Jose Adan Rivera Acevedo

Distributed Optimization for Electric Vehicle Charging Control

Technische
Universität
München

Technische Universität München

Fakultät für Informatik

Distributed Optimization for Electric Vehicle Charging Control

Jose Adan Rivera Acevedo

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende(r): Prof. Dr. Matthias Althoff

Prüfer der Dissertation:

1. Prof. Dr. Hans-Arno Jacobsen
2. Prof. Dr. Thorsten Staake

Die Dissertation wurde am 12.10.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 21.11.2017 angenommen.

Abstract

The number of Electric Vehicles (EVs) will increase dramatically in the coming years; as a result, EV charging control will be required to fully realize EVs potential to reduce CO2 emissions and support their integration into the current power infrastructure. In this work, we use the formal mathematical method of distributed optimization to address two key challenges in the area of EV charging control.

First, we consider the scalability problem faced by EV aggregators to schedule the charging of a large number of EVs and provide system services. We propose a new optimization framework for achieving computational scalability based on the Alternating Direction Method of Multipliers (ADMM). It allows for distributing the optimization process across several servers/cores. We demonstrate the performance and versatility of our framework by applying it to two relevant aggregator objectives: valley filling; and cost-minimal charging with grid capacity constraints. We evaluate the scalability of our framework in numerical experiments using realistic input data. Our results show that the solving time of our approach scales linearly with the number of controlled EVs and outperforms the centralized optimization benchmark as the fleet size becomes larger. Furthermore, to aid with the practical realization of an EV aggregator control infrastructure, we study the implementation of EV ADMM on top of a pub/sub middleware. Pub/sub middleware is based on a communication paradigm, where subscribing entities register their interest in events and are subsequently asynchronously notified of events that are issued by publishing entities. This offers a lot of flexibility for the development of distributed applications. The results of our testbed simulation experiments confirm that an EV aggregator decentralized control infrastructure can benefit from using a pub/sub middleware to reduce its communication complexity.

Second, we consider the real-time EV charging congestion control problem to allow the safe integration of large EV numbers into the current distribution infrastructure. We propose a novel distributed anytime algorithm for real-time congestion control of EV charging, which maximizes the EV chargers' utility of the grid without violating the network's operational constraints. The algorithm results from the primal decomposition solution of the EV Network Utility Maximization (NUM) problem. We provide the formulation of our solution approach, as well as the state-of-the-art dual decomposition solution. Both algorithms are then evaluated using numerical simulations under static and dynamic conditions. Our results demonstrate that the proposed primal algorithm remains stable and retains its anytime property under dynamic conditions. Compared to the state-of-the-art solution, the proposed algorithm offers improved scalability and reliability, allowing for the provision of EV charging congestion control in milliseconds instead of minutes.

Zusammenfassung

Die Anzahl an Elektrofahrzeugen (englisch: electric vehicles, EVs) wird in den kommenden Jahren dramatisch steigen. Damit einhergehend wird eine Ladesteuerung für Elektrofahrzeuge benötigt, welche es ermöglicht das CO2-Reduktionspotential von Elektrofahrzeugen auszuschöpfen und eine große Anzahl an Elektrofahrzeugen ins bestehende Stromnetz zu integrierten. In dieser Arbeit wird die Methode der Verteilten Optimierung herangezogen, um zwei zentrale Herausforderungen im Bereich Ladesteuerung für Elektrofahrzeuge zu adressieren.

Im ersten Schritt betrachten wir das Skalierbarkeitsproblem von Elektrofahrzeug-Aggregatoren bei der zeitlichen Ladeplanung für eine große Anzahl an Elektrofahrzeugen und bei der Erbringung von Systemdienstleistungen. Wir schlagen ein neues Optimierungs-Framework vor, um Berechnungsskalierbarkeit auf Basis der Methode Alternating Directions Method of Multipliers (ADMM) zu erzielen. Diese erlaubt es, den Optimierungsprozess auf verschiedene Server/Prozessorkerne zu verteilen. Wir zeigen die Leistungsfähigkeit und Vielseitigkeit unseres Ansatzes durch dessen Anwendung bei zwei relevanten Aggregator-Zielen: Verbrauchsspitzenverschiebung (engl. valley filling) und kostenminimale Aufladung unter Berücksichtigung von Kapazitätsbeschränkungen im Netz. Zudem evaluieren wir die Skalierbarkeit unseres Frameworks in umfassenden numerischen Experimenten und auf Basis realistischer Eingangsdaten. Unsere Ergebnisse zeigen, dass die Optimierungsszeit unseres Ansatzes linear mit der Anzahl der gesteuerten Elektrofahrzeuge steigt und bei höheren Elektrofahrzeugzahlen schneller Ergebnisse liefert als die herkömmliche Methode der zentralen Optimierung. Um darüber hinaus einen Beitrag zur praktischen Realisierung einer Elektrofahrzeug-Aggregator-Steuerungsinfrastruktur zu unterstützen, untersuchen wir die Implementierung von EV ADMM auf Basis von Pub/Sub Middleware. Pub/Sub Middleware basiert auf dem Kommunikationsparadigma, dass zeichnende Einheiten Interesse an Events bekunden und danach asynchron über Events, die von veröffentlichenden Einheiten herausgegeben werden, informiert werden. Dies bietet eine hohe Flexibilität für die Entwicklung verteilter Anwendungen. Die Ergebnisse aus unseren Testumgebungssimulationen bestätigen, dass eine dezentrale Elektrofahrzeug-Aggregator-Steuerungsinfrastruktur von einer Pub/Sub Middleware profitiert und damit ihre Kommunikationskomplexität reduzieren kann.

Im zweiten Schritt betrachten wir das Problem der Echtzeit-Überlastregelung beim Laden von Elektrofahrzeugen. Damit leisten wir einen Beitrag zur sicheren Integration einer großen Anzahl von Elektrofahrzeugen in die derzeitige Verteilnetzinfrastruktur. Wir schlagen einen neuen verteilten Anytime Algorithmus für die Echtzeit-Überlastregelung beim Laden von Elektrofahrzeugen vor. Dieser maximiert den Nutzen von Elektrofahrzeugladegeräten am Netz ohne die operativen Bedingungen für den Netzbetrieb zu verletzen. Der Algorithmus ergibt sich aus der Primal Decomposition Lösung des EV Network Utility Maximization (NUM) Problems. Die Formulierung unseres Lösungsansatzes als auch die des State of the Art Ansatzes auf Basis der Methode Dual Decomposition wird im Rahmen der Arbeit dargestellt. Beide Algorithmen werden im Anschluss mittels numerischen Simulationen unter statischen und dynamischen Bedingungen evaluiert. Es zeigt sich, dass der von uns vorgeschlagene Algorithmus verglichen mit der State of the Art Lösung eine verbesserte Skalierbarkeit und Zuverlässigkeit aufweist und die Überlastregelung beim Laden von Elektrofahrzeugen in Millisekunden anstelle von Minuten ermöglicht.

Acknowledgments

This doctoral dissertation is the result of work carried out at the Department of Computer Science of the Technische Uniersität München at the Chair for Application and Middleware Systems under the supervision of Prof. Dr. Hans-Arno Jacobsen.

First and foremost I would like to thank Prof. Dr. Hans-Arno Jacobsen for his continuous guidance and support. Thank you for financing me from your Alexander von Humboldt Professorship funding and giving me the freedom to pursue my research goals. You have been a demanding trainer that pushed me to achieve greater results than I ever expected of myself. It has been a privilege to work under your supervision. I would also like to thank Prof. Dr. Thorsten Staake for agreeing to be my second advisor and providing excellent feedback to improve this thesis. My thanks also extend to Prof. Dr. Matthias Althoff for having agreed to be the session chair of my dissertation committee.
I would also like to thank Dr. Christoph Goebel for all the discussions about my work and his contributions. A special thanks go to Martin Jergler and Aleksandar Stoimenov for their help with the implementation of DOPS. Martin, you are the best friend and office mate one can ever imagine. Also, I would like to thank Christoph Doblander for all the discussions and help during this time.

During my time as a Ph.D. candidate, I also had the opportunity to develop the OpenGridMap project as part of the Software Campus program. I would like to thank the Bundesministerium für Bildung und Forschung (BMBF) for financing this program. My thanks also extend to all students who worked on the project and my industry advisor Dr. Michael Metzger from Siemens AG. A special thanks to Klaus Schreiber, Shota Bakuradze, Tanuj Ghinaiya, Johannes Leimhofer, Qunjie Zhou and Pezhman Nasirifard for all your contributions.

I would like to thank my parents Maria Esperanza Acevedo Gutierrez and Jose Adan Rivera Castillo for always believing in me. I have made it this far only because of you. Last but not least I would like to thank Maria Fischl. Your love and support helped me overcome some of the greatest challenges I have ever faced. I love you.

Acronyms and Abbreviations

EV	Electric Vehicle
ADMM	Alternating Direction Method of Multipliers
NUM	Network Utility Maximization
DOPS	Distributed Optimization Publish and Subscribe
Pub/sub	Publish and subscribe
IEEE	Institute of Electrical and Electronics Engineers
ICT	Information and Communications Technology
GHG	Greenhouse Gases
AIMD	Additive-Increase/Multiplicative-Decrease
V2G	Vehicle-to-Grid
DoD	Depth of Discharge
EEX	European Energy Exchange
NHTS	National Household Travel Survey
KKT	Karush Kuhn Tucker
VLAN	Virtual Local Area Network

Contents

Chapter 1

Introduction

The number of Electric Vehicles[1] (EVs) will increase dramatically in the coming years [1]. In large numbers, EVs have the potential to significantly contribute to the reduction of CO2 emissions and help with the integration of renewable energy resources, making them a relevant component in the mitigation of climate change [2]. However, to fully realize the climate change-related benefits of EVs and to support their integration into the current power infrastructure, the charging of a vast number of EVs will have to be controlled. Thanks to raising Information and Communication Technology (ICT) investments in the power infrastructure, the capacity to control EV charging will soon be available. Nevertheless, the design of optimal EV charging control algorithms that scale to the large number of expected EVs and can cope the highly dynamic changes of the power grid remains a challenge.

In this work, we use the formal mathematical method of distributed optimization to address key challenges in the area of EV charging control. First, we look at the issue of computational scalability for solving large-scale EV aggregator optimization problems. Second, we consider the congestion control problem that emerges when large EV numbers integrate into the power distribution infrastructure. For both aspects, we leverage distributed optimization to formulate novel solution approaches that

[1] In this work the term Electric Vehicle (EV) refers specifically to Plug-in Electric Vehicle (PEV), which use electricity stored in batteries to drive the wheels and can be recharged by an external electricity source.

contribute towards the practical realization of an EV charging control infrastructure.

1.1 Motivation

The introduction of a vast number of EVs in the current electric power system is a double-edged sword. On the one hand, the load increase caused by large EV numbers can have adverse effects on the distribution network, leading to new and expensive energy peaks, overloading of lines and transformers, and a reduction of power quality [3–6]. On the other hand, large numbers of EVs can be a valuable asset to the system, because their flexibility and storage capabilities can be leveraged to support system operators with the integration of renewable energy sources and the reduction of GHG emissions [7–9]. To avoid the potential drawbacks of massive EV integration and to allow system operators to utilize EVs as a resource to optimize their operations, EV charging control is required. One major challenge for the realization of an EV charging control infrastructure is the formulation of scalable, reliable and efficient control algorithms.

Traditionally the control of power system devices operations is done in a centralized manner, i.e., centralized optimization for open-loop control [10]. Examples of classical problems for generators include unit commitment, economic dispatch, and optimal power flow. However, the introduction of thousands or even millions of EVs as controllable devices would make a centralized optimization impracticable. Several solutions are being proposed to deal with this scalability issue. One solution is to use less computationally intensive heuristic approaches to solve the centralized optimization problem. Another solution is the definition of a hierarchical control structure that separates the problem into smaller chunks to be controlled independently of each other. While both these methods address the scalability problem, they do so by sacrificing optimality. That means these methods do not guarantee an optimal result [11, 12], which translates into an underutilization of resources, higher costs or a loss of revenue. Also, their performance is coupled with the parameters of the problem [13]. Hence, extensive testing is required to determine their performance, but no consistent theoretical performance can be guaranteed when

the parameters change. Nevertheless, if some level of suboptimality and performance variation can be tolerated, these approaches result in efficient algorithms that are often better suited for practicable implementation [11, 12]. A formal alternative solution to address the scalability problem is the use of distributed optimization. Distributed optimization solves an optimization problem by decomposing it into easier to solve subproblems and a master problem that through an iterative procedure closes in on the optimal solution of the original problem. The advantage of distributed optimization is that it offers optimality guarantees that are independent of the parameters of the problem. That means that as long as the problem is properly defined the method will deliver the optimal result, regardless of the parameters [14]. What is even more appealing about this approach, is that the algorithm arising out of solving a problem with distributed optimization can be used to formulate a decentralized control protocol. Thus, distributed optimization offers a formal approach for designing optimal decentralized control protocols. Examples of the usage of distributed optimization to develop decentralized control protocols for communication networks can be found in the literature and are currently used in practice, e.g., in network congestion control [15–17].
Inspired by these previous results, we focus our work on the application of distributed optimization to address two important EV charging control problems: EV aggregator scheduling and real-time EV charging congestion control. The challenge of using the distributed optimization approach is that the problems must be formulated under a rigid structure which allows the application of its methods. At the same time, the problem formulation must include all relevant problem features required to offer an implementable EV charging control solution. By following this formal approach, we not only expect to provide a scalable computational solution for our considered problems but also gain some insights that can help with the design of a scalable EV charging control infrastructure.

1.2 Problem statement

The control of EV charging is the key to unlock the EVs' full potential to reduce GHG emissions and support their integration into the current power infrastructure.

However, the large number of expected EVs and the highly dynamic changes of the power grid make the formulation of scalable, efficient and reliable EV charging control algorithms a challenge.

The research objectives of this work are twofold:

1. *Develop a scalable EV charging control approach based on distributed optimization to allow large numbers of EVs to offer energy system services via an EV aggregator and study its implementation in a distributed environment under real-world assumptions.*

2. *Develop a real-time EV charging congestion control approach based on distributed optimization to support the integration of large numbers of EVs into the current power distribution grid.*

Once available in large numbers, EVs could provide substantial, reliable system services with limited effect on their energy charging requirements, i.e., making sure that the EVs will have enough charge to complete their trips [8]. Two key features make EVs a promising resource for providing services. EVs, or cars in general, are parked most of their lifetime. Also, their energy requirements can be predicted [7]. However, to have a significant influence on the power system as a whole, it is necessary to control the behavior of a vast number of EVs.
One promising conceptual framework to allow the provision of system services by large numbers of EVs is via the use of *EV aggregators*. In this framework, an EV aggregator is a third party that acts as an interface between an EV fleet and the rest of the power system (cf. [7, 8]). The advantage of introducing EV aggregators is that they allow the system operator to conceptualize EV groups as a single controllable unit. Thereby, the complexity of managing individual EVs can be significantly reduced. The EV aggregator must strike a balance between the charging requirements of the EVs and the service needs of the system operator. The aggregator's objective for the EV fleet may vary depending on the service requirements. Example goals are the balancing of demand and supply, load peak shaving, the flattening of the demand profile (also known as valley filling), the provision of reserve capacity in the energy market or the use of EVs to store renewable energy. Most of these goals rely

on prediction techniques that estimate supply and demand for a particular future horizon to enable a planning of system operations. This planning horizon is usually one day ahead for 24 hours and is divided into 15 min intervals. During this planning horizon, the objective of each EV is to reach a certain level of charge. Since most EVs are connected to the grid more time than the time required to fulfill their charging requirements, there is a time period, in which the EVs' charging jobs can be flexibly scheduled by the EV aggregator to offer system services. To do this, the aggregator must know the EVs' charging requirements ahead of time. These can either be communicated to the aggregator by the EV, for example, "Tomorrow I must charge 15 kW between 8 pm and 8 am", or they can be predicted by the aggregator using the travel information of each EV. A major challenge to this approach is that the behavior of people is not easy to schedule. Deviations from the schedule are prone to happen. One possible solution to mitigate the effects of schedule deviations is to implement the scheduling in a receding horizon; this means recalculate the schedule periodically as new information comes in, e.g., every 15 min. However, computing optimal schedules at such timescales for a very large EV fleet poses a challenge for the EV aggregator.

To obtain a maximal benefit from the flexibility of its EV fleet, the EV aggregator can schedule the charging of individual EVs by solving a mathematical optimization problem. Several previous studies on this topic [18–23] suggested to address the EV aggregator scheduling problem via a centralized optimization. In centralized optimization, the original problem formulation is solved using a state-of-the-art optimization software. However, the introduction of thousands or even millions of EVs as controllable resources makes a centralized optimization of large EV aggregators computationally intractable [8, 9, 24]. To overcome this scalability problem and still guarantee optimal solutions, recent studies have suggested distributed optimization algorithms for EV charging control [25–30]. Nonetheless, all these studies focus on a particular objective or control task and do not provide a general solution to implement the various EV aggregator's objectives. Moreover, none of the studies demonstrate the scalability of their method as the number of EVs increases or benchmark their distributed optimization approach against a centralized optimization approach. Furthermore, a study of the implementation of these approaches in a distributed environment under real-world assumptions is missing from the literature. Therefore, in this work, our first aim is to provide a *general* and *scalable* framework for the distributed optimization of EV aggregators and to study its implementation in a distributed environment.

While the EV aggregator conceptual framework focuses on the scheduling of EV charging based on predictions to provide services for the power system as a whole, certain problems have localized effects where the control of EVs must be done in real-time. These problems require fast control responses (minute to millisecond range) that a scheduling approach is not designed to deliver. The most prominent of these problems is the control of EV charging to avoid overloading of distribution grid components.
One of the major obstacles to the large-scale deployment of EVs is that the current power distribution infrastructure is unable to handle the load increase. Several studies corroborate that the introduction of EVs, especially with fast chargers, will surpass the maximal capacity of lines and transformers, and cause voltage swings in the distribution system [31–33]. In the worst case scenario, overloading of lines and transformers caused by EVs can even lead to the triggering of distribution grid protection devices and a subsequent blackout.
One solution to this problem is to upgrade the distribution grid to handle the load increase. Upgrading the network, however, comes at an enormous cost and would slow down EV adoption. A more practical solution is the control EV charging. The challenge in this approach is the large number of expected EVs, their unknown spatial distribution and high dynamic changes of the distribution grid. Moreover, since predictions at this time scale and for individual devices are unavailable, a blackout can only be avoided, if we can react within milliseconds, to prevent the triggering of protection devices (usually around 200 ms after an overload). Therefore, a distributed real-time EV charging control is required, that can adapt quickly to the fast dynamics of the grid, allows for local decisions of individual devices, and optimizes the global operation of the whole system.
The available literature offers some distributed approaches for real-time EV charging congestion control [34, 35]. However, these approaches are based on heuristics, which do not guarantee that the EVs will make an optimal use of the available infrastructure under all circumstances. Others [36, 37] propose distributed optimization approaches, but their resulting algorithms cannot cope with the real-time requirements of the problem and may lead to the temporary overloading of transformers and lines.

To address these issues, our second aim in this work is to provide a distributed optimization approach for EV charging congestion control, which prevents overloading of the grid's components and, at the same time, maximizes the use of the available infrastructure.

1.3 Approach

In this work, we leverage distributed optimization to address key challenges in EV charging control. We strive for a general formulation of the problems that support the application of distributed optimization techniques. Distributed optimization is chosen, because it allows for scalable optimal solution algorithms and at the same time provides a formal method to design decentralized control protocols, cf. [15–17]. First, we consider the problem of computational scalability faced by EV Aggregators to solve large-scale EV charging scheduling problems. We address this challenge by using the Alternating Direction Method of Multipliers (ADMM) for distributed optimization. In the rest of this document, we refer to our approach as *EV ADMM*. Second, we consider the problem of real-time congestion control for EVs charging in distribution networks. For this problem, a general Network Utility Maximization (NUM) problem formulation is used. In the rest of this document, we refer to this approach as *EV NUM*. In the following, we will go deeper into the specifics of our proposed EV ADMM and EV NUM approaches.

1.3.1 Distributed optimization for EV Aggregators with EV ADMM

In EV ADMM we propose and evaluate a novel general and scalable framework for the distributed convex optimization of EV aggregators. We introduce a general formulation of the EV aggregator optimization problem that allows for local and global objectives as well as local and global constraints of individual EVs and the EV aggregator, respectively. We solve the general EV aggregator optimization problem using the Alternating Direction Method of Multipliers (ADMM). This method is

chosen, because it can solve any optimization problem, provided the problem is feasible and convex [38]. Based on ADMM, we formulate a versatile and scalable distributed optimization framework (EV ADMM), which allows for distributing the optimization process across several servers/cores. Our evaluations showcase the versatility of our framework to solve various EV Aggregator optimization problems and demonstrate that the solving time of our approach scales linearly with the number of controlled EVs, outperforming the centralized optimization benchmark as the fleet size becomes larger. EV ADMM can, therefore, be used to solve convex EV charging planning and control optimization problems that have so far been considered intractable due to large EV numbers.
To aid with the implementation of algorithms like EV ADMM in a distributed environment, we also propose DOPS (Distributed Optimization Publish and Subscribe). DOPS is a framework for distributed optimization of EV charging, which works on top of a pub/sub middleware. We study the implementation of EV ADMM-like algorithms in DOPS using a testbed consisting of 15 virtual machines connected via a VLAN network and compared three different communication schemes: centralized, decentralized and decentralized with aggregation. Our results show that the in-network aggregation feature of DOPS reduces the runtime of the standard decentralized approach by drastically reducing the number of messages exchanged between the EVs and the aggregator. This work represents the first actual implementation of distributed optimization EV charging control algorithms in a distributed environment and confirms that such decentralized control infrastructures, can benefit from using a pub/sub middleware to reduce their communication complexity.

1.3.2 Distributed optimization for EV charging congestion control with EV NUM

In EV NUM, we propose a novel distributed optimization algorithm for real-time EV charging congestion control. We formulate the problem not as a time horizon scheduling problem, but as an instantaneous Network Utility Maximization (NUM) problem. The goal herein is to maximize the utility of the EV chargers considering their constraints and the constraints of the grid for a particular state of the distribution system. The standard approach for solving NUM problems relies on using dual decomposition and subgradients methods, which through a dual price exchange mechanism yields algorithms that operate on the basis of local information (cf. [15]). This approach is applied to the EV charging congestion problem in [36]. However, the drawback of the dual decomposition approach is that its iteration results are only feasible at optimality, i.e., the control values can only be used on convergence. In real-time EV charging control, a grid overload and the triggering of protection devices can only be avoided, if can react within milliseconds. Thus, EV charging is a time critical application, where the iterative process of the optimization algorithm may need to be stopped before optimality is reached. Furthermore, to cope with the real-time requirements, the produced results on each iteration should be feasible, i.e., the produced control values should be usable at any time. Hence, we need an *anytime algorithm*. Anytime algorithms are defined as algorithms that return some answer for any allocation of computation time and are expected to return better answers when given more time [39].

We propose a distributed anytime algorithm for solving the NUM problem in EV charging. Our approach to solve the NUM problem is based on dynamic budgets or upper bounds for EV chargers. We modify the original EV NUM problem to include budgets and apply primal decomposition. This results in an iterative distributed anytime algorithm that generates the optimal budgets for each EV charger. We define closed-form expressions for computations performed by EV chargers and protection devices. Given the system's current state, the different devices perform computations using only their local information and exchange messages to reach the optimal operation point for the whole system.

We conduct comprehensive evaluations of our proposed primal solution approach in the IEEE 13 Node Test Feeder and the IEEE European Low Voltage Test Feeder. There, we compared its behavior under static and dynamic conditions as well as its scalability and impact on the grid to the state-of-the-art dual decomposition approach. The results demonstrate that our proposed algorithm remains stable and retains its anytime property under dynamic conditions. Moreover, compared to the dual solution, the proposed algorithm offers improved scalability and reliability for EV charging congestion control. Our distributed anytime approach allows the implementation of EV charging congestion control in milliseconds instead of minutes.

This makes the system more robust to fast dynamic changes and provides better support for the integration of EVs into the current distribution grid.

1.4 Contributions

The contributions made in this work affect two main topics: The charging control of large EV numbers via an aggregator and real-time EV charging congestion control. In the following, we discuss our contributions on each topic.

The main contributions to the charging control of large EV numbers via an aggregator are:

i. We propose a novel general framework for the distributed optimization of convex EV aggregator problems, which we call *EV ADMM*.

ii. We demonstrate the flexibility of EV ADMM by solving two relevant EV aggregator optimization problems: Valley filling and charging cost minimization with capacity constraints.

iii. We show the scalability of EV ADMM for EV fleets with up to 1 million EVs and benchmark its runtime and peak memory use against a centralized optimization approach. We show that EV ADMM allows the solution of problems that are intractable for a centralized optimization.

iv. We also propose DOPS, a pub/sub middleware framework for distributed optimization. It provides a concise interface to implement algorithms like EV ADMM in a distributed environment.

v. We show the advantages that DOPS provides to reduce the number of messages for the implementation of EV ADMM in a distributed environment with a comprehensive benchmark of three different implementation approaches: centralized, decentralized, and decentralized with aggregation.

The main contributions for real-time EV charging congestion control are:

i. We propose a novel distributed anytime algorithm for real-time EV charging congestion control: A budged-based control algorithm that results from the distributed optimization solution of the EV NUM problem with primal decomposition.

ii. We analyze its convergence conditions and demonstrate its advantages against the state-of-the-art dual decomposition EV NUM solution.

iii. We provide a coherent exposition of the EV NUM dual and primal decomposition approaches and outline their main features concerning actual deployment.

iv. We provide a comprehensive comparison of both EV NUM solution algorithms under static and dynamic conditions for the IEEE 13 Node Test Feeder and show that our primal algorithm can offer better control performance than the state-of-the-art dual algorithm.

v. We demonstrate that the EV NUM formulation can effectively be used for EV charging congestion control in low voltage feeders. Besides, we show that our primal solution approach offers better scalability and reliability than the dual solution by conducting comprehensive 3-phase power flow simulation studies of their implementation in the IEEE European Low Voltage Test Feeder.

Parts of the content and contributions of this work are published in:

- J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen, "Alternating Direction Method of Multipliers for decentralized electric vehicle charging control," in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, Dec 2013, pp. 6960–6965 [40]

- J. Rivera, C. Goebel, and H. A. Jacobsen, "Distributed Convex Optimization for Electric Vehicle Aggregators," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1852–1863, July 2017 [41]

- J. Rivera, M. Jergler, A. Stoimenov, C. Goebel, and H.-A. Jacobsen, "Using publish/subscribe middleware for distributed EV charging optimization," *Computer Science-Research and Development*, pp. 1–8, 2014 [42]

- J. Rivera and H.-A. Jacobsen, “A distributed anytime algorithm for network utility maximization with application to real-time EV charging control,” in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, Dec 2014, pp. 947–952 [43]
- J. Rivera, C. Goebel, and H.-A. Jacobsen, “A Distributed Anytime Algorithm for Real-Time EV Charging Congestion Control,” in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, ser. e-Energy ’15, 2015, pp. 67–76 [44]
- J. Rivera and H. A. Jacobsen, “On the Effects of Distributed Electric Vehicle Network Utility Maximization in Low Voltage Feeders,” *arXiv preprint arXiv:1706.10074*, 2017 [45].

1.5 Organization

The rest of this work is organized as follows: Chapter 2 provides an introduction to the basics of distributed optimization. In Chapter 3, we discuss the related work to our EV ADMM and EV NUM approaches. Our EV ADMM approach for scalable EV aggregator distributed optimization is introduced, evaluated and discussed in Chapter 4. The implementation of EV ADMM in a distributed environment using a pub/sub middleware is shown in Chapter 5. The EV NUM approach for EV charging congestion control is presented, evaluated and discussed in Chapter 6. Finally, Chapter 7 presents our conclusions.

Chapter 2

Background

In this chapter, we review some of the basic distributed optimization concepts required to understand the technical content in this thesis. First, we review the definition of a convex optimization problem, which is the class of problems we use in this thesis. Then, we review some of the basic methods for distributed optimization, which build the basis in our approaches. Many of the examples and definitions of the next sections are adapted from [46–48].

2.1 Convex optimization

In our problem formulations, we aim to obtain a convex optimization problem: The minimization of a convex function over a convex set of constraints. The reason we strive for a convex problem is that convex optimization problems posses a single optimal point (local minimum is the global minimum) that can be reached from anywhere within its feasible space. Many distributed optimization algorithms require this characteristic. To understand the importance of this, we first need to know what a convex set and what a convex function are.

A convex set is a set where a line segment can be drawn between any two points in the set without leaving the set. Hence, if a line segment between x_1 and x_2 is defined as

$$x = \theta x_1 + (1-\theta)x_2 \tag{2.1.1}$$

with $0 \leq \theta \leq 1$, then a convex set $\mathbf{C}$ is defined as

$$x_1, x_2 \in \mathbf{C}, \quad 0 \leq \theta \leq 1 \Longrightarrow \theta x_1 + (1-\theta)x_2 \in \mathbf{C}. \tag{2.1.2}$$

There are many convex sets, like cones, hyperplanes, halfspaces, euclidean balls and ellipsoids. However, the most prominent is the polyhedra, which can be considered as the solution set of finitely many linear inequalities and equalities, i.e., $Ax \leq b$ and $Cx = b$. In our problem formulations we use polyhedra sets, which are by definition convex.

Similar to a convex set, a function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex if $\mathbf{dom}f$ is a convex set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \tag{2.1.3}$$

for all $x, y \in \mathbf{dom}f, 0 \leq \theta \leq 1$. If f is concave then $-f$ is convex. A function f is strictly convex if $\mathbf{dom}f$ is convex and

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y) \tag{2.1.4}$$

for all $x, y \in \mathbf{dom}f, x \neq y, 0 \leq \theta \leq 1$. There are several function which are convex, e.g., affine functions $ax + b$, exponential e^{ax}, powers x^{α} for $0 \geq \alpha \geq 1$, negative logarithm $-log(x)$ and many more. It is important to make a distinction between strictly convex and non-strictly convex functions, because some distributed optimization approaches only support strictly convex optimization functions. As an example, a linear function $f(x) = x$ is convex but not strictly convex, while a quadratic function $f(x) = x^2$ is convex and also strictly convex.

With the lattes, we can define a convex optimization as follows:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) < b_i, \quad i = 1, \ldots, m \end{array} \tag{2.1.5}$$

where the objective and constraints are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \tag{2.1.6}$$

if $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$. The variables x are the optimization variables of the problem, also known as primal variables. There are several reliable and efficient algorithms available to solve convex problems. The challenge is that convex problems are usually difficult to recognize or formulate. There are many methods involved in transforming optimization problem formulations into a convex form. However, notably many problems can be solved via convex optimization, and many control engineering problems are formulated as convex optimization problems. This is also the case with many EV charging control problems, which is the reason why we focus on convex optimization problems. For more details on convex optimization, we refer readers to [46].

2.2 Distributed optimization

The objective in distributed optimization is to solve an optimization problem by decomposing it into easier to solve subproblems and a master problem that through an iterative procedure closes in on the optimal solution of the original problem. If the master algorithm converges fast enough and the subproblems are sufficiently easier to solve, we get savings in computation compared to a centralized optimization. Let's consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & x_1 \in \mathbf{C}_1, \quad x_2 \in \mathbf{C}_2. \end{array} \tag{2.2.1}$$

We can see that the problem above can be separated, because we can solve for x_1 and x_2 separately (in parallel). This sort of problems are called separable or trivially parallelizable problems. However these sort of problems are the exception. Let's now consider the following problem

$$\text{minimize} \quad f(x) = f_1(x_1, y) + f_2(x_2, y), \tag{2.2.2}$$

with $x = (x_1, x_2, y)$. This new kind of problem is not trivially separable because y is a complicating or coupling variable. However, when y is fixed the problem is separable in x_1 and x_2. In this case, x_1, x_2 are private or local variables and y is a public or interface variable between the subproblems. With this definitions, we can introduce the first decomposition technique, primal decomposition.

2.2.1 Primal decomposition

In primal decomposition, the idea is to decompose the original optimization problem to first optimize over the set of local variables (subproblems) and then to optimize over the set of coupling variables (master problem) in an iterative procedure. If we consider problem (2.2.2), by fixing y we obtain the following subproblems:

$$\begin{aligned} &\text{subproblem 1:} \quad \text{minimize}_{x_1} f_1(x_1, y) \\ &\text{subproblem 2:} \quad \text{minimize}_{x_2} f_2(x_2, y), \end{aligned} \tag{2.2.3}$$

with optimal values $\phi_1(y)$ and $\phi_2(y)$. The original optimization problem is then equivalent to the master problem:

$$\text{master problem:} \quad \text{minimize} \quad \phi_1(y) + \phi_2(y), \tag{2.2.4}$$

with variable y. The method is called primal decomposition since the master problem manipulates a primal variable. It is important to mention that if the original optimization problem is convex, the master problem is also convex. The master problem can be solved using several methods, and each iteration of the master problem requires solving the two subproblems in parallel.

The primal algorithm using a subgradient method to solve the master problem can be written as:

repeat

1. Solve the subproblems (in parallel)
   Find x_1 that minimizes $f_1(x_1, y)$ and a subgradient $g_1 \in \partial\phi_1(y)$.
   Find x_2 that minimizes $f_2(x_2, y)$ and a subgradient $g_2 \in \partial\phi_2(y)$.

2. Update complicating variable.
   $y = y - \alpha(g_1 + g_2)$, where α is the step size.

Each step in the algorithm described above brings us closer to the solution of the original problem. With this example of primal decomposition, we have provided the basics to understand our approach. For more details on primal decomposition, we refer readers to [48].

2.2.2 Dual decomposition

In dual decomposition, the idea is to decompose the original problem to first optimize over the set of local variables (subproblems) and then optimize over the set of dual variables (master problem) in an iterative procedure. We consider again problem (2.2.2) and to implement dual decomposition we modify it by introducing new variables y_1, y_2 as follows:

$$\begin{array}{ll} \text{minimize} & f(x) = f_1(x_1, y_1) + f_2(x_2, y_2) \\ \text{subject to} & y_1 = y_2. \end{array} \tag{2.2.5}$$

The newly introduced y_1, y_2 are local versions of the complicating variable y and the new constraint $y_1 = y_2$ is a coupling constraint for both subproblems. The second step is to formulate the master problem, which involves formulating the dual problem. To do this we first consider the Lagrangian of our modified problem (2.2.5):

$$L(x_1, y_1, x_2, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) + \lambda^T(y_1 - y_2), \tag{2.2.6}$$

where λ is a vector of Lagrangian variables. To separate the problem we can first minimize over (x_1, y_1) and (x_2, y_2) separately:

$$\text{subproblem 1:} \quad g_1(\lambda) = \inf_{x_1,y_1} \left(f_1(x_1, y_1) + \lambda^T y_1\right) \tag{2.2.7}$$

$$\text{subproblem 2:} \quad g_2(\lambda) = \inf_{x_2,y_2} \left(f_2(x_2, y_2) - \lambda^T y_2\right). \tag{2.2.8}$$

With the above the master problem optimizes the dual variables as follows:

$$\text{master problem:} \quad \text{maximize} \quad g(\lambda) = g_1(\lambda) + g_2(\lambda). \tag{2.2.9}$$

The method is called dual decomposition because the master problem optimizes over the dual variables. Computing the result of the subproblems $g_i(\lambda)$ can be done in parallel. The master problem can be solved using a gradient descent, which to maximize uses the negative gradient $y_2 - y_1$. With this, the dual algorithm can be defined as:
repeat

1. Solve the dual subproblems (in parallel)
   Find x_1, y_1 that minimizes $f_1(x_1, y_1) + \lambda^T y_1$.
   Find x_2, y_2 that minimizes $f_2(x_2, y_2) - \lambda^T y_2$.

2. Update dual variable.
   $\lambda = \lambda - \alpha(y_2 - y_2)$, where α is the step size.

At each step, we move closer to the optimal solution of the original problem. It is important to notice that iterates are generally infeasible, i.e., $y_1 \neq y_2$. With this example of dual decomposition, we have provided the basics to understand our approach. For more details on dual decomposition, we refer readers to [48].

2.2.3 Alternating direction method of multipliers

One of the problems with primal and dual decomposition is that for the method to converge to the optimal solution, the objective function must be strictly convex [48].

This is a significant limitation because many optimization problems have linear functions as objective functions. Linear functions are convex but not strictly convex. To remedy this, the Alternating Direction Method of Multipliers (ADMM) introduces a regularization parameter to dual decomposition that keeps the algorithm stable. In fact, the only requirement for ADMM to solve an optimization problem is convexity. We now quickly explain the basic formulation of ADMM.
Let's consider the following problem:

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c. \end{array} \tag{2.2.10}$$

The problem above represents two sets of variables x and z, with separable objectives. The Lagrangian of this problem is defined as:

$$L(x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c), content \tag{2.2.11}$$

where *lambda* is a vector of Lagrangian variables.content To formulate ADMM we introduce a quadratic regularization parameter to the original Lagrangian to obtain an augmented Lagrangian function:

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c) + (\rho/2)||Ax + Bz - c||_2^2, \tag{2.2.12}$$

where $\rho > 0$ is the penalty parameter and $|| * ||_2^2$ is the second norm squared.

The ADMM algorithm can then be formulated in the same manner as the dual decomposition algorithm to solve the subproblems for the local variables x and z, and then solve the master problem for the dual variable λ. The resulting ADMM algorithm is:

$$x^{k+1} = \text{argmin}_x L_\rho(x, z^k, \lambda^k) \tag{2.2.13}$$

$$z^{k+1} = \text{argmin}_z L_\rho(x^{k+1}, z, \lambda^k) \tag{2.2.14}$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \tag{2.2.15}$$

where k is the iteration index.
Depending on the specific problems several simplifications can be made for the

ADMM algorithm. Like the dual algorithm, the ADMM algorithm approaches feasibility and the objective approaches the optimal values as it converges. However, ADMM assumes very little and can be applied to a wider range of problems than dual decomposition. For more details on ADMM, we refer readers to [47].

Chapter 3

Related Work

In this chapter, we review the related work to our two EV charging control approaches: EV ADMM and EV NUM. For EV ADMM the related work is concerned with the problem of scalable EV charging control for EV aggregators. For EV NUM the related work is concerned with the real-time control of EV charging. In the following, we discuss how our approaches differ from the existing literature.

3.1 Scalable EV charging control for EV aggregators

Several studies have addressed the problem of defining scalable methods for EV charging control. Most of the proposed approaches, however, focus on a specific control objective and do not provide a general solution to the scalability problems faced by EV aggregators. The scalability challenge lies in the definition of a versatile method that for very large EV numbers can provide optimal results in a reasonable time and make efficient use of limited resources. One solution to the scalability problem is the use of suboptimal control strategies, which provide scalability with a decreased level of optimality. In [49,50], for example, a droop control approach is used for EVs participating in frequency control. Such approaches consider a scenario where communication with the EVs is limited or unavailable. In contrast to such approaches, our approach considers a scenario, where a two-way communication system is in place, and optimal results are required. We focus on distributed optimization of EV charging for a broad range of different objectives. Similar approaches based on distributed optimization have already been proposed to solve the valley filling problem in [25–27]. Nevertheless, these approaches cannot be implemented to solve other relevant control objectives, such as charging cost minimization [21] or direct coupling between renewable energy and EV demand [22]. The previous approaches cannot be generalized because they require a strictly convex objective function and do not support global constraints, i.e., constraints defined as a function of the optimization variables of all EVs. These limitations prevent the use of these approaches to obtain scalable solutions to several relevant EV aggregator optimization problems.

The approach presented in this work and published in [40] differentiates itself from previous approaches in that it offers a general distributed optimization framework for EV aggregators, that is capable of solving various EV charging control problems. Our approach is based on the Alternating Direction Method of Multipliers (ADMM), which supports non-strictly convex cost functions and global constraints. Following our initial work, several studies have used ADMM for the optimization of EV fleet charging problems. The authors of [28] use ADMM for the load balancing of EVs at charging stations. The goal of their work was to maximize the utility functions of several devices (EVs) controlled by an aggregator. In [29], the authors compare the use of several distributed optimization algorithms for the valley filling problem. They conclude that different algorithms offer a trade-off between communication requirements and optimality: Allowing suboptimal results in ADMM can help reduce its message complexity. Finally, in [30], ADMM is used to formulate a receding horizon optimization to account for uncertainty in the valley filling problem. Nevertheless, despite the valuable contributions of these related works, none of them conducts a comprehensive scalability study. In fact, most of the studies published to date are limited to 100 EVs. Furthermore, a comparison against state-of-the-art centralized optimization has thus far not appeared in the literature. In centralized optimization, the original problem formulation is solved using a state-of-the-art optimization software. With an increasing number of EVs, it is of particular importance to determine under what conditions a distributed optimization approach offers an advantage over the centralized approach. Therefore, the work presented here and published in [41] builds on the existing literature by providing a full derivation of the general EV ADMM method and conducting a comprehensive evaluation focusing on the scalability of our distributed approach compared to the state-of-the-art centralized approach.

While the proposed EV ADMM approach can be used to speed up computations in a single machine by distributing the optimization among multiple cores, such distributed optimization approaches can also be implemented as a protocol for distributed EV charging control in a distributed environment. Most related studies in the literature, don't consider the implementation of their distributed optimization approach in a distributed environment and instead focus on the correctness and the effectiveness of the proposed algorithms. So far there is very limited work proposing and evaluating a concrete architecture for the realization and deployment of distributed optimization approaches for EV aggregators. The work in [51], stands out for proposing the scheduling of various energy devices (including EVs) with ADMM and implementing it in a distributed environment using a message passing interface. They, however, acknowledge that such algorithms require the exchange of a high number of messages and that a reduction of the message complexity is needed. This is not a trivial issue, since a large number of messages required may become prohibitive for practical application in a distributed environment. Our work differentiates itself from these previous work in two aspects. First, we focus on EV aggregator problems, which have more complex objectives and constraints than the ones they consider. Second, we address the issue of communication complexity by using of a publish and subscribe (pub/sub) middleware to implement EV ADMM. We propose DOPS (Distributed Optimization Publish and Subscribe), a framework for distributed optimization of EV charging, which leverages the in-network aggregation capabilities of the pub/sub middleware to reduce the number of messages exchanged between the EVs and the aggregator.

3.2 Real-time EV charging congestion control

Several studies have used distributed optimization techniques to design scalable EV charging control algorithms. Many of them, we discussed in the previous section. However, all of the mentioned approaches in Section 3.1 are multiperiod optimizations, which to provide EV congestion control would require very accurate predictions of the EVs' location, charging requirements and the state of the grid. Such predictions are extremely challenging for a single low voltage feeder, and can be highly inaccurate for individual EVs and individual grid devices. An alternative is to consider the solution of a single period optimization problem based on the current state of the system. The lack of prediction, however, requires very fast control responses to cope with the highly dynamic state changes of the power grid. This makes real-time EV charging a challenging time critical application.

The main challenge of real-time EV charging is the short time available to deliver a control response. The time range to avoid a fault in the distribution grid is usually in the hundreds of milliseconds [52]. Some approaches to prevent the violation of network constraints by controlling EV charging under real-time requirements have been proposed in the literature. In [34], an EV charging desynchronization approach to prevent too many charging EVs from overloading the grid is proposed. Similarly, in [35], an AIMD[1]-like control approach is proposed to avoid grid overloading. Although both methods prevent network overloading, they do not guarantee an optimal use of the available infrastructure. Hence, the power infrastructure is underutilized, and the EVs' charging speed is reduced. In contrast to these approaches, we focus on the optimal use of the available infrastructure. We consider the use of optimization-based methods and propose distributed optimization algorithms to cope with the real-time requirements.

Distributed optimization for real-time EV charging control has been used for several objectives. In [53] a distributed optimization approach was used to maximize user convenience while meeting predefined circuit level demand limits. The problem was formulated as a single period combinatorial optimization problem, subsequently

[1]The additive-increase/multiplicative-decrease (AIMD) algorithm is a feedback control algorithm best known for its use in TCP Congestion Avoidance.

relaxed into a convex optimization problem and then solved with ADMM. Their approach focuses on the control of EV charging via an aggregator to provide system services in real-time. In [54] distributed optimization is used to control EV charging in real-time, to avoid the overloading of distribution grid transformers. Particular attention is placed to the modeling of the transformers thermal constraints. Then an optimization problem to maximize the charging rate of EVs without violating the transformers maximal thermal capacity constraints is solved with dual decomposition. This results in an incentive-based real-time EV charging control protocol. However, while both mentioned approaches relate to ours in their use of distributed optimization, none of them models or considers a model of the distribution grid. This makes them inadequate for EV charging congestion control, where individual distribution lines have to be considered to avoid their overloading.

A promising approach comes from the related problem of congestion control in communication networks. There, a Network Utility Maximization (NUM) problem is solved with distributed optimization approaches to obtain Transmission Control Protocols (TCP) [15]. The use of the NUM formulation for real-time EV charging control is first considered in [36]. In the latter, the EV charging problem is formulated as a modified NUM, and dual decomposition is used to solve the problem. The NUM formulation defines an instantaneous problem. Hence, no prediction is required. Moreover, thanks to the simplicity of the NUM formulation the resulting algorithm is highly efficient. However, the proposed dual decomposition solution suffers from scalability problems. The method can become unstable with increasing EV numbers or in some situations may not adapt quickly enough to the grid dynamics, which the authors recognized in [37]. In this work, which we partially published in [43, 44], we build and improve on what we call the EV NUM approach. To improve scalability and stability, we propose a novel primal decomposition algorithm to solve EV NUM, which has the property of allowing anytime control. We provide mathematical proof for its convergence to the optimal result under static conditions. We also derive a real-time EV charging control algorithm and analyzed its behavior under dynamic conditions. Moreover, we contribute a comprehensive comparison of the dual and our primal EV NUM distributed optimization algorithms. We evaluate their convergence behavior, scalability, and grid impact on the IEEE 13 Node Test Feeder and the IEEE European Low Voltage Test Feeder. The latter realistic distribution grid analysis is a significantly larger and more representative use case compared to the all previous works. In fact, none of the previous works offer a comprehensive 3 phase power flow analysis. Since the EV NUM formulation does not consider losses and voltage constraints, the evaluation presented in this work represents a vital contribution to the implementation of EV NUM for real-time EV charging congestion control.

CHAPTER 4

Distributed Optimization for EV Aggregators with EV ADMM

In this chapter, we develop the EV ADMM framework for distributed optimization of EV aggregators.
First, in Section 4.1 we state our assumptions for the EV aggregator optimization problem. In Section 4.2 we propose a general formulation for EV aggregator optimization problems and solve it using ADMM. Our proposed formulation allows the general definition of global objectives and constraints for the whole EV fleet, and local objectives and constraints for the individual EVs. This provides the flexibility to specify various EV aggregator optimization problems. The general aggregator problem is then solved using ADMM to obtain a distributed solution algorithm, where the original problem is split into smaller individual optimization subproblems for the individual EVs and the aggregator. These individual problems are then solved in an iterative procedure to obtain the optimal solution of the original problem. The various mathematical steps required to obtain the EV aggregator ADMM solution algorithm as well as its convergence criteria and parametrization are explained.
In Section 4.3, we define our EV ADMM framework, which allows the aggregator and the EVs to formulate their local optimization independently of each other. Then by incorporating them into the EV ADMM sequence, the solution of the original EV aggregator problem can be obtained. We describe the different computing

components of EV ADMM and their interaction. Moreover, we offer the specific formulation of the individual aggregator optimization problem for the valley filling and the charging cost minimization problem, as well as, the specific formulation of the individual EV optimization problem that considers its goal to reduce battery depreciation and its energy requirement constraints.
The evaluation of the versatility and scalability of EV ADMM is presented in Section 4.4. First, we demonstrate the versatility of EV ADMM by solving different EV aggregator optimization problems. We consider individual EV aspects, like battery depreciation costs and the ability of EVs to feed energy into the grid, also known as Vehicle-to-Grid (V2G) services. In our second evaluation, we study the scalability of EV ADMM and show that for large EV numbers it outperforms the state-of-the-art centralized optimization approach. In centralized optimization, the original problem formulation is solved using state-of-the-art optimization software.
Finally, Section 4.5 presents a discussion on the implementation and applicability of EV ADMM.

4.1 Assumptions

To formulate the EV aggregator optimization problem, we assume the following:

- We consider the day-ahead scheduling of 24 hours divided into 15 min intervals for a fixed number of EVs.
- The aggregator can have various objectives for its EV fleet, depending on the service it wants to provide.
- The individual EVs can have personal objectives, such as minimizing their battery depreciation costs.
- The charging requirements of the EVs must always be fulfilled.
- All parameters of the optimization problem are assumed to be known. Therefore, all non-fixed parameters must be predicted. This also includes the charging requirements of the EVs, i.e., the time slots when an EV will be charging and

how much energy it requires. The EV one day-ahead can communicate the charging requirements, or they can be predicted using the EV travel logs, also known as the EV's driving profile.

- The problem formulation is convex. An important consequence of this assumption is that the EVs must be able to use a charging rate that is continuous between 0 and a maximal value.

4.2 The EV aggregator optimization problem

We propose a novel general formulation of the EV aggregator optimization problem. The problem is formulated as a multi-criterion optimization between the aggregator and individual EVs for a defined time horizon $t \in \{1, \dots, T\}$. For purposes of our research, we consider a fixed number of EVs $i \in \{1, \dots, N\}$. Each individual EV seeks to optimize its charging profile defined by the vector $x_i = [x_i(1), \dots, x_i(T)]^T$ for its local objective function $F_i(x_i)$ subject to its local constraint set $\mathbb{X}_i$. The aggregator seeks to optimize the sum of the individual EV charging profiles defined by the vector $x_a = [x_a(1), \dots, x_a(T)]^T$ for the global objective $F_a(x_a)$ subject to the global constraint set $\mathbb{X}_a$. Since the aggregator optimization variables are the sum of the individual EV variables, $x_a = \sum_{i=1}^{N} x_i$, all the individual optimization problems are coupled. Considering this, we define the EV aggregator optimization problem as follows:

$$\begin{array}{ll} \text{minimize} & F_a(x_a) + \gamma \sum_{i=1}^{N} F_i(x_i) \\ \text{subject to} & x_a = \sum_{i=1}^{N} x_i \\ & x_a \in \mathbb{X}_a \\ & x_i \in \mathbb{X}_i; \quad i = 1, \dots, N, \end{array} \tag{4.2.1}$$

where $\gamma \geq 0$ is a parameter representing the trade-off between the aggregator's objective and the objectives of the individual EVs. The parameters of the problem are summarized in Table 4.2.1. A positive value of the variables is defined as power consumption and a negative as power generation.

The trade-off parameter balances the importance of the individual EV objectives and

Variable	**Description**	**Type**
x_a	Aggregated EV profiles	Vector $\in \mathbb{R}^T$
x_i	Individual EV profile	Vector $\in \mathbb{R}^T$
$F_a(x_a)$	Aggregator objective function	Convex function
$F_i(x_i)$	EV i objective function	Convex function
$\mathbb{X}_a$	Aggregator constraints set	Convex set
$\mathbb{X}_i$	EV i constraints set	Convex set
γ	Trade-off parameter	Scalar

Each element in $\mathbb{R}^T$ is a vector of T real numbers.

Table 4.2.1: Nomenclature for EV aggregator optimization problem

the global aggregator objectives. As mentioned in [55] and [8], the objectives of the aggregator are usually in conflict with the individual goals of the EVs. For instance, while the aggregator wants to coordinate EVs to achieve a particular aggregated behavior, the EVs want to minimize battery depreciation costs. This usually causes the EVs to favor their objectives, which may lead to a poor aggregated performance for the global objective, cf. [40].

Our formulation of the problem is very general and can be used to describe optimization problems where multiple devices with local objectives and constraints need to be optimized under global objectives and constraints represented by an aggregator. Examples of specific formulations of the EV aggregator optimization problem will be given in Section 4.3.

To formulate a distributed optimization algorithm, we first reformulate the problem as an exchange optimization problem. To do this, we consider the EVs and the aggregator as having separate subproblems and redefine the aggregator as subproblem 0, where

$$x_0 = -x_a. \tag{4.2.2}$$

The cost function of subproblem 0 is then:

$$f_0(x_0) = \begin{cases} F_a(-x_0) & \text{if } -x_0 \in \mathbb{X}_a \\ \infty & \text{otherwise.} \end{cases} \tag{4.2.3}$$

The EVs are represented by subproblems $i = 1, \dots, N$ and their cost function is:

$$f_i(x_i) = \begin{cases} \gamma F_i(x_i) & \text{if } x_i \in \mathbb{X}_i \\ \infty & \text{otherwise.} \end{cases} \tag{4.2.4}$$

With these definitions, the optimal EV aggregator optimization problem becomes an exchange problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=0}^{N} f_i(x_i) \\ \text{subject to} & \sum_{i=0}^{N} x_i = 0. \end{array} \tag{4.2.5}$$

The exchange problem considers N devices exchanging a common good under equilibrium constraints. To simplify notation, we define a single vector that contains the optimization variables of all subproblems $x = [x_0, x_1, \dots, x_N]^T$. The variables vector x has the dimension $NT \times 1$. The formulation of the ADMM solution to the exchange problem requires the introduction of a new vector of variables $z = [z_0, z_1, \dots, z_N]^T$. Using the z variables, we reformulate the exchange problem as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=0}^{N} f_i(x_i) + g(z) \\ \text{s.t} & x_i = z_i; \quad i = 0 \dots, N, \end{array} \tag{4.2.6}$$

where

$$g(z) = \begin{cases} 0 & \text{if } \sum_{i=0}^{N} z_i = 0 \\ \infty & \text{otherwise.} \end{cases} \tag{4.2.7}$$

Problem (4.2.6) is equivalent to the original exchange problem defined in (4.2.5). With $f(x) = \sum_{i=0}^{N} f_i(x_i)$, the corresponding augmented Lagrangian function of (4.2.6) is:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(x - z) + \frac{\rho}{2}||x - z||_2^2, \tag{4.2.8}$$

where ρ is the penalty parameter of the augmented term and $y = [y_0, y_1, \dots, y_N]^T$ is a vector of the Lagrangian variables. To solve problem (4.2.6), we minimize over the primal variables x, z and maximize over the Lagrangian variables y as explained in [56]:

$$\underset{y}{\text{argmax}}\ \underset{x,z}{\text{argmin}}\ L_\rho(x, z, y). \tag{4.2.9}$$

The ADMM algorithm defines the following iterative process to solve the problem:

$$x^{k+1} = \underset{x}{\text{argmin}}\ L_\rho(x, z^k, y^k) \tag{4.2.10}$$
$$z^{k+1} = \underset{z}{\text{argmin}}\ L_\rho(x^{k+1}, z, y^k) \tag{4.2.11}$$
$$y^{k+1} = \underset{y}{\text{argmax}}\ L_\rho(x^{k+1}, z^{k+1}, y), \tag{4.2.12}$$

where k is the index of the iteration. Expanding (4.2.10), we get:

$$x^{k+1} = \sum_{i=0}^{N} \underset{x_i}{\text{argmin}} \left\{ f_i(x_i) + y_i^{k^T}(x_i - z_i^k) + \tfrac{\rho}{2}||x_i - z_i^k||_2^2 \right\}. \tag{4.2.13}$$

The problem above is separable and we can thus define one problem for each x_i-update. The iterative ADMM solution of the exchange problem can then be written as:

$$x_i^{k+1} = \underset{x_i}{\text{argmin}}\ f_i(x_i) + y_i^{k^T}(x_i - z_i^k) + \frac{\rho}{2}||x_i - z_i^k||_2^2 \tag{4.2.14}$$
$$z^{k+1} = \underset{z}{\text{argmin}}\ g(z) - y^{k^T} z + \frac{\rho}{2}||x^{k+1} - z||_2^2 \tag{4.2.15}$$
$$y^{k+1} = \underset{y}{\text{argmax}}\ y^T(x^{k+1} - z^{k+1}). \tag{4.2.16}$$

The EV ADMM solution above can be further simplified. The z-update in (4.2.15) has an analytical solution and a gradient algorithm can be used to solve the y-update in (4.2.16):

- Solution to z^{k+1} update in (4.2.15): Taking into account the definition of $g(z)$ in (4.2.7), we can formulate the optimization problem for z as:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=0}^{N} \left\{ -y_i^{k^T} z_i + \frac{\rho}{2}||x_i^{k+1} - z_i||_2^2 \right\} \\ \text{subject to} \quad & \textstyle\sum_{i=0}^{N} z_i = 0. \end{aligned} \tag{4.2.17}$$

For this problem, the corresponding Lagrangian is

$$\sum_{i=0}^{N} L(z_i, \lambda) = \sum_{i=0}^{N} \{-y_i^{kT} z_i + \frac{\rho}{2}||x_i^{k+1} - z_i||_2^2 + \lambda^T z_i\}, \tag{4.2.18}$$

where λ is a vector of the Lagrangian variables. To obtain the solution to this problem, we use the KKT conditions:

$$\nabla_{z_i} L(z_i, \lambda) = 0 \quad \rightarrow \quad z_i = x_i^{k+1} + \frac{y_i^k}{\rho} - \frac{\lambda}{\rho} \tag{4.2.19}$$

$$\nabla_{\lambda} L(z, \lambda) = 0 \quad \rightarrow \quad \sum_{i=0}^{N} z_i = 0. \tag{4.2.20}$$

Using (4.2.19) in (4.2.20), we obtain the following:

$$\lambda = \rho(\overline{x}^{k+1} + \frac{\overline{y}^k}{\rho}), \tag{4.2.21}$$

where $\overline{x}^{k+1} = 1/(N+1)\sum_{i=0}^{N} x_i^{k+1}$ and $\overline{y}^{k+1} = 1/(N+1)\sum_{i=0}^{N} y_i^k$, i.e., the averages based on the number of subproblems. Using the results of Equations (4.2.19) and (4.2.21), we finally obtain a closed-form expression for the z-update:

$$z_i^{k+1} = x_i^{k+1} - \overline{x}^{k+1} + \frac{y_i^k}{\rho} - \frac{\overline{y}^k}{\rho}. \tag{4.2.22}$$

- Solution of y^{k+1} in (4.2.16): Since L_ρ is not differentiable for y, we solve the problem using the gradient method for each element of the vector y^{k+1}:

$$y_i^{k+1} = y_i^k + \rho(x_i^{k+1} - z_i^{k+1}). \tag{4.2.23}$$

The step size ρ of the gradient method is the same as the penalty parameter of the augmented term ρ, see (4.2.8). This step size is chosen to guarantee convergence. With this step size, the first condition of the dual feasibility is always fulfilled, cf. [38].

Using (4.2.22) in (4.2.23), we obtain the following for the y-update:

$$y_i^{k+1} \quad = \quad y_i^k + \rho\overline{x}^{k+1}. \tag{4.2.24}$$

The previous result shows that y is a vector of the same element and its average is defined as:

$$\overline{y} \quad = \quad y_i. \tag{4.2.25}$$

Using the result for the y-update (4.2.24) in the z-update (4.2.22), we get the following z-update solution:

$$z_i^{k+1} = x_i^{k+1} - \overline{x}^{k+1}. \tag{4.2.26}$$

Finally, the z-update defined above is included into the x-update in (4.2.14). With this, the x-update becomes:

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\ f_i(x_i) + y_i^{k^T} x_i + \frac{\rho}{2}||x_i - x_i^k + \overline{x}^k||_2^2. \tag{4.2.27}$$

With the solution of the x-update in (4.2.27) and the gradient method for the y-update in (4.2.24), the ADMM solution to the exchange problem is:

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\ f_i(x_i) + y_i^{k^T} x_i + \frac{\rho}{2}||x_i - x_i^{k+1} + \overline{x}^{k+1}||_2^2 \tag{4.2.28}$$

$$y_i^{k+1} = y_i^k + \rho\overline{x}^{k+1}. \tag{4.2.29}$$

The solution above can be regularized by scaling the Lagrangian variables y_i with $u = y_i/\rho$: With this, the scaled ADMM solution of the ADMM exchange problem is:

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\ f_i(x_i) + \frac{\rho}{2}||x_i - x_i^k + \overline{x}^k + u^k||_2^2 \tag{4.2.30}$$

$$u^{k+1} = u^k + \overline{x}^{k+1}. \tag{4.2.31}$$

Using the exchange problem solution above and taking into account the definitions in (4.2.2), (4.2.3) and (4.2.4), we formulate the ADMM solution to the EV aggregator problem as follows:

EV Aggregator ADMM algorithm

1. For each EV $i = 1, \ldots, N$:

$$\begin{aligned} x_i^{k+1} = & \\ \text{minimize} \quad & \gamma F_i(x_i) + \tfrac{\rho}{2}||x_i - x_i^k + \overline{x}^k + u^k||_2^2 \\ \text{subject to} \quad & x_i \in \mathbb{X}_i. \end{aligned} \tag{4.2.32}$$

2. For the aggregator:

$$\begin{aligned} x_0^{k+1} = & \\ \text{minimize} \quad & F_a(-x_0) + \tfrac{\rho}{2}||x_0 - x_0^k + \overline{x}^k + u^k||_2^2 \\ \text{subject to} \quad & -x_0 \in \mathbb{X}_a. \end{aligned} \tag{4.2.33}$$

3. Incentive signal update:

$$u^{k+1} = u^k + \overline{x}^{k+1}, \tag{4.2.34}$$

where

$$\overline{x}^{k+1} = \frac{1}{N+1} \sum_{i=0}^{N} x_i^{k+1} \tag{4.2.35}$$

The EVs and the aggregator solve their optimization subproblems independently in each step k, while the continuous update of the incentive signal drives the solution to the optimum of the original EV aggregator problem. The ADMM algorithm approaches the solution of the original EV aggregator problem in (4.2.1) as long as the problem is feasible and convex.

Convergence criteria: As defined in [38], the convergence criteria for ADMM are given by the primal feasibility $r^k \in \mathbb{R}^T$ and the dual feasibility $s_i^k \in \mathbb{R}^T$:

$$r^k = \overline{x}^k \tag{4.2.36}$$

$$s_i^k = -\rho(N+1)(x_i^k - x_i^{k-1} + (\overline{x}^{k-1} - \overline{x}^k)), \tag{4.2.37}$$

with $s^k = [s_1^k, \ldots, s_N^k]^T$, the stopping criteria is:

$$\begin{aligned} ||r^k||_2 &\leq \varepsilon^{pri} \\ ||s^k||_2 &\leq \varepsilon^{dual}, \end{aligned} \tag{4.2.38}$$

where $\varepsilon^{pri} > 0$ and $\varepsilon^{dual} > 0$. The mean value of all subproblems $\overline{x}$ can be thought of as a social cost caused by the EVs not cooperating to achieve global convergence. Upon convergence, this social cost should be close to zero as specified by the primal convergence criteria in (4.2.36).

Penalty parameter ρ: Although some guidelines for choosing ρ can be found in [38] and [51], the literature does not offer a comprehensive method for this task. In this work, we have chosen to use a variable step size in order to improve convergence speed (cf. [38]):

$$\rho^{k+1} = \begin{cases} \tau^{incr}\rho^k & \text{if } ||r^k||_2 > \mu||s^k||_2 \\ \rho^k/\tau^{decr} & \text{if } ||s^k||_2 > \mu||r^k||_2 \\ \rho^k & \text{otherwise.} \end{cases} \tag{4.2.39}$$

For our experiments in Section 4.4, we use $\mu = 10$ and $\tau^{incr} = \tau^{decr} = 2$.

4.3 The EV ADMM framework

The EV ADMM framework is the protocol that results from the application of the ADMM approach to the EV aggregator optimization problem. Fig. 4.3.1 describes the framework using a sequence diagram. EV ADMM is composed of several computing components: Aggregator, messaging system, and EVs. The aggregator first uses the messaging system to send the incentive signal to the EVs. The broadcasted signal is the same for all EVs. This incentive signal is composed of the scaled price u^k, the average profile of all subproblem solutions $\overline{x}^k$, and also the penalty parameter ρ in case of a varying penalty parameter. Based on this incentive signal, the EVs and the aggregator solve their individual optimization subproblems, defined in (4.2.32) and (4.2.33). The EVs then send their solutions to the aggregator using the messaging system, $\overline{x}_{EV}^{k+1} = \frac{1}{N+1}\sum_{k=1}^{N} x_i^{k+1}$. This aggregated response can be

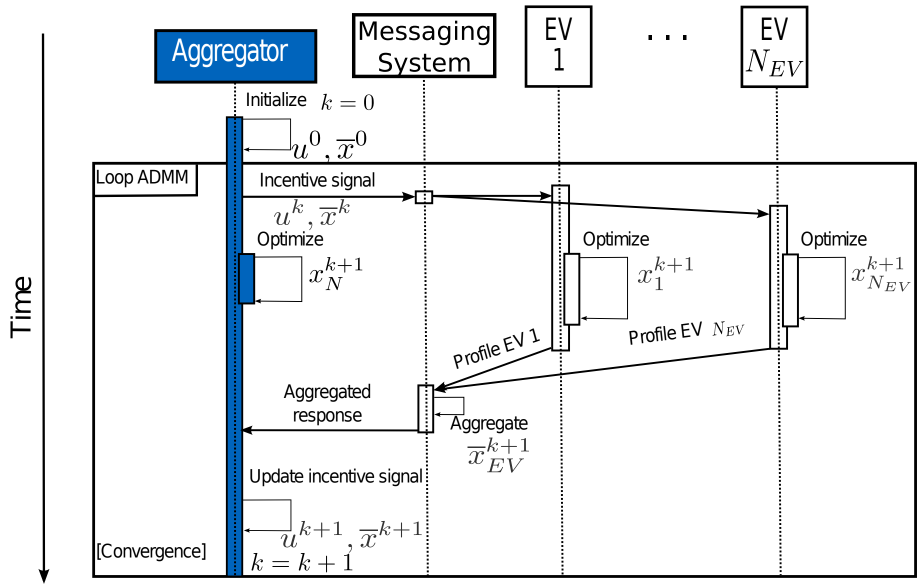


Figure 4.3.1: EV ADMM sequence diagram.

computed by the messaging system if in-network aggregation is supported. The messaging system receives the individual EV responses and sends the aggregated EV response to the aggregator. The aggregator then adds its result to obtain the average of all subproblems, i.e., $\overline{x}^{k+1} = \overline{x}_{EV}^{k+1} + \frac{x_0^{k+1}}{N+1}$. With this value, the aggregator then updates the scaled price u^{k+1} according to (4.2.34). This process is repeated until the convergence criteria in (4.2.38) are met.

Using EV ADMM, the aggregator and the EV optimization subproblems can be defined independently of each other. The solution of the original EV aggregator problem is obtained by incorporating these problems into the EV ADMM framework.

4.3.1 Aggregator optimization problem

In this section, we specify two typical aggregator objectives, valley filling and charging cost minimization. We then show how to integrate them into the proposed optimization framework. Different aggregator objectives lead to different objective functions $F_a(x_a)$ and constraint sets $\mathbb{X}_a$ in problems (4.2.1) and (4.2.33). The considered problems are summarized in Table 4.3.1.

Problem	$F_a(x_a)$	$\mathbb{X}_a$
Valley Filling	$\delta\|\|D + x_a\|\|_2^2$	-
Charging cost minimization	$p^T x_a \Delta t$	$\underline{x}_a \leq x_a \leq \overline{x}_a$

Table 4.3.1: Example aggregator individual problem definition

Valley filling: The goal of valley filling is to dispatch EV charging such that the valleys of the fixed demand are filled by the EVs' demand. This avoids new peaks in the power consumption. Our formulation of the valley filling problem is similar to the one in [26]:

$$\text{minimize} \quad \delta||D + x_a||_2^2, \tag{4.3.1}$$

where the fixed demand profile $D \in \mathbb{R}^T$ is assumed to be known and δ is a scaling parameter that can be used as an empirical value to change the unit of the aggregator's objective function. Using the formulation in (4.2.33), the corresponding aggregator update in the EV ADMM framework is:

$$x_0^{k+1} = \min_{x_N} \quad \delta||D - x_0||_2^2 + \tfrac{\rho}{2}||x_0 - x_0^k + \overline{x}^k + u^k||_2^2, \tag{4.3.2}$$

which has the following closed-form solution,

$$x_0^{k+1} = \tfrac{\rho}{\rho+2\delta}(x_0^k - \overline{x}^k - u^k) + \tfrac{2\delta}{\rho+2\delta}D. \tag{4.3.3}$$

Charging cost minimization: For the charging cost minimization problem, the goal is to minimize the charging costs of an EV fleet while satisfying constraints on the minimal and maximal aggregated (dis)charging power. In [21], a centralized solution to this problem is presented, where the aggregated charging is constrained by the available renewable energy generation. Using EV ADMM, we can formulate a decentralized solution to this problem. To do this, we first define the aggregator problem as follows:

$$\begin{aligned} \text{minimize} \quad & p^T x_a \Delta t \\ \text{subject to} \quad & \underline{x}_a \leq x_a \leq \overline{x}_a. \end{aligned} \tag{4.3.4}$$

where Δt is the time step duration. The maximal aggregated demand profile $\overline{x}_a \in \mathbb{R}^T$,

Problem	$F_i(x_i)$	$\mathbb{X}_i$
Minimize charging costs	$\alpha_i\|\|x_i\|\|_2^2$	$A_i x_i = R_i$ $\underline{S}_i \leq B_i x_i \leq \overline{S}_i$ $\underline{x}_i \leq x_i \leq \overline{x}_i$.

Table 4.3.2: Example EV individual problem definition

the maximal power profile that can be fed back into the grid $\underline{x}_a \in \mathbb{R}^T$, and the electricity price $p \in \mathbb{R}^T$ are assumed to be known.
The minimal charging cost optimization problem in (4.3.4) is reformulated for the EV ADMM framework according to (4.2.33) as:

$$\begin{aligned} x_N^{k+1} = & \\ \text{minimize} \quad & -p^T x_0 \Delta t + \tfrac{\rho}{2}||x_0 - x_0^k + \overline{x}^k + u^k||_2^2 \\ \text{subject to} \quad & -\underline{x}_a \geq x_0 \geq -\overline{x}_a. \end{aligned} \tag{4.3.5}$$

4.3.2 EV charging optimization problem

Without the loss of generality, in this section, we describe an example formulation of the individual EV charging optimization problem. This example shows a concrete definition of the EV's objective function $F_i(x_i)$ and the constraint set $\mathbb{X}_i$, which are summarized in Table 4.3.2. We consider the objective of the EV to minimize their battery depreciation cost under the constraints that their charging requirements are always meet.

In this formulation, the EV is allowed to charge more than once. We assume that the time slots for which the EV is connected are known. We also assume that the amount of energy required in each charging session is known. This implies that the EV's battery energy levels at the start of the charging session, as well as the desired energy level at the end of the charging session, are known. During each charging session, we consider the following linear battery model:

$$E_i(t+1) = E_i(t) + x_i(t)\Delta t, \tag{4.3.6}$$

where at time slot t, $E_i(t)$ is the energy stored in the battery and $x_i(t)$ is the

Variable	**Description**	**Type**
x_i	EV charging profile	Vector $\in \mathbb{R}^T$
$\underline{x}_i$	Minimal charging power	Vector $\in \mathbb{R}^T$
$\overline{x}_i$	Maximal charging power	Vector $\in \mathbb{R}^T$
α_i	Battery depreciation parameter	Scalar
A_i	Connection matrix	Matrix $\in \mathbb{R}^{c_i \times T}$
B_i	Input matrix	Matrix $\in \mathbb{R}^{T_{c_i} \times T}$
R_i	Charging requirements	Vector $\in \mathbb{R}^{c_i}$
$\underline{S}_i$	Minimal state charging power	Vector $\in \mathbb{R}^{T_{c_i}}$
$\overline{S}_i$	Maximal state charging power	Vector $\in \mathbb{R}^{T_{c_i}}$

Each element in $\mathbb{R}^T$ is a vector of T real numbers and each element in $\mathbb{R}^{c_i \times T}$ is a matrix of real numbers with the dimension $c_i \times T$.

Table 4.3.3: Nomenclature for EV optimization problem

power used to (dis)charge the battery. Using the linear battery model in (4.3.6), we define the required energy as a single equality constraint that depends only on the EV's power x_i. Similar to [21], we define the energy requirements and maximal and minimal energy levels for the battery as a series of constraints for all charging sessions, which only depend on the EV's power profile for the optimization time horizon x_i. With these constraints, we define the optimization problem of each EV i as:

$$\begin{aligned} \text{minimize} \quad & \alpha_i ||x_i||_2^2 \\ \text{subject to} \quad & A_i x_i = R_i \\ & \underline{S}_i \leq B_i x_i \leq \overline{S}_i \\ & \underline{x}_i \leq x_i \leq \overline{x}_i. \end{aligned} \tag{4.3.7}$$

The variables description of problem (4.3.7) can be found in Table 4.3.3, where c_i is the number of charging sessions of the EV and T_{c_i} is the total number of time slots that the EV is connected during the entire optimization horizon. We model the EV's objective to reduce its battery depreciation cost as a quadratic cost function. With this objective function, we obtain a convex formulation of the

problem, which is necessary to guarantee the convergence of EV ADMM to the optimal solution. This resembles the approach used for the generator's objective function in the economic dispatch problem, where a convex objective function is obtained by fitting the generator's costs with its power output. This is done to guarantee a convex formulation of the economic dispatch problem. The correctness of our model is supported by the results presented in [57], where battery depreciation was shown to be quadratic with respect to the depth of discharge (DoD). The empirical parameter α_i can be obtained by fitting historical data. The first set of equality constraints, $A_i x_i = R_i$, defines the energy requirements. Each of these equality constraints represents an energy requirement for each time the EV is plugged in. The set of inequality constraints, $\underline{S}_i \leq B_i x_i \leq \overline{S}_i$, guarantee that the power input does not violate the maximal and minimal energy that the battery can support at any time slot when the EV is connected. These constraints contain the EV dynamics while connected. Thus, matrix B_i is the input matrix that results from (4.3.6) for the time slots when the EV is connected. The last set of constraints, $\underline{x}_i \leq x_i \leq \overline{x}_i$, define the maximal and minimal EV (dis)charging power for the optimization horizon. If EV i is not connected at time t, then $\underline{x}_i(t) = \overline{x}_i(t) = 0$, else $\underline{x}_i(t) = x_i^{min}$ and $\overline{x}_i(t) = x_i^{max}$, where x_i^{min} and x_i^{max} are the minimal and maximal power that the EV charger supports.
For more details on how to derive these parameters, we refer the reader to [58].

For EV ADMM, the EV optimization problem in (4.3.7) is reformulated according to (4.2.32) into the following problem:

$$
\begin{aligned}
x_i^{k+1} = & \\
\text{minimize} \quad & \gamma \alpha_i ||x_i||_2^2 + \frac{\rho}{2} ||x_i - x_i^k + \overline{x}^k + u^k||_2^2 \\
\text{subject to} \quad & A_i x_i = R_i \\
& \underline{S}_i \leq B_i x_i \leq \overline{S}_i \\
& \underline{x}_i \leq x_i \leq \overline{x}_i .
\end{aligned}
\tag{4.3.8}
$$

4.4 Evaluation

In this section, we evaluate the performance of EV ADMM solving the EV aggregator optimization problems described in the previous section. We carry out two experiments for both problems. The first experiment evaluates the behavior of EV ADMM for different trade-off parameter γ values. The trade-off parameter balances the importance between the global aggregator objectives and the local objectives of the EV. This means that to ignore the individual EVs objective to minimize their battery depreciation cost γ can be set to zero. We also evaluate the effect of Vehicle-to-Grid (V2G) services, i.e., allowing EVs to feed energy back to the grid. The second experiment measures the scalability of EV ADMM for an increasing number of EVs and compares it to a centralized optimization approach.

Our experiments were carried out using a server with four Intel Xeon E5-2650 2GHz CPUs with four cores each, for a total of 16 cores and 64 GB of main memory. The implementations of EV ADMM, as well as the centralized optimization, were done in `MATLAB` [59] using `YALMIP` [60] to formulate the optimization problems and `GUROBI` [61] as solver.

We use publicly available data to define the optimization parameters. We consider a time horizon of 24 hours and a 15 minute time step duration resulting in $T = 96$ time steps. We obtained the demand profile D from the Munich distribution system operator website [62] and the energy price profile p for the minimal charging cost problem is from the European Energy Exchange (EEX) website [63]. Both correspond to Nov 21, 2011. Without the loss of generality, we assume that all EVs have a nominal battery capacity of 20 kWh and an average power consumption of 0.15 kWh/km. Using these parameters, we apply the method described in [9] to obtain different vehicle trip patterns from the National Household Travel Survey (NHTS). The NHTS dataset contains trips reported by more than 150,000 U.S. households [64]. Since the trip patterns describe if and where the vehicles are parked, we can define the time slots and locations for which an EV might charge. We assume that EVs only charge at home once per day. We also assume that the EVs want to fully charge their battery. The battery depreciation parameter is defined empirically as 0.0125 EUR/kW^2. The maximal (dis)charging rate for each EV is defined as

Valley filling parameters		
ε^{pri}	Primal feasibility threshold	20
ε^{dual}	Dual feasibility threshold	200
Charging cost minimization parameters		
$\underline{x}_a$	Aggregated maximal energy feed back	-137.6 kW
$\overline{x}_a$	Aggregated maximal consumption	137.6 kW
ε^{pri}	Primal feasibility threshold	0.1
ε^{dual}	Dual feasibility threshold	0.1

Table 4.4.1: Simulation parameters

$x_i^{max} = -x_i^{min} = 4\text{kW}$.

4.4.1 Experiment 1: Versatility

In this experiment, we explore the versatility of EV ADMM to solve the presented EV aggregator problems. We demonstrate the ability of EV ADMM to incorporate several aspects easily: In particular, battery depreciation costs and the ability of EVs to feed energy into the grid, also known as Vehicle-to-Grid (V2G) services. We consider a fixed number of 100 EVs for the valley filling and the charging cost minimization. The demand profile, as well as the maximal demand, were scaled for the number of EVs. The remaining simulation parameters we used in these experiments can be found in Table 4.4.1. The stopping threshold values are based on experience. The required number of EV ADMM iterations for each case can be seen in Table 4.4.2.

Valley filling: As shown in Fig. 4.4.1, valley filling is achieved if the individual EV goals are ignored, i.e., $\gamma = 0$. Moreover, allowing V2G services results in an almost flat aggregated demand profile. An all flat aggregated demand profile offers several advantages regarding grid operation; for example, the fact that with a flat valley no expensive peak power plants need to be turned on. Considering the behavior of EV ADMM, Table 4.4.2 reveals that the number of iterations remains of the same order,

	$\gamma = 0$	$\gamma = 1$
Valley filling iterations		
No-V2G	291	276
V2G	293	289
Price-basediterations		
No-V2G	889	665
V2G	1230	665

Table 4.4.2: ADMM required iterations for Experiment 1

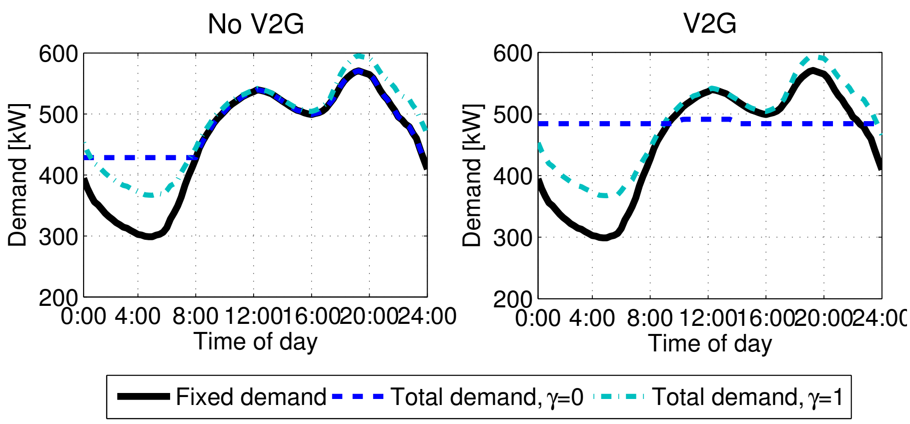


Figure 4.4.1: Valley filling optimization. Left without V2G and right with V2G for different individual EV goal importance γ.

regardless of V2G services or value of the trade-off parameter γ.

Charging cost minimization: In charging cost minimization, the aggregator modifies the energy price it gives to the EVs. Fig. 4.4.2 shows the adjusted energy price and the fleet behavior. This new optimal price corresponds to the incentive signal of EV ADMM upon convergence. We can see that the price only changes if the EVs meet the bound. We can also see that for our parameters, the EVs have no incentive to feed energy back to the grid if battery depreciation costs are considered. Regarding the EV ADMM behavior, Table 4.4.2 reveals a discrepancy in the iteration number for different trade-off values. This can be explained by the fact that for

$\gamma = 0$, we have an optimization problem with active global constraints. As seen in Fig. 4.4.2, for $\gamma = 0$, the bounds are active and for $\gamma = 1$, the bounds are not active. In EV ADMM, global constraints are considered by the aggregator, and a very exact solution of the original problem is required for these constraints to be fulfilled. Since the ADMM algorithm approximates the optimal solution more closely with each iteration, it requires more iterations to solve problems with active global constraint. This also explains the difference between no-V2G and V2G in Table 4.4.2. Fig. 4.4.2 shows that V2G has more active global constraints than no-V2G and consequently needs more iterations.

The results in the valley filling and the charging cost minimization problem for different values of the trade-off parameter γ show that its value has to be rather small to make the control approach useful on the system level, cf. [40]. Hence, if γ is too large, the aggregator losses control over the fleet. Therefore, it is advisable to allow the aggregator to define this parameter. It is important to point out that the charging requirements of the EVs are always fulfilled regardless of the value of γ, because the charging requirements are defined as constraints in the optimization problem. Any solution to the problem will always fulfill these constraints. When $\gamma = 1$ the EVs try to reduce their battery depreciation costs by charging constantly at lower charging rates. This reduces their level of flexibility, which reduces the capacity the aggregator has to control them.
The results also reveal that EV ADMM requires more iterations to solve the charging cost minimization problem than the valley filling problem. The reason for this is that the valley filling problem has a strictly convex objective function (4.3.1), whereas the charging minimization problem's objective function (4.3.4) is not *strictly* convex. In strictly convex problems, the ADMM penalty parameter ρ can be higher than for non-strictly convex problems, without the risk of becoming unstable. The valley filling is faster to solve because the penalty parameter determines the convergence speed of ADMM.

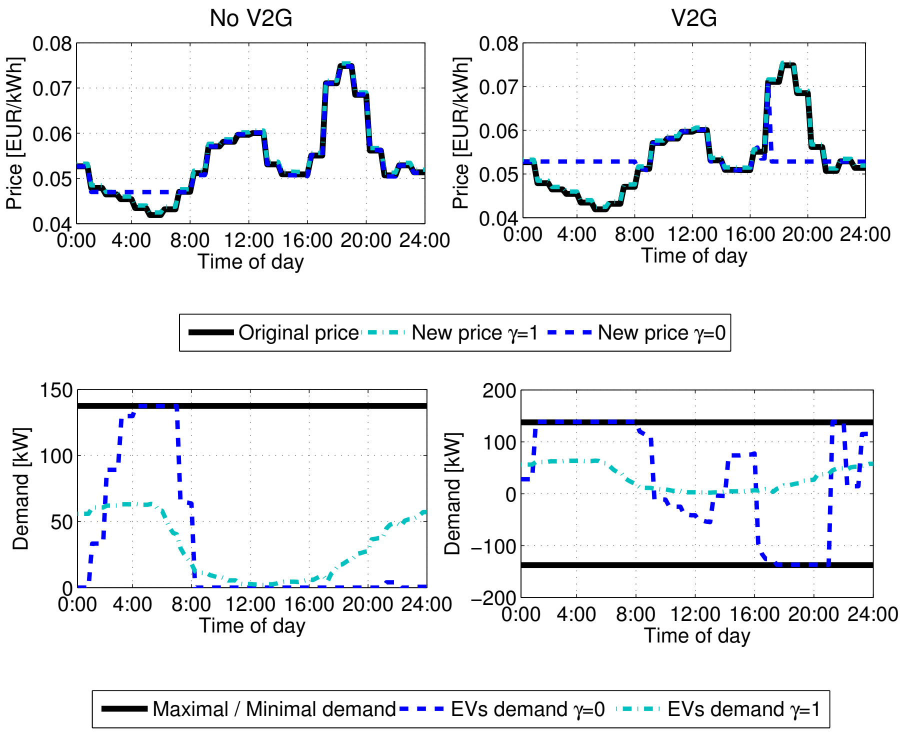


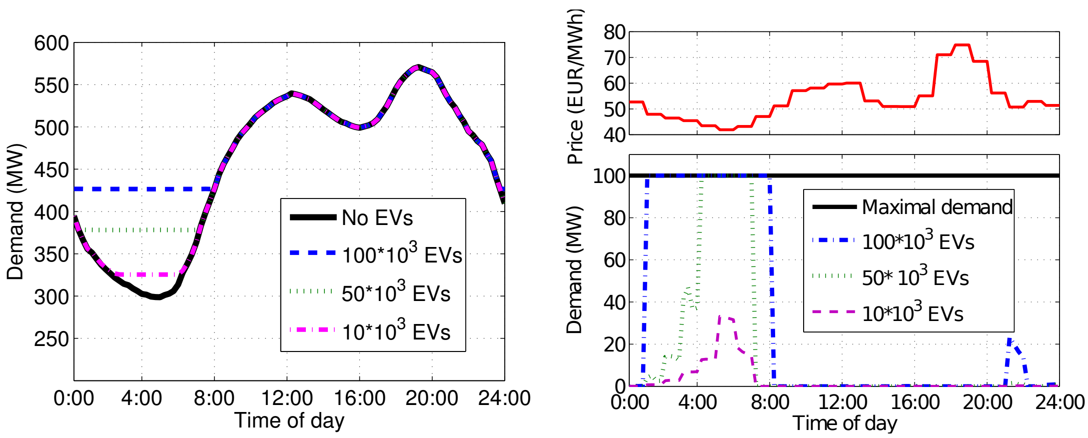
Figure 4.4.2: Charging cost minimization. Left without V2G and right with V2G for different individual EV goal importance γ.

4.4.2 Experiment 2: Scalability

We compare the scalability of EV ADMM's distributed optimization against a state-of-the-art centralized optimization. The number of electric vehicles serves as a sensitivity parameter to measure the required runtime and the peak memory use. For all considered cases, we assume that EVs are not allowed to discharge ($x_i^{min} = 0$) and that their individual objectives are ignored ($\gamma = 0$). We consider EV fleet sizes between 10,000 and 100,000 using an increment of 10,000 EVs. Table 4.4.3 shows the size of the problems regarding variables and constraints. For each fleet size, the simulation is repeated 5 times. The EV ADMM algorithm is stopped if the stopping criteria are reached or if the objective value does not decrease any further. The deviation of the final result between EV ADMM and the centralized optimization is less than 3 %. We implemented a serial and a parallel version of EV ADMM. In the

Problem	Valley filling	Charging cost minimization
Variables	96 + 96 × # EVs	
Constraints:		
-local equality	96 ×# EVs	96 *#EVs
-local inequality	2*96 ×# EVs	2*96 ×# EVs
-global inequality	0	96 *# EVs

Table 4.4.3: Problems sizes of the scalability experiment.

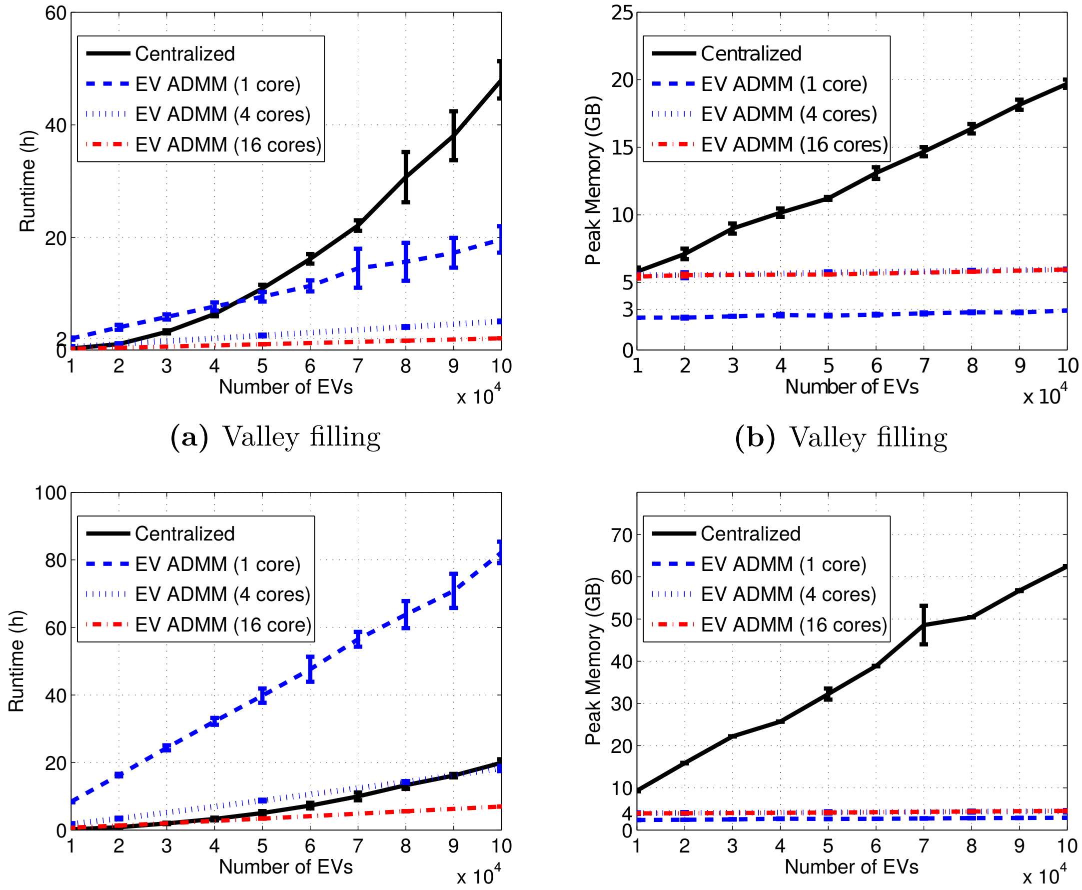


(a) Valley filling **(b)** Charging cost minimization

Figure 4.4.3: Scalability experiment optimization results.

serial implementation the subproblems are solved one at a time, i.e., only one core is used. In the parallel implementation, we consider the use of 4 and 16 cores, i.e., 4 or 16 subproblems are solved at the same time. For the parallel implementation, we make use of the `parfor` function of the `MATLAB` Parallel Computing Toolbox [65].

Fig. 4.4.3 and Fig. 4.4.4 show the experimental results for the valley filling and the charging cost minimization problems, respectively. The runtime results in Fig. 4.4.4a and Fig. 4.4.4c reveal that EV ADMM's runtime increases linearly with the number of EVs for both problems. This linear scalability is not surprising because it is a well-known characteristic of ADMM based algorithms [38, 51] . In contrast to this, the centralized optimization exhibits an exponential runtime increase. This gives EV ADMM a clear scalability advantage over the centralized approach. The advantage becomes even more apparent when we consider the effect of a parallel implementation with more cores. Fig. 4.4.5 shows the speedup achieved by parallel EV ADMM



(a) Valley filling

(b) Valley filling

(c) Charging cost minimization

(d) Charging cost minimization

Figure 4.4.4: Scalability experiments resulting runtime (left) and memory use (right).

over serial EV ADMM. We can see that a parallel EV ADMM implementation in 16 cores outperforms the centralized optimization for EV fleets with more 20 000 EVs. However, we can also see the speedup achieved by a multicore implementation is significantly lower than an estimation of dividing the solving time of serial EV ADMM by the number of cores. This can be explained by the Amdahl's law of speedup [66], which states that parallelization speedup is limited by the required sequential parts of a computer program. With a linear regression over our results, we can estimate a speedup of 55.7% per additional core provided in the case of valley filling, and 69.4% in the case of minimal cost charging. Moreover, the results for the memory use shown in Fig. 4.4.4b and Fig. 4.4.4d reveal that EV ADMM needs less memory to solve the problem than the centralized approach. This is because, in the

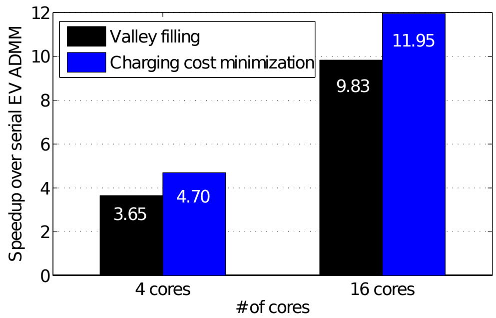


Figure 4.4.5: Parallel EV ADMM speedup

centralized approach, the parameters of all the subproblems need to be loaded into memory at the same time. Memory represents a barrier for centralized optimization that makes the optimization of very large problems intractable, because of memory capacity limitations. The advantage of EV ADMM is that the problem is split into subproblems and only the current subproblem being solved needs to be loaded into memory. Fig. 4.4.4b and Fig. 4.4.4d demonstrate that the memory increase is still lower than the memory required in a centralized optimization. Moreover, the memory does not increase significantly when more cores are used. Hence, EV ADMM allows the definition of a trade-off between running time and required computational resources (CPUs and memory).

To fully demonstrate the capabilities of EV ADMM, we solved the valley filling problem for 1 million EVs in Germany. The German demand data can be found in [63]. Due to memory limitations, this problem cannot be solved with our machine using a centralized approach. We waited a whole week (7 days) for the centralized method until a memory error caused the running process to crash. However, using EV ADMM (16 cores), we can solve this problem in less than 30 minutes, using less than 10 GB of peak memory. We solve the problem considering EVs that can charge with a maximal power of 4 kW and 20 kW. As shown in Fig. 4.4.6, the same quality of valley filling is achieved in both cases, regardless of the maximal charging power of the EVs. This was expected since the amount of required energy is the same in both cases.

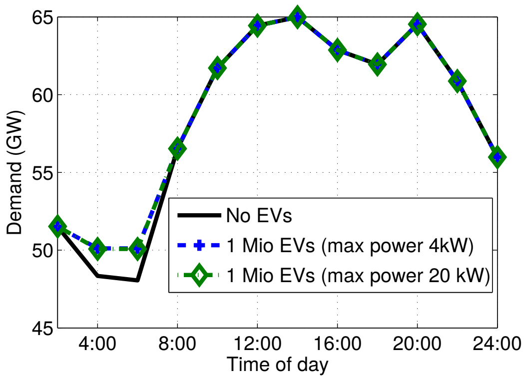


Figure 4.4.6: EV ADMM results for valley filling with 1 million EVs in Germany

4.5 Discussion

EV ADMM provides a scalable framework to solve convex EV aggregator optimization problems. In the following, we discuss its possible computational implementation and application areas.

4.5.1 Implementation

EV ADMM is a flexible algorithmic framework. Thus, the way it is implemented depends on the requirements and the available communication infrastructure.

Centralized computation

In this scenario, EV ADMM is used to parallelize the computations. A communication system is assumed, where all the required information for the optimization is communicated to a single centralized location, e.g., where the aggregator is located. In this scenario, the aggregator queries the charging requirements from all EVs, performs the optimization and sends the result back to the corresponding EVs. We demonstrated in Section 4.4.2 that EV ADMM can take advantage of the multi-core architecture of modern server processors to speed up computations. In practice, however, the centralized computation approach has two significant drawbacks. The first drawback comes from the technical challenges of implementing a centralized communication system that is both scalable and reliable. As the number of EVs increases, the aggregator needs to be able to handle an increasing number incoming messages. For very large EV numbers, the number of messages that the aggregator needs to handle could be too much overhead. The second drawback is the lack of data privacy. Since the charging requirements of all EVs are communicated to the aggregator, the aggregator eventually knows about the behavior of the individual EVs and consequently the behavior of their owners. This may lack broad acceptance by EV owners [8]. Therefore, a centralized system may not be the best solution. To distribute the computation and avoid a clustering of sensitive data, which EV ADMM supports out-of-the-box, may be the more promising approach.

Decentralized computation

Another possibility to implement EV ADMM is to leverage the decentralized nature of the algorithm and distribute the computation between the aggregator and EVs. In this scenario, as described in Fig. 4.3.1, the aggregator would broadcast the incentive signal and wait for the charging profiles to be calculated by each individual EV. Both parties would communicate iteratively until the convergence criteria is met and all final optimal EV charging profiles have been computed. The advantage of this procedure compared to the centralized approach is that no information about individual EV parameters is exchanged. Moreover, the messages are simpler, because the same incentive signal needs to be communicated to all EVs and the aggregator only requires the aggregated EV profiles to update the incentive signal. However, the problem remaining here is again the massive communication overhead, i.e., the number of messages that have to be received by the aggregator. Even worse than in the centralized setting, due to the iterative procedure of the algorithm, the aggregator has to handle the message overhead in each iteration. An extension of this approach can take advantage of an attractive characteristic of EV ADMM to reduce communication overhead. In EV ADMM, the aggregator only needs the aggregated EV profiles to update the incentive signal. If the aggregation was computed on the way from the EVs to the aggregator, the computation overhead would decrease and

the number of messages that needs to be handled would diminish. Furthermore, with this approach, the possibility to infer any sensitive information is reduced. The crucial requirement for this approach lies in a communication infrastructure that can provide aggregation as a central feature [67, 68]. This approach is discussed in Chapter 5.
Nonetheless, EV ADMM still requires a centralized location to calculate the incentive signal update. To remove this bottleneck, one can use standard solutions; in particular, a consensus algorithm to estimate the incentive signal update in a distributed way. This has been shown in related publications [69,70]. Nonetheless, one should carefully consider less communication intensive alternatives in practice. We believe that a distributed implementation of EV ADMM represents an interesting possibility, but one with many technical challenges that are currently being investigated.

4.5.2 Applications

The optimizations considered by EV aggregators can be carried out for planning or control tasks.

Planning tasks

In the case of planning, which can also be considered as open-loop control, the assumption is that the result of the optimization problem is calculated before the control horizon starts, e.g., one day ahead. This means that the time requirements for the optimization are not that strict, e.g., at least 10 hours. Our scalability evaluation in Section 4.4.2 has shown that a parallel implementation of EV ADMM is capable of solving such problems within the required time frame. Hence, EV ADMM already provides a solution for EV aggregator planning tasks that involve the solution of very large convex optimization problems. Even if some of the parameters like EV time of arrival and energy requirements are uncertain, the scalability and speed of EV ADMM allow for performing multiple optimizations runs considering different scenarios, cf. [30].

Control tasks

The implementation of EV ADMM for a closed-loop receding horizon control requires periodic solutions of the optimization problem, e.g., every 15 minutes. Although EV ADMM could fulfill these requirements for smaller fleets, for larger fleets the required computational resources may be prohibitive. Another aspect of EV ADMM that can be an issue for its use in a control context is the high number of iterations required, which translates into a high amount of messaging. For several iterations, the communication delays caused by a large number of messages can have a major effect on the solving time and stability of the algorithm. A promising approach to overcome this issue is presented in [42], where in-network aggregation is used to double the performance of a distributed EV optimization algorithm implementation. Another alternative for reducing the computation time and communication requirements is to accept suboptimal results. For this, we would need a distributed optimization algorithm that can be stopped and offer feasible suboptimal solutions before convergence (cf. [43]). The effectiveness of algorithms based on ADMM for treating uncertainty is demonstrated for 100 EVs in [30]. Nevertheless, it is clear that further research is required to efficiently apply EV ADMM for closed-loop control tasks with very large EV numbers.

Chapter 5

EV ADMM implementation in Publish/Subscribe Middleware

In this chapter, we study the use of publish and subscribe middleware to implement distributed EV charging optimization algorithms like EV ADMM in a distributed environment. We focus on the problem of an EV aggregator optimizing the day-ahead charging of an EV fleet to offer valley filling service.
First, we define our assumptions in Section 5.1. In Section 5.2, we review the basics of pub/sub middleware and outline the advantages it offers to solve our problem. In Section 5.3, we review three possible implementations of EV ADMM valley filing in a distributed environment using pub/sub middleware: centralized, decentralized and decentralized with aggregation. Then, in Section 5.4, we propose DOPS (Distributed Optimization Publish and Subscribe) an agent-based framework for the execution of distributed optimization algorithms. DOPS provides interfaces for multiple agents to execute optimization algorithms in a distributed infrastructure. We describe the different agent types and interfaces that DOPS offers. Finally, Section 5.5 presents our evaluation of EV ADMM in DOPS for the three EV ADMM implementation concepts. We compare the runtime of the three implementations taking into account computation and communication under real-world conditions. Our results demonstrate that a decentralized implementation can outperform a centralized implementation with increasing number of EVs. Moreover, our experiments show

that the in-network aggregation capability of DOPS doubles the performance of the decentralized approach. Our results, thus, confirm that distributed EV charging control infrastructures can benefit from using pub/sub middleware.

5.1 Assumptions

Since this chapter provides a proof of concept application of the results of the previous chapter, all the assumptions defined in Section 4.1 also apply here. Also, we make the following assumptions:

- We consider the real implementation of EV ADMM in a distributed environment.
- The parameters of the optimization problem are geographically dispersed in their particular locations.
- There is an event-based communication system that supports in-network aggregation.

5.2 Pub/sub middleware for EV charging control

Publish/subscribe middleware is based on a communication paradigm, where subscribing entities register their interest in an event, or a pattern of events, and are subsequently asynchronously notified of events that are issued by publishing entities [71]. The main advantage of this paradigm is that it allows for full decoupling of the communicating participants in space and individual control flow. This offers a lot of flexibility for the development of distributed applications. Loosely-coupled architectures as prevalent in publish/subscribe systems (pub/sub) [72, 73] can offer a flexible and reliable communication architecture. In this work, we exploit these advantages for the development of a platform to support distributed optimization for EV charging.

In (distributed) pub/sub, events or publications are matched against a set of filters or subscriptions, respectively, to route the information over a (distributed) broker architecture from a source to all interested sinks [72, 74]. Pub/sub could be a good candidate to serve as a messaging system between the EVs and the aggregator. However, standard the pub/sub model does not support the computation of in-network aggregates. This is a desirable feature for the efficient implementation of EV charging distributed optimization. In [75] a first approach to introduce aggregation for data-intense stream processing is proposed. In this work, we design and implement a similar approach to reduce the number of notifications that are exchanged in the broker overlay. Apart from reducing the number of messages sent over the communication network, the ability of this approach to decouple the aggregator and the individual EVs facilitates data anonymity, as the infrastructure only provides aggregated information. Furthermore, the distributed nature of the broker architecture enables the system to scale well with increasing number of EVs.

5.3 EV charging protocol implementation

We consider three different strategies to implement the EV ADMM valley filling in a distributed environment:

Centralized – An intuitive way to implement EV ADMM valley filling is in a centralized computation. In this case EV ADMM is used only to parallelize computations in a central location. In this approach, the aggregator queries the charging requirements from all EVs and calculates the charging schedules on its own. However, this approach has two major drawbacks. The first problem is the collection of all charging requirements from large numbers of EVs. Assuming a standard centralized communication model, the implementation of a scalable and reliable system becomes challenging. Especially, if there are real-time requirements to implement a model predicative style of control (cf., [40]). The second problem is not of technical nature, but highly relevant in practice: As charging requirements need to be exchanged in this approach, the aggregator eventually knows about the behavior of individual EVs and consequently the behavior of their owners. This resembles the concept of a direct control method, which may lack broad acceptance by society [8]. As a result, a centralized system may not be a practicable solution to the problem. Instead, to distribute the computation and avoid a clustering of sensitive data could be a more promising approach.

Decentralized – A second approach is to leverage the decentralized nature of the EV ADMM algorithm and distribute the computation between aggregator and EVs. In this implementation approach, the aggregator would broadcast the control signal and wait for the charging schedules to be computed by each EV. Both parties would communicate iteratively until the total demand profile is flattened to the maximal possible extent, and all final EV schedules are calculated. The advantage of this procedure compared to the centralized approach is that there is no need for exchanging any information about individual charging requirements. However, the problem remaining here is again the massive communication overhead, i.e., the number of messages that have to be received by the aggregator. Even worse than in the centralized setting, due to the iterative procedure of the algorithm, the aggregator has to handle the message overhead in each iteration and not only once. Hence, due to the massive communication cost, this solution might not be ideal.

Decentralized with aggregation – Our third approach takes advantage of an interesting characteristic of EV ADMM. To update the incentive signal, the aggregator needs the sum of intermediary schedules from all EVs. If this value were computed on the way from EVs to the aggregator, not only the computation overhead would decrease, but also the number of messages that need to be handled would diminish from a large amount to a single one containing already the necessary aggregated value. Furthermore, with this approach, the possibility to infer any sensitive information is reduced. The crucial requirement for this approach lies in a communication infrastructure that can provide aggregation as a central feature.

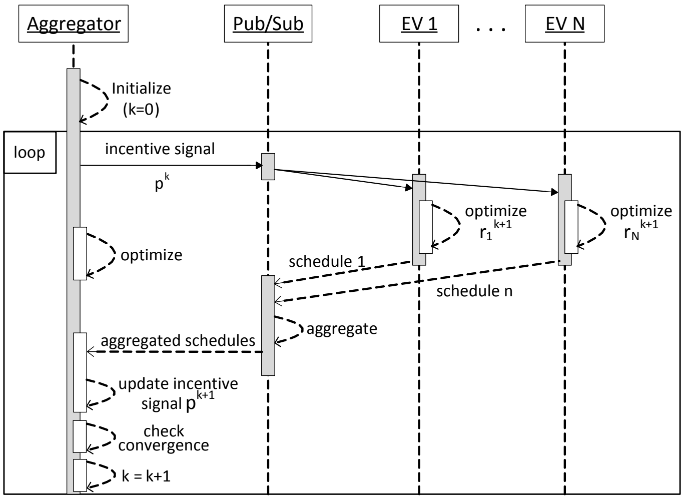


Figure 5.4.1: EV ADMM with Pub/Sub.

5.4 Distributed Optimization Publish and Subscribe

In this section, we describe DOPS, an agent-based framework for the execution of distributed optimization algorithms that follows the EV ADMM computational model depicted in Fig. 5.4.1. An agent in this context represents a particular role within the optimization process that has possibly thousands of instances (e.g., an EV). DOPS is implemented in JAVA and layered on top of the distributed content-based pub/sub middleware Padres [72], which abstracts from the communication between agents and other infrastructure capabilities (cf., Fig. 5.4.2). To reduce the message overhead regarding the number of messages being propagated and the overall communication delay, we developed an approach for in-network aggregation and added the corresponding feature to Padres. Aggregates are calculated over a synchronization context that resembles one iteration within the optimization process and can be any (self-) decomposable aggregation function [75] defined over application-specific data from all agents.

The DOPS framework provides two distinct agent types that allow for modeling different decentralized optimization approaches (cf., Fig. 5.4.3). Agents extend the pub/sub client for further functionality and essentially realize the iterative communication model that is depicted in Fig. 5.4.1. The *aggregator* agent acts as the authority to calculate and notify the incentive signal to the middleware (e.g., the

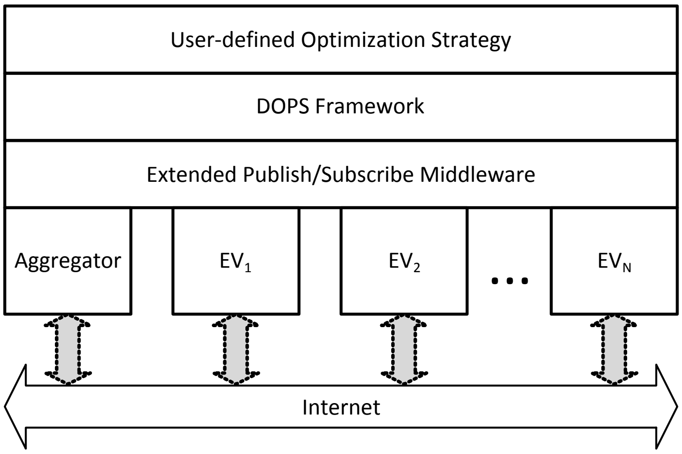


Figure 5.4.2: Conceptual overview of DOPS.

EV aggregator) and the *Controllable Device* agent (the EV) acts as an entity that optimizes based on the incentive signal and reports its local result to the middleware. The actual optimization procedure is encapsulated in a concrete optimization strategy that implements the *IOptimization Strategy* interface. This interface defines the complete optimization algorithm by specifying agent-specific code and a convergence criteria. In the following, we describe the functionality of these three interfaces. Altogether they represent everything an application developer has to implement to run own distributed optimization algorithms in a distributed infrastructure.

1. Aggregator Interface: Consists of three method stubs. These methods do not specify any optimization logic but are crucial for executing the optimization along an iterative workflow. The application developer has to specify the return values of these methods. For instance, the initialize() method loads the required data and getInitialIncentive returns the incentive. Some functionalities are already implemented, e.g., the invocation of the optimization methods from IOptimizationStrategy. A developer has to implement the methods of the interface IAggregator in a custom Aggregator class and the Aggregator parent class will do the rest.

2. Controllable Device Interface: Interface for devices similar to the aggregator interface providing three method stubs. Again, the primary purpose is to guarantee a well-defined optimization workflow.

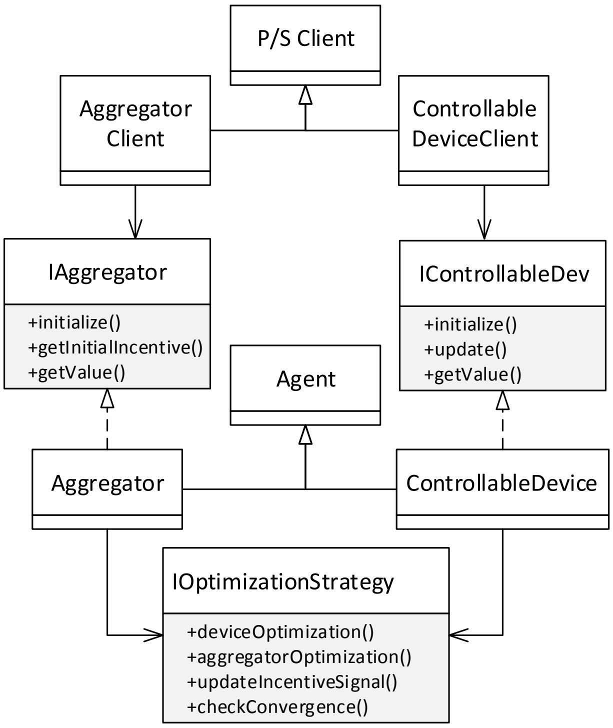


Figure 5.4.3: Relation of DOPS agents and P/S clients.

3. Optimization Strategy Interface: The actual optimization logic for both agent roles is specified in an implementation of the interface *IOptimizationStrategy*. This enables an application developer to specify the complete algorithm within a single Java class. Furthermore, it enhances flexibility as an optimization strategy can be exchanged during runtime without changing the agents.

5.5 Experimental evaluation

We evaluated our framework experimentally on a test cluster comprised of 15 virtual machines running Ubuntu 12.04.4 LTS (GNU/Linux 3.2.0-58-generic x86_64). Each machine was equipped with two vCPUs @ 2.6 GHz and 512 MB RAM. All machines were part of a VLAN and we did neither constrain latency nor bandwidth. To run DOPS we used Java version 1.7.0_51 and the Open JDK Runtime Environment

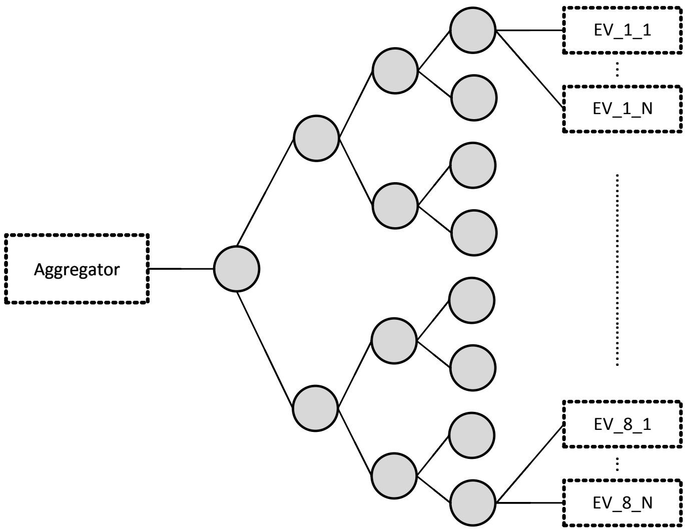


Figure 5.5.1: Experimental setup – circles are brokers withing the pub/sub overlay, boxes are DOPS agents. The dashed lines indicate different brokers to which the aggregator is connected to during experiments.

IcedTea v2.4.4.

Our general system configuration (i.e., overlay structure) is depicted in Fig. 5.5.1. Circles denote pub/sub messaging and aggregation brokers. The brokers form a balanced tree structure. The boxes denote the pub/sub clients (i.e., the agents in DOPS). The EVs (controllable device agents) are connected to edge brokers (i.e., the leafs in the broker tree) while the aggregator (aggregator agent) is connected to the root. In this setting, the aggregator runs on the same machine as the root broker. Similarly, EVs run on the same machines as the edge brokers. We implement the three different strategies described in Section 5.3 to solve the valley filling problem for EV fleets of variable size. We considered a one-day horizon (i.e., 24 hours) and a time interval $\Delta T = 15$ minutes. The base demand profile D was scaled depending on the number of EVs to guarantee that valley filling is achievable. The dataset used is publicly available and comprises of the base demand profile for the grid operator obtained from [62] and the EVs' energy requirements obtained using the methods described in [76].

Experiment 1 – Comparison of approaches: In the first experiment we compare the runtime performance of the centralized approach with both decentralized

approaches (i.e., with and without in-network aggregation). We use the total number of *controllable devices* as control parameter and run the optimization with 8, 24, 48, 72, 96, and 120 EVs. In this setting, EVs are equally distributed among edge brokers. For instance, for 24 EVs there are 3 EVs connected to each single edge broker (cf., Fig. 5.5.1). We take five measurements for each configuration and calculate the average of computation times. The results in Fig. 5.5.2 show the average runtime and its deviation. As we can see, the centralized approach performed best (about 2s for 72 EVs). The decentralized approach with aggregation (about 12s) outperforms the no-aggregation approach (about 28s). The reason for the centralized approach performing so well is that there is only a single round of communication necessary between aggregator and EVs. As opposed to the centralized approach, the distributed approaches communicate in an iterative manner. For instance, for 72 EVs the algorithm performs 27 iterations.
However, the drawback with the centralized approach is that (1) privacy of the data is not considered at all and (2) the aggregator has to handle a number of messages that is equal to the number of EVs. As this number increases, it becomes a bottleneck for the entire system, which the results for 120 EVs already indicate. Unfortunately, because of our limited computing resources, we were not able to further increase the number of EVs. Nevertheless, a power curve fitting of our measurements shown in Fig. 5.5.2 reveals that for 150 EVs the decentralized approaches will outperform the centralized approach. We can also see that the decentralized approach without aggregation suffers from the same bottleneck as the centralized approach. Again, the major drawback is that the number of messages an aggregator receives per iteration is equal to the number of EVs. By exploiting the aggregation capabilities of DOPS, this number can be reduced to only one message per iteration and performance is nearly doubled. Analyzing the trend of runtime behavior, we believe that there is a point close to the 120 EVs, at which the aggregation approach outperforms the centralized approach, i.e., according to our approximation around 140 EVs.

Experiment 2 – Random distribution of EVs: In the second experiment, we compare the runtime behavior for the decentralized approach with aggregation between an equal distribution of EVs among edge brokers and an unequal distribution. The total number of EVs is randomly distributed for each run of the optimization. The results of this experiment are shown in Fig. 5.5.3. As we can see, for up to 48 EVs in

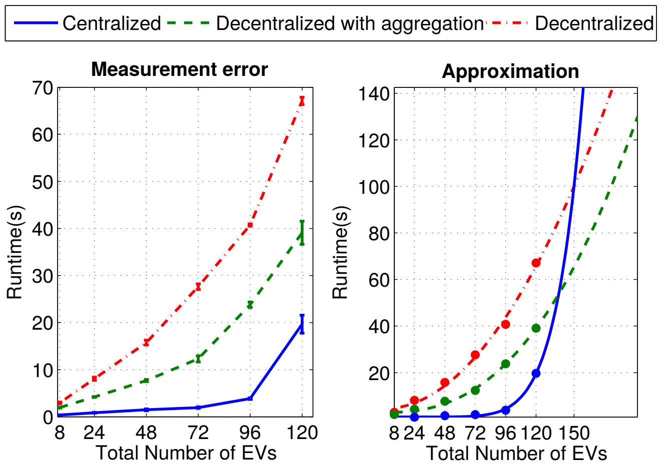


Figure 5.5.2: Experiment 1 – Runtime comparison: Centralized vs. no aggregation vs. in-network aggregation.

total, the difference in runtime between equal (7.7 seconds) and random distribution (8 seconds), remains comparable. For 72 EVs a more the equal distribution performs better, and for 120 EVs the runtime is halved. This behavior can be explained by the increased load that occurs at some of the edge brokers due to the higher amount of EVs they have to handle.

Interpretation – In conclusion, we can say that with our aggregation approach for pub/sub, the distributed optimization performance is significantly improved. In our experiments, the runtime was halved. Analytically, the number of messages that arrive at the aggregator client is reduced to a single one per iteration. This becomes relevant when we consider the introduction of EV aggregators that will have to control thousands or even millions of EVs. Unfortunately, due to the resources constraints in our testbed, especially the limited memory of 512 Mb per machine, we were not able to extend our experiments to larger EV fleet sizes. However, using extrapolation, we can predict that already for 150 EVs the decentralized approaches will gain the upper hand. Our results demonstrate that a decentralized EV charging control infrastructure can benefit from using pub/sub with aggregation capabilities.

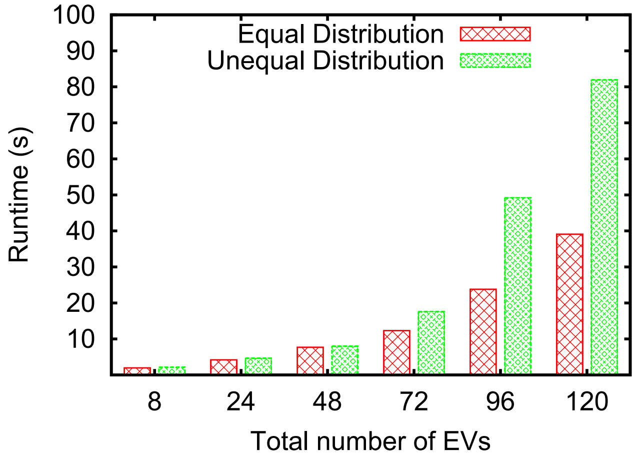


Figure 5.5.3: Experiment 2 – Runtime comparison (aggregation) between equal and unequal (i.e., random) distribution of EVs among edge brokers.

Chapter 6

Distributed Optimization for EV Charging Congestion Control with EV NUM

In this chapter, we develop a real-time distributed anytime EV charging congestion control algorithm to support the integration of EVs into the current distribution grid. Our approach is based on a primal decomposition solution of the EV NUM problem. This results in an EV congestion control protocol with closed-form expressions for the computations performed by EV chargers and protection devices. Our protocol maximizes the EVs' utilization of the available infrastructure while respecting its operational constraints.

First, we declare our assumptions in Section 6.1. In Section 6.2, we shortly review the problem of EV congestion control, define its mathematical formulation as the EV NUM problem, and present its distributed optimization solution using the state-of-the-art dual decomposition and our proposed primal decomposition. We explain how the dual solution algorithm results in an incentive-based control approach and our primal solution algorithm results in a budget-based control approach. We also discuss the theoretical convergence behavior of both approaches and demonstrate the theoretical advantages of our primal approach. Then, in Section 6.3 we define the specific algorithms that the individual EVs and protection devices must follow to implement the EV NUM dual and primal congestion control protocols. Section 6.4 shows our evaluation experiments for both approaches under static and dynamic conditions using the IEEE 13 Node Test Feeder. We show that the theoretical insights match our experimental results and demonstrate that, thanks to its anytime property, the primal approach offers a superior convergence behavior and control performance than the dual approach. Section 6.5 presents a more realistic and comprehensive evaluation of both approaches using the IEEE European Low Voltage Test Feeder. We perform a scalability evaluation of both methods and use a 3-phase power flow simulations to assess the real impact of both algorithms on the operational constraints of the grid. The results demonstrate that our primal approach outperforms the dual approach in scalability. Moreover, the evaluation shows that the dual approach can allow faster charging at the cost of possible operational constraint violations, while our primal approach does not violate operational constraints. Finally, Section 6.6 presents a short discussion on the applicability of the EV NUM formulation and both its solution algorithms.

6.1 Assumptions

To formulate the EV congestion control problem as an optimization problem we assume the following:

- EVs start charging immediately after arrival.
- EVs want to maximize their charging rate.
- Denial of service or stopping an EV from charging is not allowed.
- EVs have no deadline to finish charging. If EVs have a deadline, we do not guarantee that they will charge to their desired level.
- Distribution grid devices have a maximal overloading capacity that must be respected.
- The time to stop overloading of devices is in the millisecond range.

- Predictions are not available.
- The utility company guarantees operational voltage constraints without the control of EVs.
- For the problem formulation losses are ignored, and voltage is considered to be constant. However, a 3-phase AC power flow model is considered for our evaluations in Section 6.5.
- There is a communication network that is ubiquitous, broadband, reliable, and has a low latency.
- The problem formulation is convex. An important consequence of this assumption is that the EVs must be able to use a charging rate that is continuous between 0 and a maximal value.

6.2 Real-time EV charging congestion control

We consider the problem of controlling the charging rate of EVs in a distribution network. The distribution network consists of several devices such as lines, transformers, and switches, all of which have a maximal loading rate also known as ampacity. An overloading or congestion of the grid happens when the load at any of these devices is higher than the maximal allowed loading. Most of these devices are equipped with a protection device, which prevents equipment damage by opening the electric circuit whenever the maximal loading is reached. Protections have a reaction time of hundreds of milliseconds, and many of them already have measuring, computation, and communication capabilities [77]. EV home chargers are also expected to have computation and communication capabilities. Hence, the overloading problem resulting from large EV populations can be solved using intelligent charging as shown in Fig. 6.2.1.

We follow a best-effort approach without considering predictions. With this approach, the problem becomes a time critical problem, in which fast reactions are required to prevent the overloading of grid devices or power outages. We divide time into

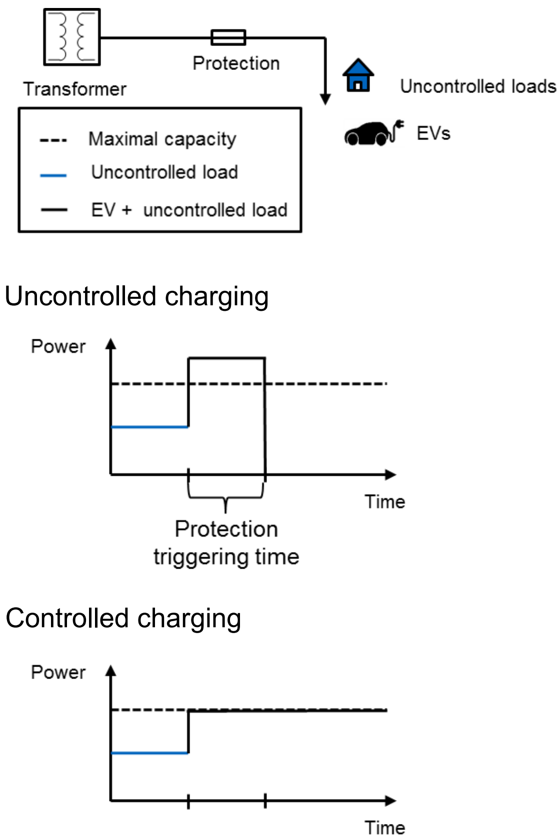


Figure 6.2.1: Uncontrolled and controlled charging.

control time slots of duration T_c. We assume that the state of the power system remains constant during each time slot. This also means that the time required for computation and communication has to be smaller than the triggering time of the protection devices. Furthermore, we assume that each EV wants to charge as fast as possible. During each control time slot, an optimization problem needs to be solved. The objective of this problem is to maximize the charging rate of EVs, such that the capacity of the grid is optimally used and all EV chargers have access to the grid resources.

Although the real-time EV charging optimization problem can be solved centrally, we strive for a distributed solution. In a centralized approach, the parameters of each EV charger and the parameters of each protection device would need to be communicated to a centralized location, where the optimization is performed. The result would then have to be sent back to the EVs. For a large distribution network, the communication and computation overhead of this procedure would not allow us to provide a solution within the required timeframe: the control delay in a centralized solution would be too long to prevent locally triggered faults. Therefore, it is more suitable to keep the computation close to the data, i.e., to allow each device to make calculations based on its local data and use message passing to coordinate them to solve the original optimization problem.

6.2.1 The EV NUM optimization problem

We formulate the real-time EV charging control problem as a NUM problem (cf. [78]), where the objective is to maximize the utility of the network's users while respecting the users' and the network's constraints. In our case, the users are the EV chargers, and the network constraints are set by the maximal available loading of the protection devices.

Let $\mathcal{V} = \{1, \ldots, N\}$ denote the set of EV chargers and $\mathcal{P} = \{1, \ldots, M\}$ the set of protection devices. Our optimization variables are the EV charging rates x_i for $i \in \mathcal{V}$. The energy flow to each EV charger traverses several protection devices before reaching its destination. We define this as the *route* to the EV charger. The set of protection devices in the route to charger $i \in \mathcal{V}$ is denoted by $\mathcal{P}_i$ and the set of charger routes leading through protection device $l \in \mathcal{P}$ is denoted by $\mathcal{V}_l$. With this, the EV NUM problem is written as follows:

$$\underset{x}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} - w_i log(x_i) \tag{6.2.1}$$

$$\text{subject to} \quad 0 \leq x_i \leq \overline{x}_i \quad \forall i \in \mathcal{V} \tag{6.2.2}$$

$$\sum_{i \in \mathcal{V}_l} x_i \leq c_l \quad \forall l \in \mathcal{P}, \tag{6.2.3}$$

$$\underset{x}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} - w_i log(x_i) \tag{6.2.4}$$

$$\begin{aligned} \text{subject to} \quad & 0 \leq x_i \leq \overline{x}_i \quad \forall i \in \mathcal{V} && (6.2.5) \\ & \sum_{i \in \mathcal{V}_l} x_i \leq c_l \quad \forall l \in \mathcal{P}, && (6.2.6) \end{aligned}$$

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i \in \mathcal{V}} - w_i log(x_i) && (6.2.7) \\ \text{subject to} \quad & 0 \leq x_i \leq \overline{x}_i \quad \forall i \in \mathcal{V} && (6.2.8) \\ & \sum_{i \in \mathcal{V}_l} x_i \leq c_l \quad \forall l \in \mathcal{P}, && (6.2.9) \end{aligned}$$

where $\overline{x}_i$ is the maximal charging rate of EV charger i and c_l is the maximal available loading capacity of each protection device l. The value function (6.2.7) is the sum of the EV chargers' utility functions. The weighting parameters $w_i > 0$ can be used to set priorities, e.g., if an EV pays a higher price for charging it should have a higher priority, or if an EV is almost fully charged, its priority can be reduced.

The logarithmic utility function makes sure that no EV charger is denied service, i.e., $x_i \neq 0$. If this happened to any EV charger, its value function cost would be infinite. The first set of constraints (6.2.8) are the minimal and maximal charging rate for each EV charger. The second set of constraints (6.2.9) defines the maximal available loading of each protection device, i.e., the maximal loading that each device supports minus the current loading used to supply the customers with uncontrollable demand.

The objective function is separable because each EV charger can compute its local objective using only its local optimization variable x_i. The constraints in (6.2.8) are also separable because they also can be checked by each EV using only its local optimization variable. However, the constraints in (6.2.9) couple the local optimization variables of each EV charger, because one EV charger requires the local optimization variables of other EV chargers. These coupling constraints prevent us from distributing the computation across the EV chargers and protection devices. Hence, the problem is not separable.

The EV NUM problem is *convex*, since its cost function is convex and its constraint space also convex. Thus, there exists only one optimal point x^* that solves our problem [46].

6.2.2 State-of-the-art: Incentive-based control

The distributed optimization solution of the EV NUM problem using dual decomposition has been studied in [36]. The goal of this approach is to allow each EV charger to optimize its charging rate independently using only local information and an incentive value generated by the protection devices. The result is an iterative distributed algorithm that generates the optimal incentives to solve the original problem described in Section 6.2.1. Since these incentives control the EVs, we refer to this approach as an *incentive-based control*. This work provides a derivation of the incentive-based algorithm that is an alternative to the one presented in [36].

To formulate the dual decomposition solution we use the dual ascent method [47]. The problem's Lagrangian is defined as:

$$L(x, \lambda) = \sum_{i \in \mathcal{V}} - w_i log(x_i) + \sum_{l \in \mathcal{P}} \{\lambda_l (\sum_{i \in \mathcal{V}_l} x_i - c_l)\}, \tag{6.2.10}$$

where $\lambda = [\lambda_1, \ldots, \lambda_M]^T$ is a vector of Lagrangian multipliers for each coupling constraint, also known as dual variables. By reordering the terms in the Lagrangian, we see that our Lagrangian is the sum of individual EV charger Lagrangians:

$$L(x, \lambda) = \sum_{i \in \mathcal{V}} L_i(x_i, \lambda) =$$
$$\sum_{i \in \mathcal{V}} \left\{ -w_i log(x_i) + \sum_{l \in \mathcal{P}_i} \{\lambda_l (x_i - c_l) \right\}. \tag{6.2.11}$$

The dual function of our problem is then defined as:

$$G(\lambda) = \sum_{i \in \mathcal{V}} \left\{ \min_{0 \leq x_i \leq \overline{x}_i} \{-w_i log(x_i) + \sum_{l \in \mathcal{P}_i} \{\lambda_l (x_i - c_l)\} \right\}. \tag{6.2.12}$$

With the dual function, the dual problem is written as:

$$\underset{\lambda \geq 0}{\text{maximize}} \quad G(\lambda). \tag{6.2.13}$$

Since our original problem is convex, strong duality holds [46]. This means, that the optimal values of the original problem and the dual problem are the same. To recover the optimal result x^* from the dual optimal point λ^*, we have to solve:

$$x^* = \min_{0 \leq x \leq \overline{x}} L(x, \lambda^*). \tag{6.2.14}$$

For this we use the gradient method. Since G is differentiable, its gradient, $\nabla G(\lambda)$, is evaluated as follows:

$$\nabla G(\lambda) = \sum_{l \in \mathcal{P}} \{\lambda_l (\sum_{i \in \mathcal{V}_l} x_i - c_l)\}. \tag{6.2.15}$$

We see in (6.2.10) and (6.2.11) that λ and x are both separable. Therefore, we can split the minimization with respect to x into N separable problems and the maximization with respect to λ into M separable problems. The dual ascent algorithm is then given as follows:

$$x_i(k+1) = \min_{0 \leq x_i \leq \overline{x}_i} L_i(x_i, \lambda(k)) \tag{6.2.16}$$

$$\lambda_l(k+1) = \left[\lambda_l(k) + \kappa(\sum_{i \in \mathcal{V}_l} x_i(k+1) - c_l)\right]^+, \tag{6.2.17}$$

where k is the iteration index, $\kappa \geq 0$ is the step size of the gradient method, and the function $f(a) = [a]^+ = \max\{a, 0\}$, is a projection onto the positive orthogonal. The first step (6.2.16) is an x minimization step and the second step (6.2.17) is a dual variable update. The dual variables can be interpreted as a vector of prices. Thus, step (6.2.17) is basically a price update.

The x minimization step (6.2.16) has an analytic solution. The solution is obtained by setting the partial derivative of (6.2.11) with respect to x_i to zero, solving for x_i, and then projecting this value onto $0 \leq x_i \leq \overline{x}_i$. For more details on projections see [46]. With this solution, we write the dual decomposition algorithm for our problem as:

For each EV charger, $i = 1, \ldots, N$:

$$x_i(k+1) = \min \left\{ \frac{w_i}{\sum\limits_{l \in \mathcal{P}_i} \lambda_l(k)}, \overline{x}_i \right\}. \tag{6.2.18}$$

And for each protection device, $l = 1, \ldots, M$:

$$\lambda_l(k+1) = \max \left\{ \lambda_l(k) + \kappa(\sum_{i \in \mathcal{V}_l} x_i - c_l), 0 \right\}. \tag{6.2.19}$$

We see in (6.2.18) that each EV defines its charging rate based on the sum of the prices set by the protections on its route. Thus, each protection along the EV charger's route needs to communicate this price to the EV charger. In (6.2.19), we see that each protection device does a price update based on the current available loading capacity, which is the maximal available capacity, c_l, minus the sum of the chargers loading. This can be measured by the protection device locally. Thus, there is no need for the EV chargers to communicate their charging rates to the protection devices. This property enables the one-way communication design proposed in [36].

The main drawback of the EV NUM dual decomposition solution is that the coupling constraint $\sum_{i \in \mathcal{V}_l} x_i - c_l$ can be violated while the optimization algorithm is running: The constraint is sure to be satisfied only upon convergence, which means that the protection devices may already be triggered in the meantime. In [36] this is addressed by introducing an emergency response mode in which the charging of all EVs is interrupted if the protection device is about to be triggered. However, as we will later see, due to the highly dynamic nature of the fixed loads in the grid and the timescales on which the proposed control operates, the emergency response strategy may result in a prolonged period during which EV charging is not possible. Another drawback of this algorithm is that its performance and convergence strongly depend on the step size κ. For certain values of κ, the algorithm may not converge or may become unstable. This instability leads to high oscillations in the computed EV charging rates, and in turn to a fault in the protection devices.

Convergence: The convergence and stability of the dual decomposition algorithm for the NUM problem have been studied in [16]. In [36], it is confirmed for real-time EV charging control.
The incentive-based approach converges when a price equilibrium has been reached, i.e., when the prices don't change significantly in subsequent rounds:

$$||\lambda(k+1) - \lambda(k)||_2 \leq \epsilon_i, \tag{6.2.20}$$

where $\epsilon_i > 0$ is the convergence parameter. The convergence of the algorithm is guaranteed if:

$$0 < \kappa \leq \frac{2}{\overline{x}_{max}\overline{L}\overline{N}}, \tag{6.2.21}$$

where $\overline{x}_{max} = \max \overline{x_i}$, is the maximal charging rate across all EV chargers, $\overline{L} = \max_i \sum_l R_{li}$, is the maximal number of protection devices that any route to an EV charger goes through, and $\overline{N} = \max_l R_{li}$, which is equivalent to the number of EV chargers.

The evaluations in [36] showed that a higher value of κ leads to faster convergence. However, as seen in (6.2.21), the value of κ needs to be smaller for a higher number of EV chargers and a larger grid. Hence, the larger the system, the longer the algorithm takes to converge.

6.2.3 Novel approach: Budget-based control

To improve on the drawbacks of the EV NUM dual solution, we propose a distributed anytime primal algorithm. The idea is to allow each EV charger to optimize its charging rate independently using local information, but to stay below a budget defined by the protection devices. When we modify the original problem to include budgets and apply primal decomposition to it, the result is an iterative distributed algorithm that generates the optimal budgets for each EV charger. The main advantage of this approach is that the resulting algorithm has the *anytime property*. This means that the charging rates produced by the algorithm with each iteration are always feasible. Moreover, the longer the algorithm runs, the closer the result gets to the optimal solution. Since the control of the EVs is based on setting maximal upper bounds on its charging rate or budgets, this control approach can be characterized as a *budget-based control*.

To formulate our approach, we first modify the original EV NUM problem in Section 6.2.1 to include a budget for each EV charger. Let $b = [b_1, \ldots, b_N]^T$ denote a vector of the budgets for each EV charger $i \in \mathcal{V}$. Then, the problem can be written as follows:

$$\underset{x,b}{\text{minimize}} \quad \sum_{i \in \mathcal{V}} - w_i log(x_i) \tag{6.2.22}$$

$$\text{subject to} \quad 0 \leq x_i \leq \overline{x}_i \quad \forall i \in \mathcal{V} \tag{6.2.23}$$

$$x_i \leq b_i \quad \forall i \in \mathcal{V} \tag{6.2.24}$$

$$\sum_{i \in \mathcal{V}_l} b_i \leq c_l \quad \forall l \in \mathcal{P}. \tag{6.2.25}$$

In the formulation above we see that the budgets are included as optimization variables. Moreover, in Equation (6.2.24) we see that each EV budget represents an upper bound constraint for its corresponding EV charger. Since the budgets are to be defined by the protection devices, in Equation (6.2.25) the maximal available loading constraints have been modified to depend on the budgets instead of the EV charging rates. This formulation delivers the same optimal results as the original problem in Section 6.2.1. As we proofed in [43]: If an EV charger cannot reach its maximal charging rate $\overline{x}_i$, the value for its charging rate becomes its budget, i.e., $x_i = b_i$. Hence, the modified problem becomes equal to the original problem.

In a distributed optimization scheme, each EV charger i optimizes using only its local decision variable, x_i. In Section 6.2.1, we explained that our original problem is not separable due to the coupling constraints. However, in our new formulation, the coupling constraints depend only on the value of the budgets. Thus, if the budgets are fixed values, the problem becomes completely separable. Therefore, we can think of the budgets as interface variables between the individual problems of each EV charger.

Our new formulation can be represented as the sum of individual EV charger problems, which are coupled by the budget variables:

$$\begin{aligned} \min_{b} \quad & \sum_{i\in\mathcal{V}} \left\{ \max_{\mu_i\geq 0} \min_{x_i} \{-w_i log(x_i) + \mu_i(x_i - b_i)\} \right\} && (6.2.26) \\ \text{subject to} \quad & 0 \leq x_i \leq \overline{x}_i, \quad \forall i \in \mathcal{V} && (6.2.27) \\ & \sum_{i\in\mathcal{V}_l} b_i \leq c_l, \quad \forall l \in \mathcal{P}, && (6.2.28) \end{aligned}$$

where μ_i is the Lagrangian variable from constraint $x_i \leq b_i$. The requirement $\mu_i \geq 0$, comes from the Karush Kuhn Tucker (KKT) conditions (cf. [46]). We separate this problem into a set of subproblems for each EV charger that optimizes the local variables x_i and a master problem that optimizes the interface variables b_i. Let $\phi_i(b_i)$ denote the optimal value for the following problem:

$$\max_{\mu_i\geq 0} \min_{0\leq x_i\leq \overline{x}_i} \quad \{-w_i log(x_i) + \mu_i(x_i - b_i)\}. \tag{6.2.29}$$

With that definition, the master problem is formulated as:

$$\begin{aligned} \min_{b} \quad & \sum_{i\in\mathcal{V}} \phi_i(b_i) \\ \text{subject to} \quad & \sum_{i\in\mathcal{V}_l} b_i \leq c_l, \quad \forall l \in \mathcal{P}, \end{aligned} \tag{6.2.30}$$

with $\phi_i(b_i) = -\mu_i b_i$.

The master problem can be solved with an iterative method, e.g., gradient projection. Each iteration requires solving the subproblems (6.2.29) in order to evaluate the master problem (6.2.30) (cf. [48]).

The subproblem of each EV charger (6.2.29) has an analytic solution. This solution is obtained by applying the KKT conditions and solving the equations after the optimization variables. The result for each EV charger $i = 1, \ldots, N$ is:

$$x_i(k+1) \quad = \quad \min\{b_i(k), \overline{x}\} \tag{6.2.31}$$

$$\mu_i(k+1) = \begin{cases} 0 & \text{if } x_i(k+1) = \overline{x} \\ w_i/x_i(k+1) & \text{otherwise} \end{cases}, \qquad (6.2.32)$$

where k is the iteration index. These computations can be done in parallel by each EV charger once it knows its current budget $b_i(k)$. The resulting optimal value for the Lagrangian variable μ_i can be interpreted as the marginal benefit of the EV charger for the assigned budget. In Equation (6.2.32), we see that the marginal benefit is zero when $x_i(k+1) = \overline{x}$. This is because at this point the EV has achieved its maximal benefit. In turn, when the EV charger's maximal benefit cannot be achieved, its Lagrangian is the derivate of its utility function, which is the textbook definition of the marginal benefit. Hence, $\mu_i = \nabla U_i(x_i) = w_i/x_i$.

To update the budgets, we need to solve the master problem in (6.2.30). We could use the gradient projection method explained in [79]. However, this approach would require the current available loading c_l of all protection devices to be sent to a central location in each iteration. In a large distribution grid, this would cause a large communication overhead. Therefore, we propose using the *sequential projections method* described in [48] together with gradient descent, which results in a *gradient sequential projection method*. Our approach consists in projecting the budget updates onto the individual constraints of each protection device as they are communicated along the network back to the EVs.

For a sequential projection, we first need a communication node where the EVs send their current budgets $b_i(k)$ and their marginal benefits $\mu_i(k+1)$. In the following, we assume that this central communication node is located at the substation or the root node of the network. Then, we sequentially project the gradient updates of these budgets onto the individual constraint set of each protection device along the corresponding route. We formalize this as:

$$b(k+1) = \mathbf{P}_{\mathcal{C}_M}\{\ldots \mathbf{P}_{\mathcal{C}_2}\{\mathbf{P}_{\mathcal{C}_1}\{b(k) + \alpha\mu(k+1),\}\}\ldots\}, \qquad (6.2.33)$$

where $\mathbf{P}_{\mathcal{C}_l}$ is the projection operator over the set:

$$\mathcal{C}_l = \{b | \sum_{i \in \mathcal{V}_l} b_i \leq c_l\}. \tag{6.2.34}$$

Equations (6.2.33) and (6.2.34) describe a procedure that updates the budgets as they are communicated along the protection devices back to the EV chargers. It loops over all protection devices, $l \in \{1, \ldots, M\}$. We define the starting value of the budgets as the gradient update:

$$b^0 = b(k) + \alpha\mu(k+1). \tag{6.2.35}$$

Then, the problem to solve for each protection device is:

$$b^l = \mathbf{P}_{\mathcal{C}_l}\{b^{l-1}\}. \tag{6.2.36}$$

This projection is an optimization problem:

$$\begin{array}{ll} \min_{b^l} & ||b^l - b^{l-1}||_2^2 \\ \text{s.t.} & b^l \in \mathcal{C}_l. \end{array} \tag{6.2.37}$$

The problem above is solved using the KKT conditions. It represents a projection onto a halfplane, which has an analytic solution, (cf. [46]). Hence, to solve the master problem, each protection device updates the budget of each EV charger route that goes through it, i.e., $\forall i \in \mathcal{V}_l$:

$$b_i^l = \begin{cases} b_i^{l-1}, & \text{if } \sum\limits_{i \in \mathcal{V}_l} b_i^{l-1} \leq c_l \\ b_i^{l-1} + \frac{(c_l - \sum\limits_{i \in \mathcal{V}_l} b_i^{l-1})}{N_l}, & \text{otherwise} \end{cases}, \tag{6.2.38}$$

where N_l is the number of EV routes that go through protection device l, defined as $N_l = \sum\limits_{i \in \mathcal{V}_l} 1$.

The resulting solution approach is a distributed algorithm, where the EV chargers define their charging rate and marginal benefit according to Equations (6.2.31) and (6.2.32). These values are then sent to the central communication node. There, the values are used to perform a gradient update of the budgets according to Equation (6.2.35). Afterwards, the updated budget values are propagated down through the protection devices back to the EVs. While they are propagated, each protection device projects them onto their constraint set (6.2.38). Once the budgets have been propagated through the network back to the EV chargers, a new iteration starts. This procedure guarantees that the resulting budgets are feasible for all protection devices in each iteration. Moreover, due to the gradient step update, we also close in on the optimal solution of the master problem in each iteration until it is finally reached.
Unlike the incentive-based approach, which requires only one-way communication from the protection devices to the EV chargers, the proposed dynamic budget approach requires two-way communication between protection devices and EV chargers. However, as we will later see, this drawback is more than compensated by its advantages.

Convergence: Our primal algorithm converges when the update of the budgets does not change significantly:

$$||b(k+1) - b(k)||_2 < \epsilon_b,$$

where $\epsilon_b > 0$ is the convergence parameter of our approach.

Our approach uses the gradient projection algorithm to solve the master problem (6.2.30) of the primal decomposition. As explained in [79], the gradient algorithm converges to the optimal solution if:

$$0 < \alpha \leq \frac{2}{K}, \tag{6.2.39}$$

where K is the Lipschitz constant, which is defined as the maximal absolute value of the master problem's cost function derivative: $|\sum_{\mathcal{V}} \nabla \phi_i(b_i)| = \sum_{\mathcal{V}} \mu_i \leq K$. From our definition of the marginal profit, $\mu_i = 1/x_i$, we see that if $x_i = 0$, then $\mu_i = \infty$. This means that our cost function is not Lipschitz. To overcome this, we can define an upper bound for the value of μ or we can avoid assigning zero budgets to any EV charger. For our evaluation, we assigned an upper bound value of $\overline{\mu} = 10^{10}$. However, even if $\alpha > 2/K$ our algorithm still converges to produce a feasible result. The convergence value is simply further away from the optimal value.

Furthermore, the algorithm remains stable even when the weight of the utility function changes over time, provided that the changes take place after the algorithm has converged, i.e., when the time between weight changes is larger than the algorithm's convergence time. Changing the weight values over time is important for defining priorities between EVs. The weights can be used to give more or less priority to EVs based on their state, e.g., remaining connection time or battery state of charge.

We provide a more comprehensive mathematical examination on the convergence of this algorithm in [43].

6.3 Algorithms implementation

We now describe the algorithms that the protection devices and the EV chargers implement in the incentive- and budget-based approaches. As in [36], we define the control time slot duration as $T_c = 20$ms. We also assume that the algorithms perform one iteration with each clock tick. Therefore, the computation and the communication for one iteration has to be performed within this time period. This assumption results in the following requirement:

$$T_c > d, \tag{6.3.1}$$

where d is the delay caused by computation and communication.

6.3.1 Incentive-based algorithm

The algorithms for the incentive-based approach defined in [36] are based on the results discussed in Section 6.2.2. As defined by Equation (6.2.19), each protection device performs a price update based on its measured congestion states and sends this price to the EVs. It is assumed that all protection devices are synchronized and perform Algorithm 1 at the same time.

Algorithm 1: Congestion price update at each protection l with capacity c_l

input : $c_l, \kappa > 0$
while *true* **do**
  Measure load
  congestion state ← c_l − load
  price ← $\max\{\text{price} - \kappa \times \text{congestion state}, 0\}$
  Send price along with all received prices to children
  Wait until the next **clock tick**
end

According to Equation (6.2.18), each EV charger receives the prices from the protection devices and updates its charging rate based on the sum of these prices. Again, it is assumed that all EVs are synchronized and perform Algorithm 2 at the same time.

Algorithm 2: Rate adjustment at EV charger i

input : $\overline{x}_i$, new congestion prices
while *true* **do**
  λ ← vector of new congestion prices
  aggregate price ← $\sum_{l \in \mathcal{P}_i} \lambda_l$
  rate ← $\min\{\frac{w_i}{\text{aggregate price}}, \overline{x}_i\}$
  Start charging the battery at rate
  Wait until the next **clock tick**
end

In our implementation of the incentive-based approach, we do not consider the emergency respond mode proposed in [36], where EV charging is stopped when a protection's maximal loading violation time gets close to its triggering time.

6.3.2 Budget-based algorithm

The algorithms for the budget-based approach are derived from the results obtained in Section 6.2.3. First, the substation receives the budgets and the marginal benefits of the EV chargers. Based on Equation (6.2.35), Algorithm 3 is performed at the substation.

Algorithm 3: Budget update in substation

```
input  : α, new marginal benefits, budgets
while true do
    μ ← vector of new marginal benefits
    budgets ← budgets + α × μ
    Send budgets to protection device l = 1
    Wait until the next clock tick
end
```

The updated budgets are propagated through the protection devices back to the EV chargers. As the budgets go through the protection devices, each protection device executes Algorithm 4, which is based on the sequence defined by Equation (6.2.38).

Algorithm 4: Budget projection onto protection l with capacity c_l

```
input  : c_l, new budgets
while true do
    budgets ← new budgets ∈ V_l
    aggregate budgets ← Σ budgets
    if aggregate budgets > c_l then
        budgets ← budgets + (c_l − aggregate budgets) / length(budgets)
    end
    Send budgets to children
    Wait until the next clock tick
end
```

Once the budgets go through all protection devices back to the EVs chargers, each EV charger uses Algorithm 5 to update its charging rate. This algorithm is based on Equations (6.2.31) and (6.2.32). This computation can be performed in parallel and at the same time by each EV charger. We assume that all EVs are synchronized. This entails that all EVs must wait until all EVs have finished their computation.

Algorithm 5: Rate adjustment at EV charger i

```
input   : x̄, budget
while true do
    rate ← min{budget, x̄}
    Start charging the battery at rate
    if budget < x̄ then
        marginal benefit ← w_i/budget
    end
    Send budget and marginal benefit to the substation
    Wait until the next clock tick
end
```

6.4 Evaluation on the IEEE 13 Node Test Feeder

We now evaluate the incentive- and the budget-based approach in a static and a dynamic setting. In the static setting, we assume that the parameters of the optimization problem do not change over time. This means that the fixed load consumed at each node does not change over time. In the dynamic setting, we allow the fixed load to change with time. For both cases, we assume that a constant number of EVs remain connected.

We consider the distribution network depicted in Fig. 6.4.1. This network is based on a simplification of the IEEE 13-bus test feeder [80]. As in [36], the network is assumed to be balanced, which allows for a single phase analysis. Moreover, the voltage is assumed to have a constant value of 4.16kV. Normally, this voltage level must be further reduced by field transformers for the consumers. However, this is ignored for the sake of simplicity. Based on these assumptions, we can focus on the currents that flow through the network and control the amount of current used by each EV charger.

In our evaluation, we consider a benchmark case of $N = 18$ EV chargers and $M = 13$ protection devices. Without loss of generality, we assume the same priority level for

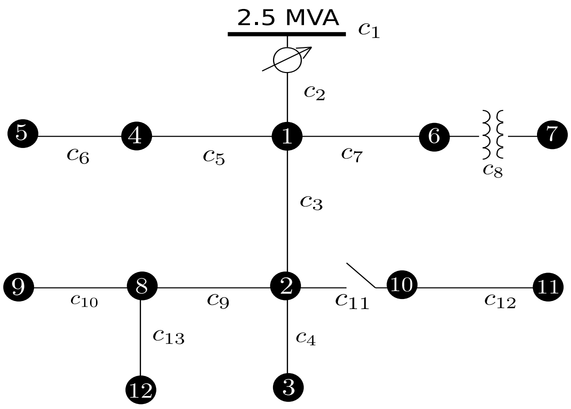


Figure 6.4.1: IEEE 13-node test feeder.

Table 6.4.1: Evaluation Parameters

Bus	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Scenario A (static)										
Load [A]	30	40	40	10	20	70	20	100	20	40
18 EVs	2	2	2	2	2	0	2	2	2	2
Scenario B (static)										
Load [A]	30	40	40	10	20	160	20	10	20	40
18 EVs	2	2	2	2	2	0	2	2	2	2
Scenario real load data (dynamic)										
Load (`2012-May-<day>`)	1, 2	3, 4	5, 6	7, 8	9, 10	11, 12	13, 14	15, 16	17, 18	19, 20
18 EVs	2	2	2	2	2	0	2	2	2	2

all EV chargers, $w_i = 1$, and a maximal charging current of 16 Amperes, $\overline{x}_i = 16$. The location of the EV chargers, as well as the fixed load values of all nodes, can be found in Table 6.4.1. With the position of the EV chargers and the topology of the grid shown in Fig. 6.4.1, we define our routing matrix as:

$$R = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{bmatrix}.$$

The vector of maximal available loading for the protections c_l is defined by their maximal load rating c_0, minus the capacity that is used to supply the fixed loads. In Table 6.4.1, we consider two static fixed load scenarios for which the maximal available loading vector c_A and c_B can be defined. In our evaluation parameters, these vectors are:

$$\begin{aligned} c_0 &= \begin{bmatrix} 600.96 & 730730 & 730 & 140 & 140 & 340 & 340 & 230 & 230 & 340 & 340 & 230 \end{bmatrix}^T \\ c_A &= \begin{bmatrix} 210.96 & 340 & 450 & 700 & 600 & 100 & 310 & 320 & 100 & 210 & 220 & 320 & 190 \end{bmatrix}^T \\ c_B &= \begin{bmatrix} 210.96 & 340 & 450 & 700 & 60 & 100 & 310 & 320 & 10 & 210 & 310 & 320 & 190 \end{bmatrix}^T. \end{aligned}$$

The control time slot duration is defined as $T_c = 20$ms and the triggering time of the protection devices is defined as $T_f = 200$ms.

6.4.1 Static evaluation

We consider two scenarios: a low loading Scenario A and a high loading Scenario B in Table 6.4.1. In Fig. 6.4.2, we see the protection devices that would be triggered if we allowed all EVs to charge at their maximal rate. We observe that for load Scenario A Protection 1 and 5 would fault, and for load Scenario B Protection 9

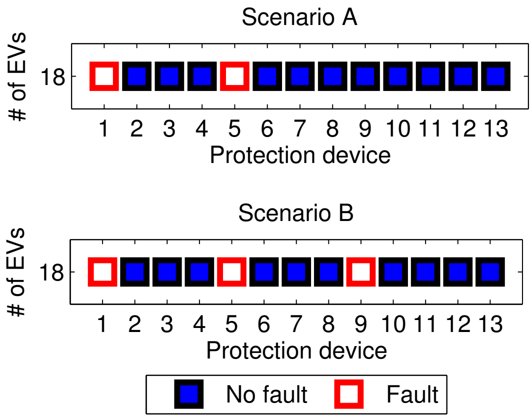


Figure 6.4.2: Overloaded protection devices at maximal EV charging rate.

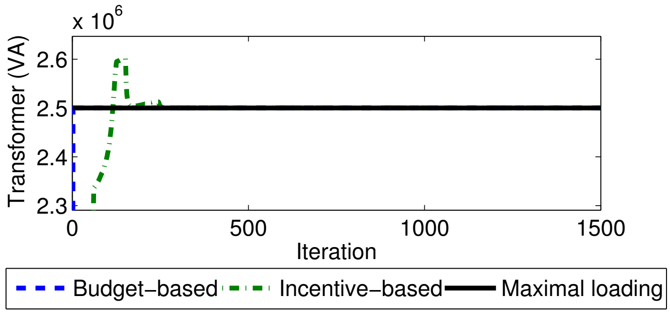


Figure 6.4.3: Scenario A loading.

would fault also.

The result of both optimization approaches in Scenarios A and B for the transformer are shown in Figures 6.4.3 and 6.4.4 for $\alpha = 1$ and $\kappa^* = 8.68 \times 10^{-5}$. As one can see, the budget-based approach outperforms the incentive-based approach, as fewer iterations are required to reach the optimal use of the infrastructure. Moreover, we see that the incentive-based approach causes a violation of the transformer's maximal loading rate in both scenarios. Since we assume that each iteration takes $T_c = 20$ms and we define the triggering time of the protections as $T_f = 200$ms, a violation of the maximal loading for more than 10 iterations would lead to a blackout. This is the case for the incentive-based approach in both scenarios. The performance of the incentive- and the budget-based approaches is mainly influenced by the respective step sizes, κ and α. The definition of the step size κ for the incentive-based approach

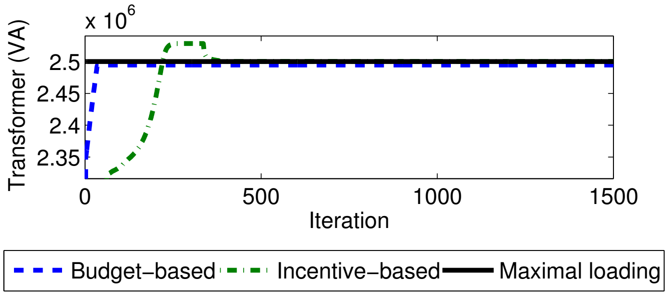


Figure 6.4.4: Scenario B loading.

was discussed in Section 6.2.2. As already discussed, if the value of κ is too high, the algorithm may become unstable, causing the charging rate of the EVs to oscillate. To avoid this, κ should be smaller than the maximal value for which stability is guaranteed, i.e., $\kappa^* = \frac{2}{16^2 \times 18 \times 5} = 8.68 \times 10^{-5}$. The stable step size becomes smaller as the number of EVs or the size of the grid increases. This leads to scalability problems.

For the budget-based approach, the definition of the step size α was discussed in Section 6.2.3. We established that the maximal value needed for reaching the optimal solution is $\alpha^* = \frac{18}{10^{10}}$. However, this approach remains stable even with a higher step size. The step size affects only the distance between the value to which we converge and the optimal solution.

We also studied the behavior of both approaches for different step size values using Scenario B. We allow the algorithms to run for 3,500 iterations and compare the number of iterations required to reach 95 % convergence to the optimal result. The optimal result was obtained by solving the problem with a centralized approach. We only consider the results that converged within the convergence criteria, $\epsilon_i = \epsilon_b = 10^{-3}$. As shown in Fig. 6.4.5, the higher the step size values, the faster the corresponding optimization algorithm's convergence. We see that the incentive-based approach offers a more rapid solution for the same step size value. However, the incentive-based approach becomes unstable when the value of its step size is higher than $\kappa = 7 \times 10^{-4}$. In turn, our budget-based approach still converges to 95 % optimality even for high step sizes. This means that we can choose larger step size than the incentive-based approach, thereby making our approach converge faster.

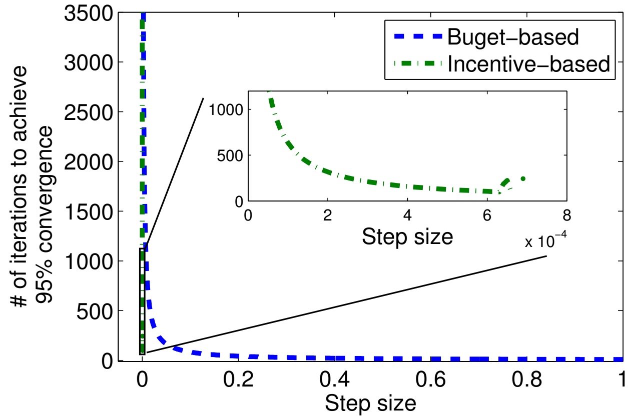


Figure 6.4.5: Number of iterations (Scenario B).

The static evaluation shows that our budget-based approach exhibits better reliability than the incentive-based approach. Under the defined assumptions, our approach guarantees that the maximal loading of the protection devices is not violated. Moreover, our evaluation reveals that our approach remains stable and can, therefore, offer better scalability than the incentive-based approach.

6.4.2 Dynamic evaluation

In this section, we analyze the behavior of the incentive and the dynamic budget-based approaches when the loads change over time. The changes in load cause changes in the maximal available capacity of the protection devices, c_l. Due to the highly dynamic changes of loads and the short available reaction time, it is important that any EV charging control approach remains stable and reacts quickly to the changing conditions of the grid.

We consider two scenarios. In the first, Scenario ABA, we iterate for 5 seconds between Scenarios A and B for a total period of 15 seconds (cf. [36]). The parameters can be found in Table 6.4.1. The second scenario uses real load measurements obtained from the `Smart*` data set [81]. This data set consists of the real load measurements of houses over several days with a resolution of 1 second. In our real load scenario, we use the measurements from the homeA-circuit. We assume that

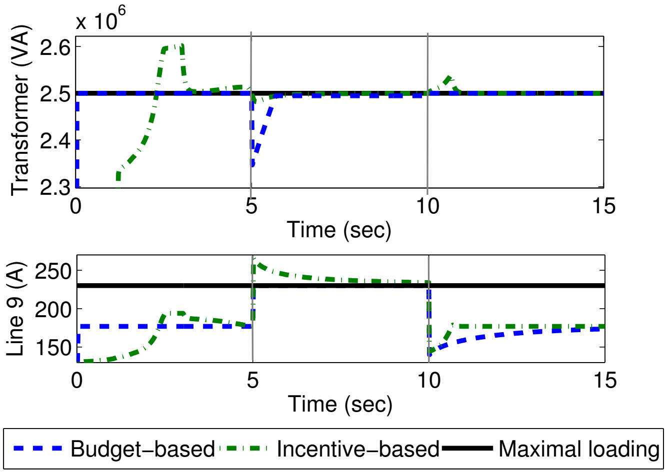


Figure 6.4.6: Loading in Scenario ABA.

each node consists of two houses. Therefore, we use the measurements of the homeA circuit from two different days, dividing the measured power by 120 V to obtain the currents (cf. Table 6.4.1). We focus on the period between 4 pm and 6 pm, when the highest energy consumption is expected.

The results for Scenario ABA for the transformer and Line 9 are shown in Fig. 6.4.6. We see that both approaches remain stable and can reach the optimal point under load changes. However, for Line 9, the load change causes the incentive-based algorithm to violate the maximal line loading. This line remains overloaded for several seconds, which would certainly lead to a blackout. This could be prevented by applying an emergency response mode as in [36], which would stop all EVs from charging and then resume the incentive-algorithm. However, if this is implemented, all EVs would be denied service. The advantage of the budget-based approach is that it does not need an emergency response mode because of the anytime property. We see in Fig. 6.4.6 that for the budget-based approach, the protection's maximal loading is never violated.

For the real load scenario, if we allowed all EV chargers to charge at their maximal rate, the only device that would fault is the transformer. In Fig. 6.4.7, we see the result of both approaches for the transformer: Around 5 pm the incentive-based algorithm overloads the transformer for several minutes. Therefore, if an emergency

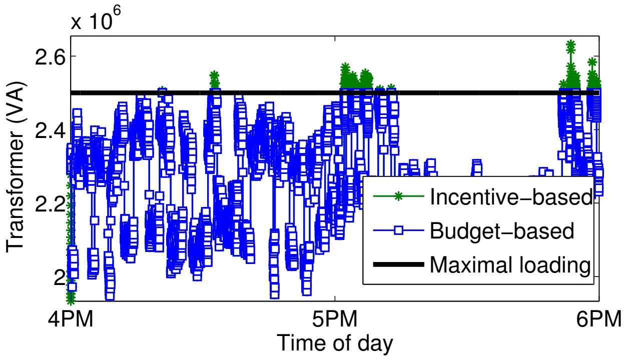


Figure 6.4.7: Loading in real load scenario.

response is implemented, the EVs would not be able to charge for several minutes. In contrast to this, we can see that the budget-based approach does not overload the transformer and retains its anytime property even under highly dynamic load changes.

The dynamic evaluation of both approaches revealed that the anytime property of the budget-based approach gives it a significant advantage over the incentive-approach, especially when dealing with highly dynamic changes in the grid.

6.5 Evaluation on the IEEE European Low Voltage Test Feeder

We now conduct three experiments to evaluate the dual (incentive-based) and primal (budget-based) EV NUM solution algorithms on a larger more representative power distribution grid. The first experiment evaluates the scalability and convergence of both algorithms under static conditions. The second experiment looks at the behavior of both algorithms under dynamic conditions. Finally, the third experiment shows the impact that both algorithms have on the voltages and currents of a distribution grid. The source code and data of all our experiments can be found online [1].

[1] github.com/chepeadan/EVNUM

All our experiments are based on the IEEE European Low Voltage Test Feeder [82]. Our evaluation grid, shown in Fig. 6.5.1, is a three-phase radial distribution feeder at the voltage level of 416 V (phase-to-phase) with a total of 906 buses, 905 lines, and 55 loads. All relevant data for our experiments are available in the test case data, except for the lines' ampacity. Thus, we define the ampacity based on empirical data of similar standard test networks [80]. The ampacity values used in our evaluations are summarized in Table 6.5.1. We assume that each load has one EV charger with a maximal charging capacity of 20 kW (3-phase). We also assume that the EVs start arriving at 5 p.m. according to a Poisson distribution with an arrival rate of 1 per minute. Furthermore, the EVs are assumed to be fully discharged upon arrival and wishing to be fully charged to their maximal capacity of 24 kWh.

To formulate the EV NUM optimization problem, we assume constant voltages. Hence, our optimization variables x are the currents that the EV chargers draw from the network. The maximum available capacity of the network devices c_l is defined by the lines' ampacity minus the current drawn by the loads. The maximum charging rate $\overline{x}$ is given by the maximum charging current of the EV chargers. Without loss of generality, we assume that all EVs have the same importance, i.e., $w_i = 1$.

To make use of the EV NUM formulation, one needs to assume a constant voltage. The reason is that if the voltage is constant, then the main network constraints are the line ampacity limits, which are linear and can be expressed with the NUM formulation $Rx < c$, see also [36, 37, 43, 44, 83]. While omitting voltage constraints is risky, as our evaluations will show, the NUM model offers a good approximation, when ampacity violations happen before voltage violations. In such cases, the NUM formulation provides a good trade-off between model accuracy and the required simplicity to implement distributed optimization methods.

6.5.1 Static evaluation

We evaluate the convergence of the dual and primal EV NUM algorithm for different step size values and consider the scalability behavior of both algorithms for varying EV numbers and grid size. Constant optimization parameters characterize the static

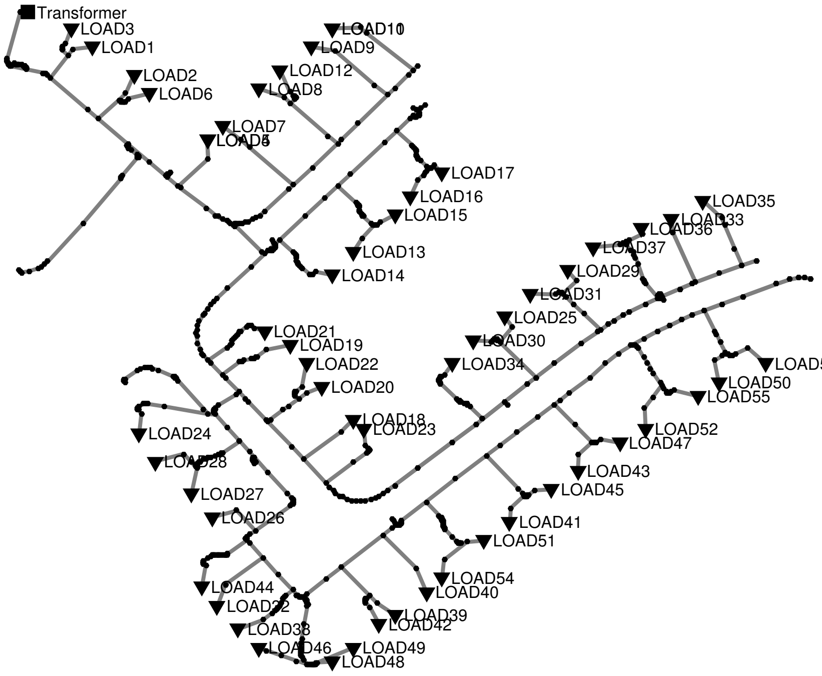


Figure 6.5.1: IEEE European Low Voltage Test Feeder.

evaluation, i.e., the parameters of the EV NUM problem don't change over time. The static case is the result of the fixed loads having a constant value, which leads to a constant maximal available loading of the devices c_l. Without loss of generality, we assume that the network is single phase for the static experiments. Hence, the size parameters for our EV NUM problem are $N = 55$ and $M = 905$. A centralized solution of the EV NUM problem is used as a reference for the optimal result.

The convergence experiment results for the dual algorithm in Fig 6.5.2 show that the dual algorithm becomes unstable when the step size is too large ($\kappa = 0.0001$). To avoid instability, a theoretical upper bound for the step size value was defined in Section 6.2.2, which guarantees dual algorithm convergence ($\kappa^* = 3.619e - 8$). Nonetheless, the theoretical upper bound is usually too conservative and a stable and faster step size can be used, e.g., $\kappa = 1e - 05$.

The convergence results for the primal algorithm in Fig. 6.5.3 show that the primal algorithm does not become unstable and converges within an increasing distance

Table 6.5.1: Power line ampacity used in evaluation.

Line code name	Ampacity (A)
2c_.007	56
2c_.0225	83
2c_16	83
35_SAC_XSC	110
4c_.06	210
4c_.1	560
4c_.35	210
4c_185	405
4c_70	560
4c_95_SAC_XC	180

to the optimum with increasing step size. Hence, our experimental results for the primal algorithm confirms the theoretical behavior discussed in Section 6.2.3 and demonstrate that the primal algorithm does not suffer from the instability issues of the dual algorithm.

To compare the scalability of both algorithms, we look at their convergence behavior as the EV NUM problem size parameters vary and measure the number of iterations required to reach 95% convergence. First, we modify N from 10 to 50 in increments of 10, which is equivalent to changing the number of EVs in the network. Then, we modify M from 100 to 900 in increments of 100, which is equivalent to modifying the size of the network. The results in Fig 6.5.4a and Fig 6.5.4b reveal that the dual algorithm is highly sensitive to changes in the number of EVs and grid size. In contrast, the results for the primal algorithm in Fig. 6.5.5a and Fig. 6.5.5b show that its convergence behavior remains almost constant for changes in the problem size parameters. Hence, our experiments reveal that the primal algorithm is less sensitive to variations in problem size and therefore offers significant scalability advantages over the dual algorithm. Moreover, unlike the primal algorithm, the dual algorithm does not guarantee feasible control values on each iteration, which increases the chance of a blackout. While the computation time of both algorithms can be neglected (closed-form solutions), both require communication on each iteration. If we assume a maximal communication delay of 20 ms per iteration and a minimal protection tripping time of 200 ms, then to guarantee feasible control values, the

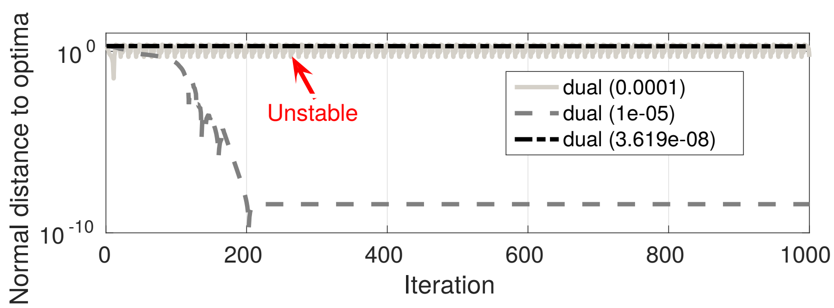


Figure 6.5.2: Dual convergence for different step size

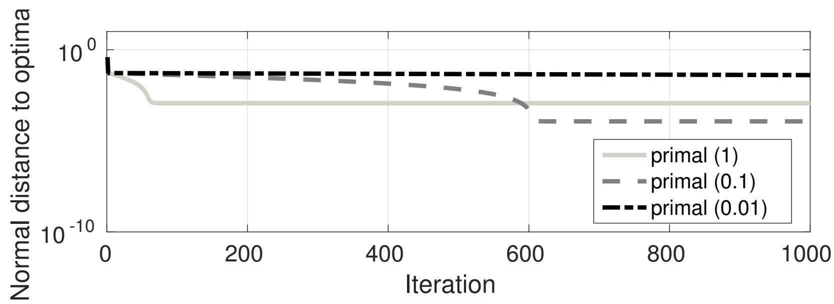
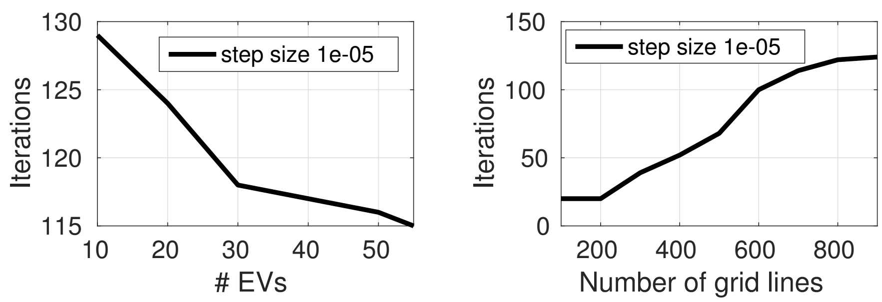


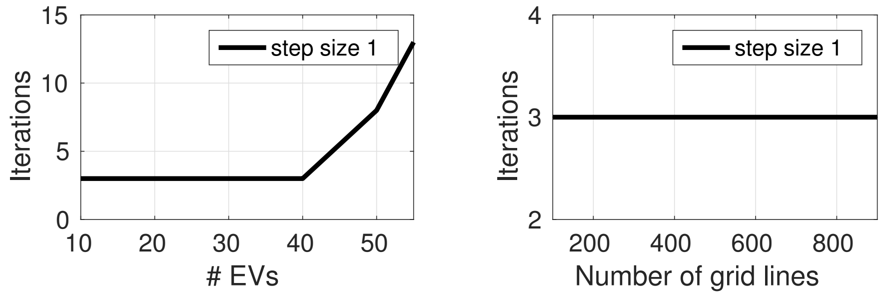
Figure 6.5.3: Primal convergence for different step size



(a) Different number of EVs **(b)** Different grid size

Figure 6.5.4: EV NUM dual algorithm static behavior (each iteration $\sim$ 20 ms)

dual algorithm would need to converge in less than 10 iterations. This is well bellow the actual number of iterations required by the dual algorithm in our results.



(a) Different number of EVs **(b)** Different grid size

Figure 6.5.5: EV NUM primal algorithm static behavior (each iteration $\sim$ 20 ms)

6.5.2 Dynamic evaluation

In the dynamic evaluation, we consider the load dynamics, which cause the maximal available network load c_l to change over time. We make use of the load shapes included in the IEEE European Low Voltage Test Feeder data, which are time series with a one-minute time resolution over 24 hours. The dual and primal algorithm are implemented in one-minute resolution, i.e., on each iteration of the algorithms the maximal available load of the devices c change. The problem size parameters for our three-phase evaluation network are $N = 55$ and $M = 3 \cdot 905$.

We consider the results of the EV NUM optimization for the main power line, which is the line most affected by congestion. The result for the dual algorithm in Fig. 6.5.6 shows that the maximum load condition of the line is violated as soon as EVs start to arrive. This result is expected, since the dual algorithm does not offer any guarantees that the network constraints $Rx < c$ will be fulfilled during runtime. To avoid overloading of the devices, the dual algorithm needs to be given enough time to come close to convergence. Hence, the dual control algorithm would need to be implemented at a higher frequency.

The result for the primal algorithm in Fig. 6.5.7 shows the desired behavior for real-time EV control: The algorithm makes maximum use of the network without overloading it. As explained in Section 6.2.3, the primal algorithm has the anytime property, which guarantees that the constraints of the EV NUM problem are fulfilled on each iteration. The anytime property gives the primal algorithm an advantage

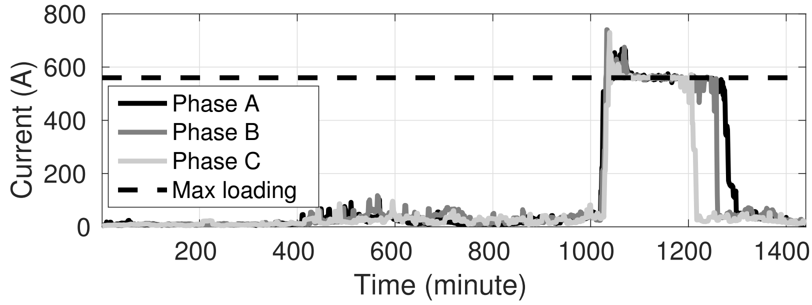


Figure 6.5.6: Dual algorithm dynamic behavior result for the main power line

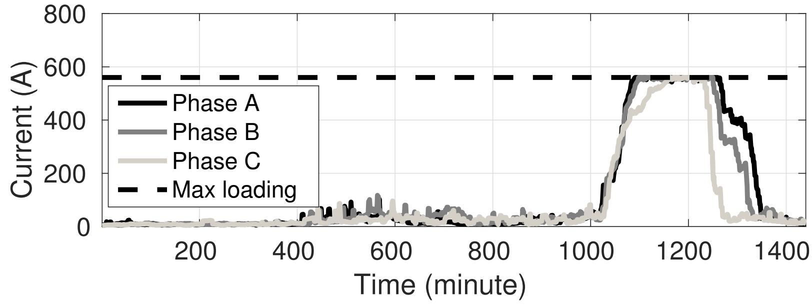


Figure 6.5.7: Primal algorithm dynamic behavior result for the main power line

over the dual algorithm because we don't need to provide the algorithm with enough iterations to guarantee control values that respect the network constraints. Hence, the primal algorithm can be implemented at a lower frequency than the dual algorithm.

6.5.3 Effects on a distribution network

To evaluate the effects of EV charging on the IEEE European Low Voltage Test Feeder, we conduct 3-phase power flow simulations using GridLab-D [84]. First, we evaluate the effect of charging the EVs without control for different charging rates. The results in Fig. 6.5.8 reveal that our evaluation grid can support the charging of EVs with a maximum charging power of 4 kW without any charging control. However, for a charging power of 7 kW, we start to see ampacity and voltage violations. With a charging power of 20 kW, we also see violations of the transformer's maximal loading. Hence, real-time EV charging control is required to allow higher charging powers than 4 kW and make better use of the grid's capacity without violating its

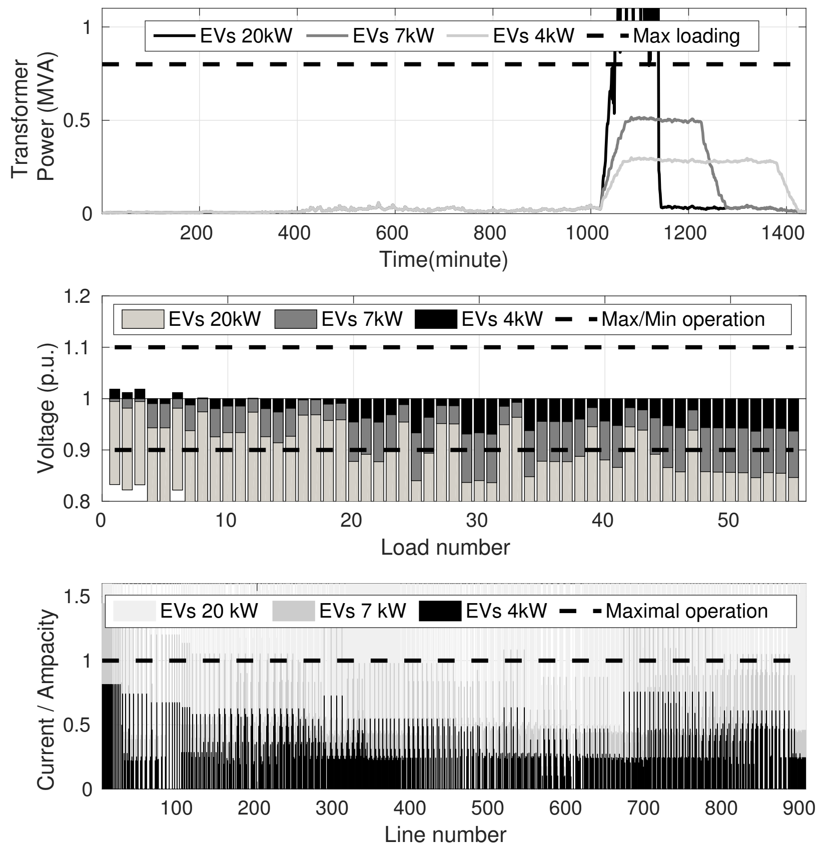


Figure 6.5.8: EV charging effects without control on the IEEE European Low Voltage Test Feeder.

constraints. This experiment also revealed that for this particular feeder ampacity violations happen before voltage violations.

We evaluated the impact of the EV NUM dual and primal algorithms on the grid by running simulations using the load results of our dynamic evaluation in Section 6.5.2. Fig. 6.5.9 demonstrates that the dual algorithm violates the network constraints, whereas the primal algorithm is able to guarantee charging rates that remain within the network constraints. Our GridLab-D 3-phase power flow simulation results match the behavior obtained in Section 6.5.2 with the EV NUM formulation. Hence, our experiments show that the EV NUM problem can capture the relevant constraints to design an effective real-time EV control algorithm for distribution grids, where the ampacity violations are the limiting factor. Moreover, this result demonstrates

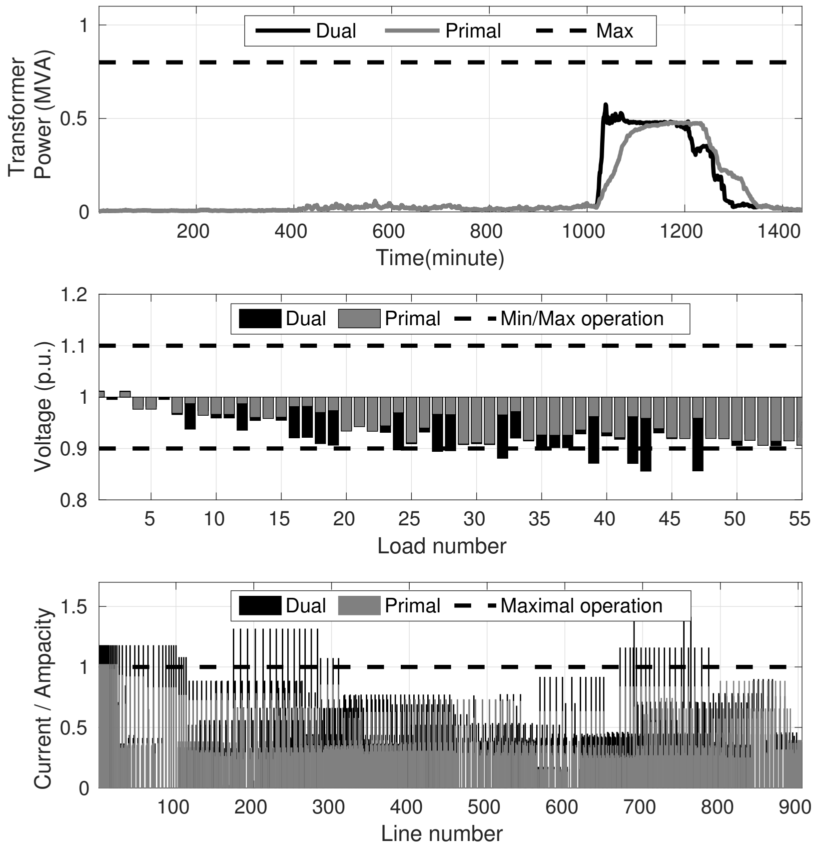


Figure 6.5.9: EV NUM control effects on the IEEE European Low Voltage Test Feeder.

the effectiveness of the primal algorithm for real-time EV charging control.

6.6 Discussion

6.6.1 EV NUM for real-time EV charging control

The EV NUM problem formulation offers a simplified model for the real-time EV charging congestion problem. The goal of using EV NUM is to capture the simplest form of the problem that still encompasses the relevant features required to design a distributed control algorithm. Several arguments can be made for the use of a more comprehensive model. However, more complex formulations do not allow the formulation of efficient distributed algorithms, that can cope with the real-time requirements. Another argument in favor of the simple EV NUM model is that more complex models need more network data. When it comes to massive application of the control algorithm, it is not certain that accurate network data of the distribution grids are available. In fact, the accuracy and even the availability of accurate distribution network models is known to be an issue [85]. Therefore, we consider that the EV NUM problem formulation can be an alternative to more complex models with higher computational demands and that require highly accurate network data.

6.6.2 Dual vs. primal EV NUM

Our evaluations have shown that the primal algorithm has an advantage over the dual algorithm regarding both scalability and reliability. The scalability advantage results from the stability issues of the dual algorithm. Our static evaluations in Section 6.5.1 confirm that the dual algorithm's stability depends on the step size. As explained in Section 6.2.2, the theoretical maximum stable step size is inversely proportional to the number of EVs and the size of the distribution grid. Hence, as the size of the problem increases, the step size must be reduced to guarantee stability, which in turn increases the number of iterations required to reach convergence. The primal algorithm does not need to reduce its step size as the problem size increases and therefore scales better to larger problems. Regarding the reliability advantage, our dynamic evaluations of Section 6.5.2 show that the dual algorithm requires a higher update frequency than the primal algorithm to avoid the violation of grid constraints. This means that the dual algorithm needs to be given more iterations to converge before its resulting control values can be applied. This exceeds the time required to avoid overloading. To avoid overloading in the dual algorithm, an emergency mechanism is proposed in [36] that stops all EVs from charging and restarts the algorithm. But this leads to temporary charging service denial. The reliability advantage of the primal algorithm comes from its anytime property, which guarantees that the problem constraints are satisfied on each iteration.

Nevertheless, the primal algorithm requires more communication than the dual algorithm. Therefore, the dual algorithm might be the preferred option when the problem size is relatively small and temporary violations of the grid constraints are permitted. The experiments in Section 6.5.3 show that the dual algorithm allows the EVs to charge faster at the cost of temporary grid constraint violations.

Chapter 7

Conclusions

In this thesis, we used distributed optimization to address two key challenges in the control of EV charging. First, we considered the scalability problem faced by EV aggregators to schedule the charging of a large number of EVs and provide system services. Then, we examined the real-time EV charging congestion control problem to allow the safe integration of large EV numbers into the current distribution infrastructure. For both challenges, we have leveraged distributed optimization to define our EV ADMM and EV NUM solution approaches. In this chapter we summarize our conclusions, discuss the limitations of our methods and provide an outlook on future work.

7.1 Summary

In EV ADMM, we proposed a novel scalable distributed convex optimization framework for EV aggregators based on the solution of a general EV aggregator optimization problem with ADMM. To demonstrate its scalability, we evaluated our framework with up to 1 million EVs for the valley filling problem. We also solved the valley filling and the charging cost minimization problem with up to 100 000 EVs and benchmarked our distributed solution against a centralized solution.

In contrast to previous works, EV ADMM can be generalized to several EV aggregator problems, because it allows the specification of global and local objectives and constraints. We demonstrate this feature in our versatility experiments by considering battery depreciation costs and V2G services for two EV aggregator problems. The versatility experiments also revealed that EV ADMM solves strictly convex problems, such as valley filling, faster than non-strictly convex problems, such as charging cost minimization. Furthermore, the results indicate that the presence of active global constraints increases the number of EV ADMM iterations and consequently EV ADMM's runtime.
The scalability comparison against the state-of-the-art centralized optimization has shown that EV ADMM offers better scalability regarding runtime and peak memory use. EV ADMM scales linearly with an increasing number of EVs and the ability to parallelize EV ADMM provides a significant speedup with moderate memory increase compared to a serial implementation. Thus, EV ADMM provides a controllable trade-off between runtime and memory use. Furthermore, the results demonstrate that for fleets with up to 100s of EVs, a centralized optimization can offer faster solutions. For larger fleets, however, our distributed optimization approach provides a distinct advantage regarding runtime and memory use. EV ADMM can, therefore, be used to solve convex EV charging planning and control optimization problems that have so far been considered intractable due to large EV numbers and memory limitations.

To support the real implementation of EV ADMM in a distributed environment, we also designed and implemented DOPS, a general purpose framework for distributed computation of EV ADMM-like algorithms over a pub/sub middleware. An application developer only needs to implement the concise interface of DOPS and specify the optimization procedures of agents to execute optimization algorithms in a distributed infrastructure. To show the applicability of our framework, we solved the valley-filling problem in EV charging control using three different communication schemes: centralized, decentralized and decentralized with in-network aggregation. Our experimental evaluation was based on publicly available data and executed on an overlay tree consisting of 15 virtual machines connected via a VLAN network. The results demonstrate that in-network aggregation significantly improves the runtime performance of a decentralized implementation. In fact, the aggregation capability of

DOPS doubled the runtime performance of the decentralized approach. Although the centralized approach still performed best for 120 EVs regarding runtime, we predict that already for 150 EVs the decentralized approach will be faster. Nevertheless, the exact number of EVs where a decentralized implementation becomes faster than a centralized one will depend on the technical infrastructure and the particular use case.

In summary, with EV ADMM, we have designed a method that solves the computational scalability problems of EV aggregators and, with DOPS, we have also contributed to its real implementation in a distributed environment.

In EV NUM, we proposed and evaluated a distributed anytime algorithm for the real-time congestion control of EV charging. The main challenge in real-time EV charging control is the short time available to offer a control response, usually in milliseconds. To address this, we formulated the EV NUM problem, which maximizes the EV chargers' utility of the grid without violating its operational constraints. By solving the EV NUM problem with primal decomposition, we formally designed a real-time EV charging control protocol that has the anytime property. This means that the results of each iteration are feasible. Moreover, the longer the algorithm runs, the closer the result gets to the optimal solution.
We evaluated our algorithm against the state-of-the-art dual decomposition solution on the IEEE 13 Node Test Feeder and the IEEE European Low Voltage Test Feeder. Our simulation-based evaluation on the smaller grid shows that our approach exhibits better reliability and can offer results faster than the state-of-the-art EV NUM dual approach. Moreover, our dynamic load evaluation demonstrated that our primal approach can cope better with dynamic changes in the grid. Our more comprehensive experiments on the larger network demonstrate that the primal algorithm outperforms the dual algorithm in scalability and reliability because it does not suffer from convergence problems. Also, its anytime property allows it to adapt quickly to the fast-changing dynamics of the grid. Moreover, our 3-phase power flow simulations show the effectiveness of the NUM formulation to model networks where the ampacity constraints are the main bottleneck. As expected, the constraint violations of the dual algorithm resulted in ampacity and voltage violations, while our primal approach caused no violations. Nevertheless, the advantages of our approach

come at the cost of greater message complexity than that of the dual approach. All in all, considering that the safe operation of the grid is the primary concern in EV charging congestion control, our distributed anytime algorithm offers a good trade-off between communication complexity, scalability, and reliability.

In summary, with EV NUM we have addressed the problem of real-time EV charging congestion control by formally designing a real-time EV charging control protocol with the anytime property. Our solution allows for the provision of optimal EV charging control in milliseconds instead of minutes, which is required to prevent the triggering of protection devices.

7.2 Limitations

The convexity assumption causes the main limitations of our approaches. In both our approaches we have the requirement of a convex optimization problem formulation. Convexity allows for theoretical guarantees that our distributed optimization algorithms will converge to the optimal solution of the original problem. Without convexity, our algorithms are not guaranteed to converge. This limits the general applicability of our approaches. While many EV charging problems can be formulated as a convex problem, there are certain problems where the constraints need to be relaxed or ignored to allow a convex formulation. The convexity requirement can lead to simplified models that might not include relevant constraints or do not fully guarantee the control requirements. However, this is a common limitation among works that use distributed optimization techniques. One important consequence of this is that in our approaches the EV charging rates have to be continuous between 0 and a maximal value. Currently, EV charging is usually implemented using discrete charging rate values. However, the technical requirements for continuous charging rates are available.

7.3 Future work

Future studies on EV ADMM should focus on relaxing the convexity requirement for EV aggregator optimization problems. Relaxing the convexity requirement would allow discrete charging rates. EV ADMM can work for non-convex problems. However, there are no theoretical guarantees that the algorithm will converge. Further studies are required to determine if EV ADMM can be generalized to non-convex EV aggregator problems. Another direction for future work in EV ADMM is its implementation for receding horizon closed-loop control to deal with prediction uncertainty. A first study is carried out in [30]. However, a more comprehensive study on the scalability and performance of receding horizon EV ADMM would be an excellent addition. Finally, we consider that the model used in EV ADMM for EV aggregators could be used to achieve scalability for aggregators of other controllable devices, like Home Energy Management Systems (HEMS).

Future research on EV NUM should focus on integrating more complex constraints into the model, like operational voltage constraints. Besides overloading grid components, a large number of EVs can also lead to voltage swings. The operation of the distribution network requires a voltage variation of no more than 10 % from its nominal value. In EV NUM, we assume that this is handled by the utility company using other devices, e.g., tap changer transformers. However, EVs could also contribute to control voltage levels. The problem in our EV NUM approach is that implementing voltage constraints would require power flow constraints. These constraints would make the NUM problem non-convex. Our distributed optimization method is only guaranteed to work with a convex problem formulation. Hence, future contributions on EV NUM could develop methods to include convex voltage violations constraints into the model.

Finally, we would like to point out that while our contributions solve significant technical problems in EV charging control, several non-purely technical challenges should be part of future research. For instance, there is the issue of control acceptance, i.e., the problem of defining the right incentive that makes people willing to allow the charging control of their EV. Another relevant issue is the truthful reporting of EVs, i.e., to guarantee that the values reported to the control by EVs are accurate. To address these and similar challenges, we consider that a combination of disciplines is required. These issues should be studied not only from the technical side but also consider insights from behavioral economics, finance, and other disciplines. Examples of such research in related fields can be found in [86–88]. We think these are highly relevant topics, where more focus is required.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Amsterdam Roundtables Foundation and McKinsey & Company, "Electric vehicles in Europe: gearing up for a new phase?" Online: www.mckinsey.com, Tech. Rep., 2014, accessed: 23.06.2015.

[2] "Global EV Outlook 2016," Online: https://www.iea.org/publications/freepublications/publication/Global_EV_Outlook_2016.pdf, International Energy Agency, Tech. Rep., 2016, accessed: 18.05.2017.

[3] C. Sullivan. (2009, August) Will Electric Cars Wreck the Grid? Scientific American. Accessed: 18.05.2017. [Online]. Available: http://scientificamerican.com/article/will-electric-cars-wreck-the-grid

[4] G. Putrus, P. Suwanapingkarl, D. Johnston, E. Bentley, and M. Narayana, "Impact of electric vehicles on power distribution networks," in *Vehicle Power and Propulsion Conference, 2009. VPPC'09. IEEE.* IEEE, 2009, pp. 827–831.

[5] P. Richardson, D. Flynn, and A. Keane, "Impact assessment of varying penetrations of electric vehicles on low voltage distribution systems," in *Power and Energy Society General Meeting, 2010 IEEE.*

[6] J. C. Gomez and M. M. Morcos, "Impact of EV battery chargers on the power quality of distribution systems," *Power Delivery, IEEE Transactions on*, vol. 18, no. 3, pp. 975–981, 2003.

[7] W. Kempton and J. Tomić, "Vehicle-to-grid power fundamentals: Calculating capacity and net revenue," *Journal of power sources*, vol. 144, no. 1, pp. 268–279, 2005.

[8] D. Callaway and I. Hiskens, "Achieving controllability of electric loads," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 184–199, jan. 2011.

[9] C. Goebel and D. S. Callaway, "Using ICT-controlled plug-in electric vehicles to supply grid regulation in california at different renewable integration levels," *Smart Grid, IEEE Transactions on*, vol. 4, no. 2, pp. 729–740, June 2013.

[10] J. Zhu, *Optimization of power system operation.* John Wiley & Sons, 2015, vol. 47.

[11] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[12] K. Y. Lee and M. A. El-Sharkawi, *Modern heuristic optimization techniques: theory and applications to power systems.* John Wiley & Sons, 2008, vol. 39.

[13] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. Resende, and W. R. Stewart, "Designing and reporting on computational experiments with heuristic methods."

[14] S. Boyd, "Alternating direction method of multipliers," EE364b, 2011.

[15] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, 1998.

[16] S. Low and D. Lapsley, "Optimization flow control. I. Basic Algorithm and Convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[17] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[18] L. Pieltain Fernández, T. Román, R. Cossent, C. Domingo, and P. Frías, "Assessment of the impact of plug-in electric vehicles on distribution networks," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 206–213, Feb 2011.

[19] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *Power Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 371–380, Feb 2010.

[20] J. Peppanen and S. Grijalva, "Neighborhood electric vehicle charging scheduling using particle swarm optimization," in *PES General Meeting| Conference & Exposition, 2014 IEEE*. IEEE, 2014, pp. 1–5.

[21] O. Sundström and C. Binding, "Optimization methods to plan the charging of electric vehicle fleets," in *Proc. Int'l Conf. on Control, Communication and Power Engineering (CCPE)*, San Diego, CA, 2010, pp. 323–328.

[22] A. Papavasiliou, S. Oren, and R. O'Neill, "Reserve requirements for wind power integration: A scenario-based stochastic programming framework," *Power Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 2197–2206, nov. 2011.

[23] M. R. V. Moghadam, R. Zhang, and R. T. B. Ma, "Demand response for contingency management via real-time pricing in Smart Grids," in *2014 IEEE International Conference on Smart Grid Communications, SmartGridComm 2014, Venice, Italy, November 3-6, 2014*.

[24] C. Goebel, H.-A. Jacobsen, V. del Razo, C. Doblander, J. Rivera, J. Ilg, C. Flath, H. Schmeck, C. Weinhardt, D. Pathmaperuma, H.-J. Appelrath, M. Sonnenschein, S. Lehnhoff, O. Kramer, T. Staake, E. Fleisch, D. Neumann, J. Strüker, K. Erek, R. Zarnekow, H. Ziekow, and J. Lässig, "Energy informatics," *Business & Information Systems Engineering*, vol. 6, no. 1, pp. 25–31, 2014.

[25] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control for large populations of plug-in electric vehicles: Application of the Nash certainty equivalence principle," in *Control Applications (CCA), 2010 IEEE International Conference on*, sept. 2010, pp. 191–195.

[26] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, dec. 2011, pp. 5798–5804.

[27] L. Gan, U. Topcu, and S. H. Low, "Stochastic decentralized protocols for electric vehicle charging with discrete charging rate," in *IEEE Power and Energy Society General Meeting*, San Diego, CA, Jul. 2012.

[28] A. Mercurio, A. Di Giorgio, and F. Purificato, "Optimal Fully Electric Vehicle load balancing with an ADMM algorithm in Smartgrids," in *Control &*

Automation (MED), 2013 21st Mediterranean Conference on. IEEE, 2013, pp. 119–124.

[29] W.-J. Ma, V. Gupta, and U. Topcu, "On distributed charging control of electric vehicles with power network capacity constraints," in *American Control Conference (ACC), 2014.* IEEE, 2014, pp. 4306–4311.

[30] M. Gonzalez Vaya, G. Andersson, and S. Boyd, "Decentralized control of plug-in electric vehicles under driving uncertainty," in *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES.*

[31] M. Bradley and Associates, "Electric vehicle grid integration in the U.S., Europe, and China," M.J Bradley and Associates, Tech. Rep., 2013.

[32] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, "Integration of electric vehicles in the electric power system," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 168–183, 2011.

[33] S. Shao, M. Pipattanasomporn, and S. Rahman, "Challenges of PHEV penetration to the residential distribution network," in *Power & Energy Society General Meeting, 2009. PES'09. IEEE.* IEEE, 2009, pp. 1–8.

[34] K. Turitsyn, N. Sinitsyn, S. Backhaus, and M. Chertkov, "Robust broadcast-communication control of electric vehicle charging," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on.* IEEE, 2010, pp. 203–207.

[35] "AIMD-like algorithms for charging electric and plug-in hybrid vehicles," in *Electric Vehicle Conference (IEVC), 2012 IEEE International.*

[36] O. Ardakanian, C. Rosenberg, and S. Keshav, "Distributed Control of Rlectric Vehicle Charging," in *Proceedings of the fourth International Conference on Future Energy Systems*, ser. e-Energy '13. New York, NY, USA: ACM, 2013, pp. 101–112.

[37] O. Ardakanian, S. Keshav, and C. Rosenberg, "Real-Time Distributed Control for Smart Electric Vehicle Chargers: From a Static to a Dynamic Study," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2295–2305, Sept 2014.

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundation and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[39] M. Boddy and T. Dean, "Solving Time-dependent Planning Problems," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, vol. 2, 1989.

[40] J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen, "Alternating Direction Method of Multipliers for decentralized electric vehicle charging control," in *2013 IEEE 52nd Annual Conference on Decision and Control (CDC)*, Dec 2013, pp. 6960–6965.

[41] J. Rivera, C. Goebel, and H. A. Jacobsen, "Distributed Convex Optimization for Electric Vehicle Aggregators," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1852–1863, July 2017.

[42] J. Rivera, M. Jergler, A. Stoimenov, C. Goebel, and H.-A. Jacobsen, "Using publish/subscribe middleware for distributed EV charging optimization," *Computer Science-Research and Development*, pp. 1–8, 2014.

[43] J. Rivera and H.-A. Jacobsen, "A distributed anytime algorithm for network utility maximization with application to real-time EV charging control," in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, Dec 2014, pp. 947–952.

[44] J. Rivera, C. Goebel, and H.-A. Jacobsen, "A Distributed Anytime Algorithm for Real-Time EV Charging Congestion Control," in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, ser. e-Energy '15, 2015, pp. 67–76.

[45] J. Rivera and H. A. Jacobsen, "On the Effects of Distributed Electric Vehicle Network Utility Maximization in Low Voltage Feeders," *arXiv preprint arXiv:1706.10074*, 2017.

[46] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[47] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[48] S. Boyd, L. Xiao, A. Mutapic, J. Dattorro, and J. Mattingley, "Subgradient Methods, Decomposition Methods, Alternating Projections," Notes for EE364b, Stanford University, 2007.

[49] Y. Ota, H. Taniguchi, T. Nakajima, K. Liyanage, J. Baba, and A. Yokoyama, "Autonomous Distributed V2G (Vehicle-to-Grid) Satisfying Scheduled Charging," *Smart Grid, IEEE Transactions on*, vol. 3, no. 1, pp. 559–564, March 2012.

[50] M. Moghadam, R. Zhang, and R. Ma, "Randomized response electric vehicles for distributed frequency control in smart grid," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 139–144.

[51] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Foundations and Trends® in Optimization*, vol. 1, no. 2, pp. 73–126, 2014.

[52] H. L. Willis, *Power distribution planning reference book.* CRC press, 2004.

[53] C. K. Wen, J. C. Chen, J. H. Teng, and P. Ting, "Decentralized Plug-in Electric Vehicle Charging Selection Algorithm in Power Systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1779–1789, Dec 2012.

[54] R. Hermans, M. Almassalkhi, and I. Hiskens, "Incentive-based coordinated charging control of plug-in electric vehicles at the distribution-transformer level," in *American Control Conference (ACC), 2012.* IEEE, 2012, pp. 264–269.

[55] N. Leemput, J. Van Roy, F. Geth, P. Tant, B. Claessens, and J. Driesen, "Comparative analysis of coordination strategies for electric vehicles," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, dec. 2011, pp. 1–8.

[56] D. P. Bertsekas, *Nonlinear programming.* Athena Scientific, 2004.

[57] L. Serra, Z. Chehab, Y. Guezennee, and G. Rizzoni, "An aging model of Ni-MH batteries for hybrid electric vehicles," in *Vehicle power and propulsion, 2005 IEEE conference*. IEEE, 2005, pp. 8–pp.

[58] J. Rivera, "Optimal charging of electric vehicles with renewable energy in smart grids," Master thesis, Technische Universität München, Nov 2012, Department of Electrical Engineering.

[59] The MathWorks Inc, "MATLAB R2014a," 2014.

[60] J. Lofberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, Sept 2004, pp. 284–289.

[61] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2014. [Online]. Available: http://www.gurobi.com

[62] SWM München, "Netzdaten," Online: http://www.swm-infrastruktur.de, 2012, accessed: 13.06.2012.

[63] E. E. Exchange, "European Electricity Index (EEX)," Online: http://www.eex.com/en/, 2012, accessed: 13.06.2012.

[64] C. Goebel, "On the business value of ICT-controlled plug-in electric vehicle charging in California," *Energy Policy*, vol. 53, no. C, pp. 1–10, 2013.

[65] G. Sharma and J. Martin, "MATLAB®: a language for parallel computing," *International Journal of Parallel Programming*, vol. 37, no. 1, pp. 3–36, 2009.

[66] G. M. Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities," in *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, ser. AFIPS '67 (Spring).

[67] N. K. Pandey, K. Zhang, S. Weiss, H.-A. Jacobsen, and R. Vitenberg, "Distributed Event Aggregation for Content-based Publish/Subscribe Systems," in *Distributed Event-based Systems (DEBS)*, 2014.

[68] ——, "Minimizing the Communication Cost of Aggregation in Publish/Subscribe Systems," in *35th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2015.

[69] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *Signal Processing, IEEE Transactions on*, vol. 61, no. 10, pp. 2718–2723, 2013.

[70] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "D-ADMM: A distributed algorithm for compressed sensing and other separable optimization problems," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*.

[71] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, Jun. 2003.

[72] E. Fidler, H. A. Jacobsen, G. Li, and S. Mankovski, "The padres distributed publish/subscribe system," in *In 8th International Conference on Feature Interactions in Telecommunications and Software Systems*, 2005, pp. 12–30.

[73] G. Cugola, E. Di Nitto, and A. Fuggetta, "The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS," *IEEE Transactions on Software Engineering*, vol. 27, no. 9, pp. 827–850, Sep. 2001.

[74] G. Li and H.-A. Jacobsen, "Composite Subscriptions in Content-based Publish/Subscribe Systems," in *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, ser. Middleware '05. New York, NY, USA: Springer-Verlag New York, Inc., 2005, pp. 249–269.

[75] N. K. Pandey, K. Zhang, S. Weiss, R. Vitenberg, and H.-A. Jacobsen., "Distributed Event Aggregation for Content-based Publish/Subscribe Systems," in *Distributed Event-based Systems (DEBS)*, 2014.

[76] C. Goebel and M. Voß, "Forecasting Driving Behavior to Enable Efficient Grid Integration of Plug-in Electric Vehicles," *IEEE Online Conference on Green Communications*, 2012.

[77] M. Chaturvedi, "Substation IED communications," in *Power Engineering Society Winter Meeting, 2002. IEEE*, vol. 1, 2002, pp. 596 vol.1–.

[78] F. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, proportional Fairness and Stability," in *Journal of the Operational Research Society*, vol. 49, 1998.

[79] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[80] W. Kersting, "Radial Distribution Test Feeders," in *Power Engineering Society Winter Meeting, 2001. IEEE*, vol. 2, 2001, pp. 908–912 vol.2.

[81] S. Barker, A. Mishra, D. Irwin, E. Cecchet, J. Albrecht, and P. Shenoy, "Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes," in *Proceedings Data Mining Applications In Sustainability*, 2012.

[82] IEEE, "IEEE Radial Distribution Test Feeders," Online: https://ewh.ieee.org/soc/pes/dsacom/testfeeders/, accessed: 15.02.2016.

[83] O. Ardakanian, C. Rosenberg, and S. Keshav, "Real-time distributed congestion control for electrical vehicle charging," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 3, pp. 38–42, 2012.

[84] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, 2008.

[85] B. Meehan, *Modeling Electric Distribution with GIS*. Esri Press, 2013.

[86] C.-M. Loock, T. Staake, and F. Thiesse, "Motivating Energy-Efficient Behavior with Green Is: An Investigation of Goal Setting and the Role of Defaults." *Mis Quarterly*, vol. 37, no. 4, pp. 1313–1332, 2013.

[87] M. Weiss, F. Mattern, T. Graml, T. Staake, and E. Fleisch, "Handy feedback: Connecting smart meters with mobile phones," in *Proceedings of the 8th international conference on mobile and ubiquitous multimedia*. ACM, 2009, p. 15.

[88] V. Tiefenbeck, T. Staake, K. Roth, and O. Sachs, "For better or for worse? Empirical evidence of moral licensing in a behavioral energy conservation campaign," *Energy Policy*, vol. 57, pp. 160–171, 2013.