. Conversely, the agent works mainly with PyTorch Tensors. To convert the state DataFrame into a PyTorch Tensor, we employ `agent.df_to_state()`. Similarly, to convert the agent's action into the required Pandas format that the environment prefers, we use `agent.action_to_env()`.

```
# Import modules
import torch
import pandas as pd
import numpy as np
...

# Import user-defined modules
from EvGym.charge_world import ChargeWorldEnv
from EvGym.charge_sac_agent import agentSAC_sagg, SoftQNetwork


def main():
    ...
    # Initialize environment
    world = ChargeWorldEnv(df_sessions,
                           df_price,
                           contract_info,
                           rng)

    ...

    # Initialize agent
    agent = agentSAC_sagg(df_price, args, device).to(device)
    qf1 = SoftQNetwork(args).to(device)
    qf2 = SoftQNetwork(args).to(device)
    rb = ReplayBuffer(
                args.buffer_size,
                ... )
    ...

    # Training
    for episode in range(episodes):
        df_state = world.reset()
        obs = agent.df_to_state(df_state, ts_min)
        ...
        for t in range(timesteps):
            ...
            actions, _, _ =
            agent.get_action(torch.Tensor(obs).to(device))
            ...
            # Take a step, it receives the action of the agent
            df_state, rewards, terminations, infos =
            world.step(agent.action_to_env(actions))
            next_obs = agent.df_to_state(df_state, t)
            ...
            obs = next_obs
            ...
```

Listing A.1: Some examples of the implementation of the environment and the agent

A.2 Parameters

The main implementation parameters are shown in Table A.1.

Parameter	Value	Description
`--agent`	SAC-sagg	Agent to use for real-time scheduling
`--save-name`	sac_a	Name used for logs, results, etc.
`--pred-noise`	0.00	Nosie for price predictions in training
`--seed`	42	Seed for random number generators
`--years`	200	Number of episodes to train
`--batch-size`	512	Batch size to sample from the replay buffer
`-alpha`	0.02	Temperature parameter in SAC
`--policy-frequency`	4	How often to update the policy (timesteps)
`--target-network-frequency`	2	How often to update the second Q NN (timesteps)
`--disagg`	PF	(Proportional fairness) Disaggregation algorithm
`--buffer-size`	1e6	Number of experiences to save in replay buffer
`--save-agent`	True	Save the weights of the trained agent
`--general`	True	Run training (`False` is for deployment)

Table A.1: The main implementation parameters.

A.3 Architecture

The architecture for the actor, the policy network, is shown in Table A.2. The architecture for the two critics, soft Q networks, is shown in Table A.3.

Layer	In	Out
Linear (ReLU)	59	256
Linear (ReLU)	256	256
Head 1, Mean: Linear (Sigmoid)	256	1
Head 2, Logstd: Linear (Tanh)	256	1

Table A.2: Architecture for the Actor.

Layer	In	Out
Linear (ReLU)	60	256
Linear (ReLU)	256	256
Linear	256	1

Table A.3: Architecture for the Critics.

A.4 Running the Scripts

The command to run the experiment with the parameters in Table A.1 is shown in Listing A.2.

```
python3 RunSACChargeWorld.py --agent SAC-sagg --save-name sac_a \
                             --pred-noise 0.00 --seed 42 --years 200 \
                             --batch-size 512 --alpha 0.02 \
                             --policy-frequency 4  \
                             --target-network-frequency 2 \
                             --disagg PF --save-agent True \
                             --general True
```

Listing A.2: Run a training experiment.