# LINEAR OPTIMIZATION METHODS FOR VEHICLE ENERGY AND COMMUNICATION NETWORKS

A DISSERTATION
SUBMITTED TO THE INSTITUTE FOR
COMPUTATIONAL AND MATHEMATICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nicole Anahita Taheri

June 2012

# Abstract

Over the few next decades, the personal vehicle may evolve into a device distinct from what exists today. This thesis considers energy and communication systems among vehicles that utilize two developing technologies: battery-powered drive trains and accurate sensors. We use linear optimization to construct separate algorithms for networks of Plug-in Electric Vehicles (PEVs) and networks of Sensor-equipped Vehicles (SVs).

Plug-in electric vehicles will have flexible charging options, and may be capable of transmitting electricity back to the grid (i.e., discharging). We construct an automated mechanism for a fleet of PEVs that efficiently organizes distributed energy trading to benefit both the consumers and the electric utilities. A linear programming model of the fleet provides a *composite valuation*, which can be used in an online environment managed by a fleet aggregator to allocate feasible energy exchange schedules that decrease the peak electricity demand and reduce the cost to consumers. The resulting charging and discharging schedules are assigned to tens of thousands of vehicles instantly as they plug into the grid and are robust to unexpected events in driving patterns. We give empirical results based on electricity and gasoline pricing, electricity demand, vehicle characteristics, and driving behaviors.

Sensor-equipped vehicles use accurate sensors to detect their surroundings; such technology has been used to help improve traffic flow and safety. We consider a network of SVs that measure the distances between one another. In this framework, a semidefinite programming (SDP) relaxation is an efficient computational method to solve the sensor network localization problem of determining the locations of vehicles on the road given some of the pair-wise distances between them. We present two results about the SDP relaxation that can be applied to localizing sensor-equipped vehicles. First, we provide the first non-asymptotic bound on the required sensor communication range to ensure a unique localization of the sensors. Second, we show that if the graph induced from a network of $n$ sensors in dimension $d$ is a $(d+1)$-lateration graph with points in general position, then the graph admits a positive semidefinite stress matrix with rank $n$, and hence the SDP relaxation will produce the correct localization of the points.

For my parents, Reza and Jane

# Acknowledgments

Yinyu Ye has been a wonderful thesis adviser. His energy and enthusiasm for his work is inspiring, and I hope to one day have a research group like his.

The main project in this thesis was initiated and explored in collaboration with Robert Entriken at the Electric Power Research Institute (EPRI), from whom I have learned about the electricity sector and electric transportation, in addition to valuable life lessons and skills. EPRI funded three years of this research, and I am grateful for their support and essential insights.

Michael Saunders has been a reliable mentor and friend during my time at Stanford, as well as my reference for LaTeX and proper grammar. Walter Murray introduced me to optimization, and is a main reason I became interested in the topic. Margot Gerritsen is a strong role model for me; she is an impressive researcher, leader and advocate for computational mathematics. Erica Plambeck's work on green technology is admirable, and it was a privilege to have her as the chair of my defense committee.

I collaborated on research projects with fellow graduate students Santiago Akle, Onkar Dalal, and Davood Shamsi, who taught me about math, research, and cooperation. Weekly meetings with Professor Ye's research group were consistently interesting and provided insightful feedback. I would especially like to thank Zizhuo Wang for a number of helpful discussions.

This thesis would not have been possible without the support of my family: Reza and Jane Taheri, and Kiana and Mark Hackett. Their encouragement and over-confidence in my abilities has served as constant support.

I am indebted to James Merrick for serving as a valuable sounding board during the last year of this work, even when he would rather be enjoying the sunshine. The friendships of Dana Doolin, Sara Herrera, Priya Singh, and Meg Underwood have been invaluable; their successes and compassion in life have been a great reminder of life outside of graduate school.

My two best friends along the way, Aubrey Overson and Morgan Davis, have been encouraging and also distracting; I am grateful to them for both of those qualities. Our weekly dinner club (with Aubrey, Morgan, Andrew Pariser, Matt Claypotch, Rio Goodman, Nick Henderson, Nick West, and Mary Cameron) has been a highlight of my time at Stanford, and it has been a privilege to be part of such an exclusive club. I relied on the Nicks for technical support and advice, on Andrew and Mary for loyal friendship, and on Rio and Matt (a.k.a. Potch) for both high and low comedy.

And finally, the Institute for Computational and Mathematical Engineering (ICME) has been my home for the past 5 years, where I fit in like never before, made truly wonderful friends, and became a computational mathematician.

# Table of Contents

# List of Tables

# List of Figures

# 1 Introduction

Over the next few decades, the personal vehicle is expected to evolve into a device distinct from what exists today. Future vehicles will most likely communicate with each other to collaborate and improve standards such as safety, traffic flow and environmental benefits [15, 38].

In this thesis, linear optimization techniques are used to construct algorithms that determine the optimal or feasible dynamics of vehicle networks. We focus on two developing vehicle technologies: battery-powered drive trains and accurate sensors. For Plug-in Electric Vehicles (PEVs), we use linear programming to model energy demands of a vehicle fleet and utilize the vehicles' batteries to benefit both consumers and utilities. For Sensor-equipped Vehicles (SVs), we use a Semidefinite Programming relaxation to determine the relative vehicle locations, given a subset of the distances between vehicles. We consider networks of PEVs and SVs separately, and use linear optimization to construct mechanisms that facilitate communication or energy flow among vehicle networks.

The approaches, models, and algorithms applied to these two types of vehicle networks are separate and distinct, but nonetheless related. For both vehicle networks considered, we use linear optimization and operations research methods to facilitate cooperative behavior among vehicles.

## 1.1 Plug-in Electric Vehicle Network

According to the United States Environmental Protection Agency [21], the average gasoline-powered vehicle currently emits over 11,000 pounds of $CO_2$ each year; a high penetration of PEVs could reduce the emissions of passenger vehicles by over 60% [17]. Given a growing prevalence of PEVs in the vehicle market, and that the average vehicle is currently expected to be on the road for around 10 years [27], there may be a noticeable growth in the penetration of PEVs in the next two decades. The Electric Power Research Institute (EPRI) expects that by 2030, the penetration of PEVs in the vehicle market will range from 6% to 30% [24], which could reduce the overall greenhouse gas emissions by up to 5% [67].

However, a high penetration of PEVs charging on an unprepared power grid may impact grid stability [24]. According to the National Household Transportation Survey [64], the average PEV will need around 9 kWh of energy each day to drive the necessary amount; in comparison, the average household in the United States used around 31.5 kWh per day in 2010 [66]. Thus, a vehicle charging at home could increase the electricity demand of a household by almost one-third. EPRI predicts that if PEVs are charging during the summer peak hours, a 30% penetration of PEVs can lead to a necessary increase in total grid generation capacity of 5–6% [24]. (A similar statistic can be reached from assuming a 29% increase in residential electricity load, which currently constitutes around 38% of the total average demand [65].)

An efficient market mechanism could be created to charge PEVs during times that are optimal for the electric utilities, i.e., by facilitating charging of vehicle batteries to occur in off-peak hours. PEVs currently on the market are able to plug into the electricity grid and control which times during the connection period the vehicle battery will actually charge [8], allowing consumers to charge their vehicles when the price of electricity is low. In the next few years, PEVs may also be capable of transmitting electricity back to the grid (known as discharging).

Similar systems for selling electricity back to the grid currently exist for consumer-owned photovoltaic systems that generate surplus electricity, where consumers are paid an amount proportional to the electricity transmitted to the grid [9]. An analogous arrangement could exist for battery discharging; such a transfer would most likely result in a monetary profit for the consumer. Vehicles charging during off-peak hours and discharging during peak hours can also help decrease the gap between electricity demand peaks and valleys, which is a key feature in maintaining stability of the power grid.

In Part I of this work, we design a robust and distributed algorithm that can utilize the batteries of PEVs to benefit both vehicle owners and electric utilities. Vehicles benefit from charging at the minimal cost, and utilities benefit from a reduced impact on the peak electricity demand. We assume that vehicles will be capable of both charging and discharging their batteries from the electricity grid. A Linear Program (LP) is used to model the energy demands of the vehicle fleet, and provides a *composite valuation* that represents the combined benefit of electricity trading to consumers and utilities in each hour. We use this set of hourly values to construct and simulate an algorithm that dynamically assigns robust charging and discharging schedules for each vehicle in the fleet.

For example, suppose a vehicle is plugged in from 6pm until 7am, giving an 13-hour window in which the vehicle battery is available to the grid. If the battery can obtain sufficient charge in one hour, this charge can be scheduled to occur during an ideal hour between 6pm and 7am, which takes into account the electricity price, electricity demand, vehicle characteristics, and expected driving schedule. The remaining 12 hours could be spent idle, or they could be spent transferring energy to and from the grid: storing electricity during periods of low power demand and transmitting energy back to the grid when the demand is high.

We propose a mechanism to determine lowest-cost charging and discharging schedules that meet load curtailment requests from an electric utility, i.e., minimizing the increase in peak electricity demand that may result from PEV charging. A clustering of driving patterns from a previous similar day is used to predict the energy demands of the fleet. Our algorithm depends on a unique composite valuation that measures the benefit of electricity trading in each hour. The composite valuation is constructed using real data on driving schedules, electricity pricing, and load predictions. We attempt to create a procedure that can realistically be used by an aggregator of a fleet of PEVs, and use real data to produce our results.

Our algorithm is robust, as it ensures that each vehicle will have enough energy to drive given unexpected scenarios. It is also dynamic, in that vehicles are assigned charging and discharging schedules at the moment they plug into the grid. And finally, our algorithm is distributed, because each individual vehicle will have sufficient information to determine an energy exchange schedule without requiring back-and-forth communication with the aggregator and the remainder of the fleet.

### 1.1.1 Online Linear Programming

Online optimization makes dynamic decisions with the intent of minimizing a given objective function while the problem data reveals itself sequentially over time. In other words, the decision variables of the optimization problem are dynamically constructed as the problem data becomes available. Online linear programming is a subset of online optimization, in which both the objective and the constraints of the dynamically revealed static problem are linear.

In order to make online decisions based on little information and attempt to minimize a given objective, one can use the dual variables of a related optimization problem. Dual variables are also known as shadow prices for each constraint, since they represent the improvement to the objective value given a relaxation in the constraint. That is, they represent the marginal price of each good for the given objective. Thus, dual variables from a similar optimization problem can be a helpful measure for making near-optimal online decisions.

The dynamically constructed decision variables should also be generated with the intent of satisfying the constraints of the corresponding static optimization problem. The uncertainty in online optimization problems makes it difficult to ensure that the constraints of the static problem will be met. However, the formulation can be made robust by ensuring that the decision variables will be feasible in even the worst-case scenario.

We can model the economic network of a fleet of plug-in electric vehicles and the electricity market with a static linear program. The constraints of this LP ensure that the energy demand of the vehicles in the fleet are met, and that the impact of PEV charging on the electricity load is optimized. The decision variables are the optimal schedules for charging and discharging each vehicle battery, and the objective function minimizes the cost of charging the fleet of vehicles. The goal of this LP is to meet the demands of the vehicles and the electricity market, while charging the fleet at the lowest possible cost.

The input data for the LP is initially unknown, including the demands of the vehicles, the electricity demand, and when vehicles will be plugged-in to the grid. Because this information becomes available over time (as vehicles plug into the grid and the estimates of electricity load become more accurate), we can use online linear programming techniques to attempt to assign the optimal charging and discharging schedules for each vehicle. Our algorithm depends on the shadow prices resulting from a related LP and is robust to unexpected events in the energy demands of the fleet.

## *1.2 Sensor-equipped Vehicle Network*

Sensor-equipped vehicles may be able to detect nearby vehicles and objects to improve safety or traffic flow. Several vehicles currently on the market have sensors to assist with parking or to detect nearby vehicles in a blind spot [12,45]. In the next few years, sensors may exist that can use distance information to reduce traffic congestion and prevent collisions [13, 33]. Future luxury vehicles may come standard with such technology, but current vehicles that are not equipped with sensors can be retrofitted with accurate sensors, allowing even older vehicles to be capable of such features. This sensor technology could also be implemented in potential autonomous vehicles [43].

A key requirement for vehicle collaboration is the ability to determine the exact locations of a set of vehicles. Sensors can detect the distance to objects within 600 meters at an accuracy of ±1 centimeter [53], and use this set of distances to determine the relative location of a group of vehicles in a given area.

The results in Part II establish conditions on the sensors of SVs to ensure that the set of known distances can be used to determine the relative vehicle locations. We can apply these conditions to determining the density, radio range, and placement of sensors on SVs and the surrounding road.

### 1.2.1 Sensor Network Localization

The class of problems that determines the locations of objects given a subset of the pair-wise distances is known as Sensor Network Localization (SNL). A sensor network is a collection of *sensors* whose locations are unknown, and *anchors* whose locations are known. Each sensor (and anchor) can detect the distance from itself to a subset of the remaining sensors (and anchors) in the network. An SNL problem attempts to determine the exact locations of the sensors, given a set of distances between some of them.

Sensor network localization is related to the graph realization problem, which considers a weighted graph $G = (V, E)$, where each edge $(i, j) \in E$ has a non-negative weight $d_{ij}$. The graph realization problem aims to establish a realization of $G$ in Euclidean space $\mathbb{R}^d$, for a dimension $d$. That is, a solution to the problem will map the points in the graph $G$ to locations in $\mathbb{R}^d$ so that the Euclidean distance between every pair of adjacent vertices is equal to the corresponding edge weight.

The SNL problem, and similarly the graph realization problem, can be formulated as a set of quadratic norm constraints that is difficult to solve. However, a semidefinite programming (SDP) relaxation is an efficient computational method to solve an SNL problem, and often provides the exact localization of the sensors. In Part II of this work, we establish conditions on the density and structure of sensors in a given area to ensure that the SDP will accurately determine their relative locations.

We assume that anchors are placed on static objects surrounding the road, as this allows more control over the sensor network. The likelihood that the set of known distances between sensors will map to a unique set of sensor locations increases with the density of sensors and the number of distances known between them.

In Chapter 7, we provide the first nonasymptotic lower bound on the required radio range of randomly distributed sensors in a given area that ensures our localization method provides the correct locations of the sensors with high probability. A decomposition of the area into smaller subregions is used to determine the likelihood that, for a given density of sensors in dimension $d$, the graph induced from the network is a $(d+1)$-lateration graph and the network is uniquely localizable.

In Chapter 8, we show that for a certain class of sensor networks, a certificate of unique localizability can be produced to ensure the there is only one realizable set of sensor locations in all dimensions. We construct a novel matrix purification algorithm to prove that if the induced graph on $n$ sensors in dimension $d$ has points in general position that form a $(d+1)$-lateration graph, then the matrix solution to the dual of the SDP relaxation has rank $n$, and hence the matrix solution to the SDP will have rank $d$ and produce the correct sensor locations.

# Part I

# Network of Plug-in Electric Vehicles

# 2 Background

Plug-in electric vehicles are currently available on the general market, including the Chevrolet Volt, Nissan Leaf, and Tesla Roadster. These vehicles have the potential to become popular among consumers; the cost of running a vehicle on electricity is around one-third the cost of running a vehicle on gasoline [28], and the direct emissions of greenhouse gases is around two-thirds that of a standard vehicle with an internal combustion engine [22].

As of December 2011, the number of Chevrolet Volts and Nissan Leafs on the road in the United States totaled around 15,000 [69], which is less than 0.01% of the estimated 250 million passenger vehicles on the road [10]. The penetration of PEVs is expected to increase to between 6% and 30% in the next two decades [23], which may result in a needed increase in total grid generation capacity.

However, this impact on the grid could be mitigated with a mechanism that implements *smart charging* of PEVs, i.e., by facilitating charging of vehicle batteries to occur in off-peak hours. This could reduce the potential increase in grid generation capacity required to account for PEV energy demands, and could decrease the gap between electricity demand peaks and valleys. Distributed trading of energy between PEVs and utilities could also benefit consumers, by minimizing their costs. Moreover, new business and service industries could be created to manage such energy trading.

In Part I, we design and implement an online, distributed and robust algorithm to determine near-optimal electricity trading between PEVs and the electricity market. Our algorithm models the energy demands of a fleet of vehicles with a linear program, and uses online linear programming techniques to determine charging and discharging schedules of PEVs.

## *2.1 Vehicle-to-Grid Services*

The term *Plug-in Electric Vehicles* refers to the inclusive set of Battery Electric Vehicles (BEVs) that have no gasoline tank and only an electric battery, and Plug-in Hybrid Electric Vehicles (PHEVs) that have both a gasoline tank and an electric battery that receives power from the electricity grid. (For specifications of vehicle characteristics, see Table 5.2.)

**Definition 1.** *A* Plug-in Electric Vehicle *(PEV), or Plug-in Electric Car, is any vehicle that uses electricity from the grid as an alternative to liquid fuel.*

We assume that PEVs have flexible charging times, and that they can transmit electricity back to the grid, i.e., provide vehicle-to-grid (V2G) services (see [63] for more information on V2G technology). The idea of vehicles giving electricity back to the grid may seem counterintuitive, yet this capability could lead to great benefits. Simply put, the consumer could make a profit from selling electricity back to the grid when electricity prices are high, and buying electricity to charge when the price of electricity is low. For example, the baseline summer time-of-use (TOU) rates in the Pacific Gas and Electric (PG&E) transmission area range from $0.09 per kilowatt hour (kWh) during off-peak hours to $0.27 per kWh during peak hours; this could lead to a profit of $0.18 per kWh for a vehicle owner by discharging the vehicle in a peak hour and charging during an off-peak hour.

Moreover, the utility companies could use a large number of vehicles connected to the grid as distributed energy storage devices, which may be used to reduce the gap between the electricity demand peaks and valleys. This exchange of energy can be viewed as electricity trading that would benefit both the consumers and the electric utilities.

**Definition 2.** *A* fleet of Plug-in Electric Vehicles *is a large group of PEVs; we consider a fleet to be around 10,000 vehicles.*

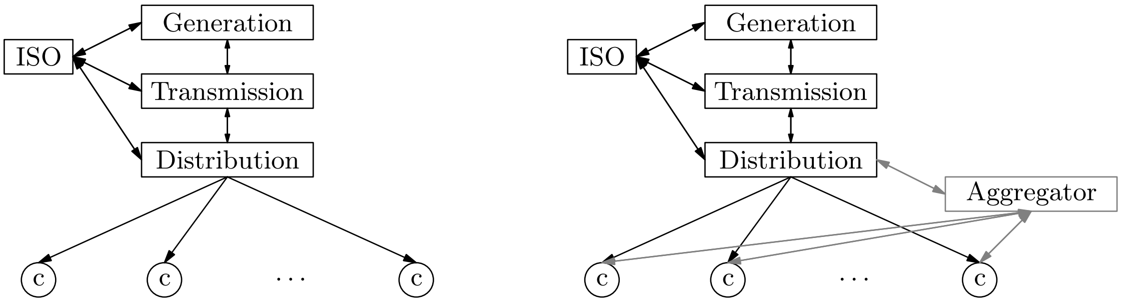
**Definition 3.** *A fleet of vehicles* provides grid services *if the vehicle batteries charge and discharge in a way that benefits the electric utilities.*

The algorithm constructed in this work schedules charging and discharging of individual vehicles in a fleet with the objective of maximizing the profit of the fleet vehicle owners, while providing grid services. Our algorithm can be viewed as an automated Demand Response (DR) mechanism for a fleet of PEVs that could define the role of electric vehicles in a smart grid. It is expected that such a DR service will be managed by an aggregator (an entity between fleets of vehicles and utilities). The aggregator will schedule an exchange of energy between the fleet and the utility such that each vehicle has a sufficient amount of energy to drive, the cost to the PEV owner is minimal, and the utilities benefit from a reduction in the gap between peak and off-peak electricity demands.

## 2.2 The Aggregator

In order for the demands of both PEV owners and electric utilities to be taken into account, there must be implicit communication between the consumers and the system dispatcher. It is commonly thought that there will be a third party aggregator to serve as a middleman who communicates the cumulative demands of the vehicle fleet to the electric utility, as opposed to individual vehicles separately communicating with the utility [46].

Note that current owners of photovoltaic systems may communicate individually with the grid to sell potential surplus electricity. However, the amount of power generated by photovoltaic systems cannot be controlled, and is simply determined by the amount of sunlight at the given time. On the other hand, the demand and supply of energy via PEVs does not have the same dependence on external forces. PEVs may be plugged-in overnight, requiring about 4 hours to fully charge an empty battery with a 240-volt supply; these vehicles can then provide grid services for the remainder of the time. An aggregator can collaborate with the system dispatcher and the consumers to determine energy exchange schedules that meet the needs of both parties.



(a) Current Market

(b) Future Market

Figure 2.1: The Current and Future Infrastructure of the Electricity Grid

**Definition 4.** *An* aggregator *manages the communication and electricity distribution between a group of electricity consumers and an electric utility.*

In the current system, an electric utility manages three sectors that cooperate to deliver electricity to consumers: generation, transmission, and distribution; we refer to the cumulation of these entities as the *grid*. The generation sector generates electricity to meet the demands of consumers; the transmission sector transfers electricity in bulk from the generators at high voltages; the distribution sector uses transformers to reduce the voltage to the level appropriate for actual use, and carries electricity from the transmitters to consumers.

We design an algorithm that can be used by an aggregator of a fleet of vehicles. As shown in Figure 2.1, the aggregator is envisioned to be a new service in between the distribution sector (and load serving entity) and the consumers, where each $c$ in the diagram represents a consumer. The independent system operator (ISO) currently monitors and regulates the wholesale of electricity between the generation and transmission sectors; an aggregator would monitor the buying and selling of electricity between the distribution and the consumer level.

On a lower level, the system between the aggregator, PEV consumers, and electricity market is illustrated in Figure 2.2. The general idea is that the aggregator will collaborate with the market to determine an acceptable load curtailment request for the fleet, using information on electricity pricing and an estimate of the vehicles' energy demands. The aggregator then communicates information to the vehicles, which will inform the PEVs on the best times to provide grid services.

Currently, there are aggregators in the system that regulate consumer electricity use by rewarding decreases in electricity demand during certain time periods, in order to meet a load scheduling obligation to the market (or system dispatcher). These DR resources are managed by either third-parties (such as EnerNOC and ZigBee Smart Energy) or the utilities themselves (for example, both PG&E and Duke Energy have DR programs, among many others). Existing programs focus on the Commercial and Industrial Sector, having a small number of large customers who typically

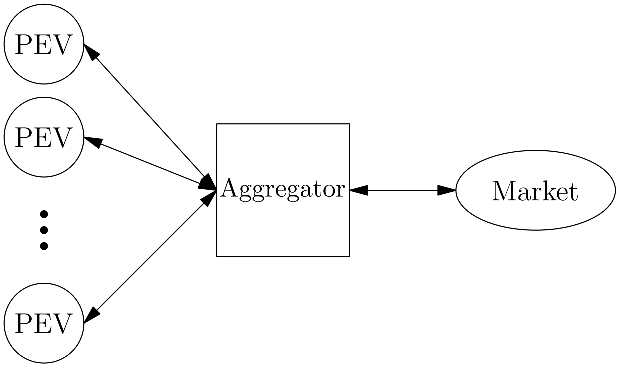


Figure 2.2: Low Level Representation of Aggregator Network

use hundreds of kilowatt hours per day [65]. These programs are not automated, requiring human intervention to tailor the DR resource for each customer and to monitor customer electricity usage alongside the electricity load and pricing.

By contrast, some estimates say there may be 100 million PEVs on the road in the United States by 2030 [25], each with a maximum charging or discharging rate of about 6.6 kW. A V2G system will require unique management oversight for a mutual electricity transfer between the grid and such a large number of participants with relatively small individual contributions. Thus, DR services for a fleet of PEVs providing unprecedented energy exchange cannot be individually monitored for each vehicle. Aggregators need an automated mechanism to facilitate charging and discharging of fleet vehicles to meet their energy needs while satisfying a scheduling obligation to the market.

We construct a mechanism that can be implemented by a device attached to each vehicle in a fleet; these devices can be controlled by an aggregator to regulate vehicle charging. Installing the device on a vehicle could benefit the consumer by reducing the cost of charging, and could benefit the electric utilities by meeting a commitment to the market, such as not increasing the peak power demand or maximally charging vehicles in off-peak hours. Such a device is a practical extension of services that are currently provided with PEV charging stations, which allow the user to control charging but require settings to be determined by the user [8].

It should also be mentioned that such managed charging is usually envisioned to be an ‘opt-in’ service, in which a consumer has the option to participate, but is not mandatory.

## 2.3 Algorithm Structure

The structure of our algorithm is illustrated in Figure 2.3. The aggregator will collect information from the market to form a linear programming model of the energy flow throughout the network shown in Figure 2.2. Information from this LP is then obtained and passed to the individual vehicles, who implement the dynamic algorithm to determine their charging and discharging schedules.

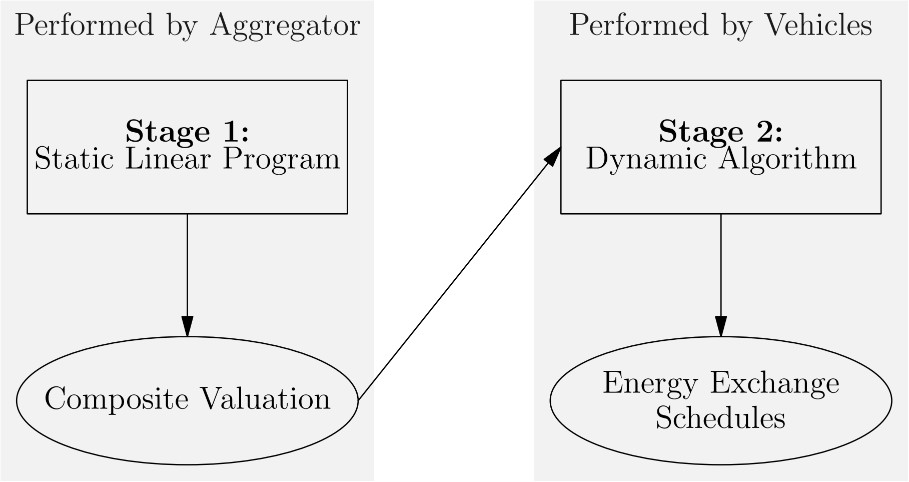


Figure 2.3: Structure of Algorithm

Specifically, the aggregator receives information from the market on electricity demand and pricing, and estimates the fleet energy demands by clustering similar driving patterns from a previous period. This information can be used to form a linear programming model of the energy flow between the fleet and market, which determines the load curtailment requests. A dual variable from this static LP is used to form a composite valuation that is sent to each individual vehicle. Given this composite valuation, each PEV will have enough information to dynamically determine a charging and discharging schedule that will be of minimal cost and ensure that the fleet as a whole is meeting the load schedule obligation from the market.

The resulting energy exchange schedules are made robust by ensuring each vehicle has sufficient energy in its battery to meet an uncertain (but bounded) driving load. The algorithm is also run on a rolling horizon, computing new charging and discharging schedules periodically to account for changes in the vehicle driving schedule.

Chapter 3 constructs the linear programming model of the fleet, Chapter 4 provides the proof that a given dual variable of the linear programming model can be used to construct a composite valuation, and Chapter 5 explains the dynamic algorithm used to assign energy exchange schedules to the vehicles.

## *2.4 Related Work*

The challenge of charging plug-in electric vehicles from the grid in a beneficial way is a relatively recent field. However, there have been several works in the past few years that address this question, the majority of which use an optimization formulation to model a fleet of PEVs from the perspective of the aggregator.

Ma et al. [42] designed an algorithm for back-and-forth communication between vehicles and the aggregator that will converge to an optimal set of prices under the Nash Uncertainty Equivalence. That is, if the electricity price strictly increases with demand and each individual vehicle solves a series of static problems that account for the demands of the fleet as a whole, then the charging schedules of the fleet will optimally fill valleys in the electricity demand. This algorithm is not online, requiring repeated communication between the vehicles and the aggregator, and does not consider vehicle-to-grid capabilities.

The work of Wu et al. [71] builds an online algorithm to charge vehicles in the hours with the lowest possible electricity price. Real data is used to test the algorithm, although the potential benefit to utilities and vehicle-to-grid are not considered.

Sioshansi and various co-authors have a number of works that consider the overall benefits of energy storage and PEVs, including [54–57, 72]. The majority of these works use real data and construct an optimization formulation of the problem to determine the environmental, monetary, or societal impacts of a power grid that incorporates PEVs and distributed energy storage devices. The implications of these works include showing that the pollution output of PEVs can be less than or equal to that of conventional vehicles at the same cost to consumers, comparing the welfare benefit of energy storage owned by aggregators against storage owned by consumers, and determining the benefits to utilities if a fleet of PEVs were to provide ancillary services given a number of assumptions (e.g., varying gasoline prices).

There have also been works that use a fleet of PEVs to provide frequency regulation; for example, in 2010 Han et al. [34] constructed an algorithm to provide frequency regulation with a fleet of PEVs to maximize the revenue for the fleet.

### 2.4.1 Our Contribution

The contribution of our work to this field is three-fold. First, our algorithm is *distributed*, in that vehicles determine the best energy exchange schedules individually in an online fashion. Second, our algorithm is *robust* to uncertainty in driving schedules of each vehicle. And finally, our algorithm uses a unique set of *composite values* that equally accounts for the benefits to electricity consumers and utilities.

# 3 Linear Programming Formulation

We approach this problem from the perspective of the aggregator, who makes commitments to both the PEV consumers and their utility. The aggregator formulates a static LP based on information about electricity pricing, electricity demand, and driving schedules, and uses a dual variable from this LP to construct a composite valuation. The composite valuation is then passed to individual vehicles, so that each PEV can determine an energy exchange schedule that is acceptable to both consumers and utilities.

We do not assume the aggregator knows the driving loads of vehicles ahead of time– the availability and driving patterns of each vehicle will be learned as the vehicles plug-in in an online fashion. Once a vehicle connects to the grid, a charging and discharging schedule is determined from the composite valuation without knowing the behavior of future vehicles that may plug into the grid. We use online linear programming techniques to ensure the resulting schedules are near-optimal, or close to the schedules that would have been assigned had all information about the energy demands of the fleet been known in advance.

In our model, an *optimal* set of charging and discharging schedules for a fleet of vehicles will:

- minimize the cost (or maximize the profit),
- ensure each vehicle battery has sufficient energy to meet the driver's needs,
- and meet a *market obligation*.

## *3.1 Market Obligation*

From the perspective of an electric utility, the electricity consumed by a fleet of vehicles should have minimal impact on grid performance. Specifically, we consider the impact PEV charging can have on the generation sector; if a large number of vehicles are simultaneously charging, a new peak demand can be created during off-peak hours or the peak demand can drastically increase, which may result in a need to build more generators. Similarly, from the point of view of the distribution sector, the transformer that reduces the electric voltage to the appropriate amount for households is only built to sustain a certain amount of electricity. Since the demand of each vehicle may be close to one-third of the demand of a household, the increased electricity usage resulting from PEV charging may decrease the reliability of these transformers [52].

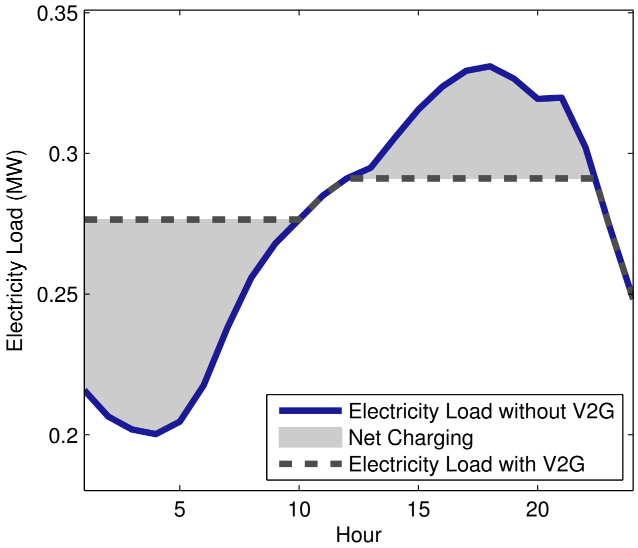


Figure 3.1: Example of Valley-filling Charging and Peak-shaving Discharging

However, if a smart design is used to provide grid services with PEV fleets, this potential strain on the grid may be either removed or reduced. In order to ensure the fleet of vehicles is exchanging energy with minimal impact on the grid, a limit, or constraint, needs to be placed on the total amount of charge done by a fleet.

**Definition 5.** *A fleet of vehicles will meet a* market obligation *if the total net charging stays below an upper limit set by the aggregator, in conjunction with the electric utility.*

If the batteries of a vehicle fleet were to charge during electricity demand valleys and discharge during demand peaks, they could provide grid services by narrowing the gap between the maximum peak demand and the minimum off-peak demand. This is illustrated in Figure 3.1.

Such valley-filling and peak-shaving could be extremely beneficial to the power grid. According to the National Renewable Energy Laboratory (NREL), using distributed energy storage devices, such as PEVs, to provide grid services will have a number of advantages [40]:

> Distributed energy also has the potential to mitigate congestion in transmission lines, reduce the impact of electricity price fluctuations, strengthen energy security, and provide greater stability to the electricity grid.

The aggregator can meet a load curtailment request from the utility by ensuring that the exchange of energy with a fleet of PEVs will meet a market obligation. This market obligation can take the form of an upper bound on the total amount of charging done by the fleet during a given time period.

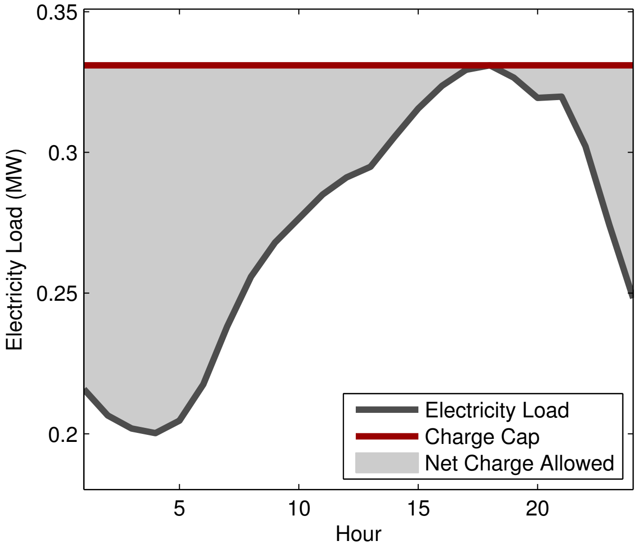


Figure 3.2: Example of a Charge Cap at the Peak Load Value

### 3.1.1 Charge Cap

Because the availability and demands of vehicles in a fleet cannot be controlled, it may not be possible for a fleet of vehicles to precisely fill demand valleys and shave demand peaks. In other words, it may not be possible to require a fleet of PEVs to charge more than a set lower limit during off-peak hours and discharge more than a given amount during peak hours. Therefore, in our formulation we represent a market obligation as an upper bound constraint on the total amount of hourly electricity added to the load by fleet charging. That is, during each time period considered, an upper bound is placed on the total amount of electricity that can be supplied to the fleet.

**Definition 6.** *We say that the market places a* charge cap *on the total amount of charging that can be done by the fleet in each time period, in order to stay within a given upper limit.*

The value of this charge cap can be agreed upon between the electric utility and the aggregator, while ensuring the vehicles can charge the required amount. For example, say the aggregator manages a neighborhood of 100 homes with 30 PEVs. On a given day, the charge cap could to be set to the value of the maximum peak demand of the 100 homes (without PEV charging) on that day. Given that the vehicles could sufficiently charge and the total fleet charging stays below the charge cap, no new peak will be created. Figure 3.2 illustrates this example.

In our simulations, we assume that time is split up into hourly periods, so that each hour has a charge cap and base consumer demand. (We henceforth consider time in hourly periods, but note that this discretization can be easily adjusted.) For an $n$-hour period, let $\texttt{elecLoad} \in \mathbb{R}^n$ represent the base electricity load, and let $\texttt{chargeCap} \in \mathbb{R}^n$ represent the charge cap for each hour. To meet the market obligation in hour $h$, the amount of charging done by the fleet must be at most

$$\texttt{chargeCap}_h \texttt{ - } \texttt{elecLoad}_h.$$

This is represented in Figure 3.2 as the *net charge allowed*, or the maximum amount of net charging done by the fleet to meet the hourly market obligation. When vehicles only charge from the grid (that is, when they do not participate in vehicle-to-grid services), this charge cap sets an upper limit on the total amount of charging that can be done in a given hour. However, when vehicles participate in V2G, this will set an upper bound on the amount of *net* power obtained from the grid. In other words, a subset of the fleet vehicles could charge in the hour of peak demand, as long as other vehicles in the fleet were simultaneously discharging a sufficient amount. We formulate an LP that models the expected fleet energy demands and the market obligation (i.e., the charge cap), with the objective of minimizing consumer cost. The aggregator can solve this LP to obtain the composite valuation, which will be distributed to the PEVs in the fleet.

In our model, the charge cap is an output of the LP formulation. That is, the aggregator uses the LP to determine not only the composite valuation, but also the best possible charge cap. Given the information from the electricity market, the aggregator can find the smallest possible charge cap value that ensures the fleet energy demands are met.

## *3.2 Vehicle Transport Needs*

A fleet of vehicles may be used to meet a market obligation, but their main purpose will be to store energy for the vehicle to drive. Our algorithm ensures that each vehicle has sufficient energy to meet its driving demands, or *transport load*.

During any hour, the vehicle may be partaking in exactly one of four actions: charging, discharging, driving, or idling (i.e., neither charging nor discharging). The vehicle is only able to charge or discharge when it is plugged-in, and is only able to either charge or discharge in each hour (i.e., the vehicle can either charge or discharge in a given hour, not both). We assume that vehicles will connect to the grid (or plug-in) each time they are parked for at least one hour.

Vehicles are assumed to plug-in periodically over a given period of time (say, 12 hours) and report their tentative driving schedule for the next $n$ hours. We assume the information on the vehicle specifications can be obtained (such as charging and discharging rates, battery capacities, etc.) and that the aggregator has determined an appropriate charge cap to place on the total charge done by the fleet in each hour.

The output of our algorithm is an *energy exchange schedule* for the considered $n$ hour period, which determines during which hours the vehicle should charge and discharge, until the end of the considered time horizon. The length of this time horizon can easily be adjusted, but in our results we consider a time horizon of 120 hours, or 5 days into the future. We use the time horizon of

$n = 120$ to allow each vehicle's state of charge to evolve sufficiently to ensure the initial assumptions about state of charge do not affect the final results.

In our simulations of the algorithm, vehicles plug into the grid at various times over a given period, and the schedule of each vehicle will be determined instantly as it plugs-in. For example, if all cars plug-in over a 12-hour period (between Hour 1 and Hour 12) and we are considering a 5-day period (from Hour 1 to Hour 120), each schedule will start at the hour the vehicle plugs-in and end in Hour 120. The optimization problem (LP-fleet) is solved once at the start of the 5-day period.

The energy exchange schedule for each PEV should

1. be determined instantly as vehicles plug into the grid,
2. provide enough charge to each individual PEV to meet its driving load,
3. ensure the net charge of the fleet meets the market obligation, and
4. be of minimal cost (or maximal profit) to the consumers.

Our algorithm to construct such energy exchange schedules achieves all of the above for a time horizon of hour $n$. We assume a few hundred (say, around 400) driver behaviors from a similar day in a similar region are known, in addition to the predicted electricity load for the next $n$ hours.

## *3.3 Clustering*

If the aggregator had perfect foresight of each vehicle's future transport load, energy exchange schedules could be allocated to each individual vehicle that solve a static optimization problem; this static problem would minimize the cost to the fleet, while making sure the market obligation is met and each vehicle battery has enough energy to meet its transport load.

However, the aggregator does not have perfect foresight. Moreover, it would be far too complicated to keep track of what each individual vehicle may *possibly* do and an LP with constraints for each individual PEV to meet its energy demand is computationally intractable because it is extremely large. Therefore, we use clustering to estimate the expected future demand. The purpose of clustering is to model the expected energy demands of the fleet of vehicles for the given period.

**Definition 7.** *A* base driving profile *is a set of driving distances, departure times, and arrival times over a 24-hour period that constitute a general driving pattern that is similar to the driving patterns of a relatively large number of drivers. In other words, it is the "best fit" to a driving schedule of a group of drivers with similar driving patterns.*

Given a time horizon $n$, we can cluster driving patterns from a previous similar $n$-hour time period to establish a number of distinct base driving profiles. A previous similar period, with the same temperature and in the same region, can be considered. These base driving profiles will then be set to the *centroid* of each cluster. Here, the centroid is the mean of all driving patterns that form the cluster.

Each cluster will represent a general driving pattern, and given the density of each cluster, we can estimate the demand for the considered time period. We take driving patterns from a previous similar period and form $k$ clusters from the data, each with a weight $b_\ell$. This weight takes into account the expected number of vehicles associated with each base driving profile, and also the expected size of the vehicle batteries in each cluster. The weights are scaled so that, for $m$ vehicles, $\sum_{\ell=1}^{k} b_\ell = m$.

These clusters and their respective weights can then be used in an LP model of the estimated fleet energy demands and market obligation. The primal solution to the linear program (LP-fleet) contains an energy exchange schedule for each base driving profile for the given time period; these energy exchange schedules will minimize the total consumer cost, meet the market obligation and satisfy the transport loads of each base driving profile.

However, since the demands and availabilities of each individual vehicle are distinct, the dynamic algorithm uses information provided by the composite valuation. That is, the solution to (LP-fleet) is not individualized for each vehicle, so the information encoded in a dual variable of (LP-fleet) is used to assign energy exchange schedules to individual vehicles instantly as they plug into the grid. By using the composite valuation, the net charge of the fleet will meet a market obligation and incur the lowest-cost to consumers.

### 3.3.1 Clustering Method

The output of the clustering method is three-fold: a set of base driving profiles (or the centroids of each cluster formed), the variance of driving patterns in the cluster, and the expected number of fleet vehicles in each cluster and their battery sizes. An individual driving pattern of a vehicle $j$ can be represented in a vector format:

$$t_j = \left[t_{j,1}, t_{j,2}, \cdots, t_{j,24}\right] \in \mathbb{R}^{24},$$

for $j = 1, \ldots, m$, where $m$ is the number of drivers considered. To find the base driving profiles, we use the $k$-means clustering algorithm [41] with Euclidean distances on these transport load vectors.

In our simulations, we used real driving behaviors obtained from the National Household Travel Survey (NHTS) [64]. This survey has been conducted every 5–8 years since 1969; we use the most recent survey from 2009, which is an organized and well-documented dataset containing the travel schedule of over 150,000 households, each for a 24-hour period [26]. This dataset is *not* a collection of trips made by plug-in electric vehicles; thus, by using this data, we are assuming that driving patterns with PEVs will be the same as driving patterns of internal combustion engine (i.e., standard gasoline-powered) vehicles. We cluster 400 driving profiles from a similar time period (i.e., same day of the week in the same region) in order to estimate the expected hourly demand of a fleet of 10,000 PEVs.

Figure 3.3 shows the cluster valuation metric we used to determine the number of clusters to form. Our cluster valuation metric is defined as follows. Each cluster has a centroid (or mean),

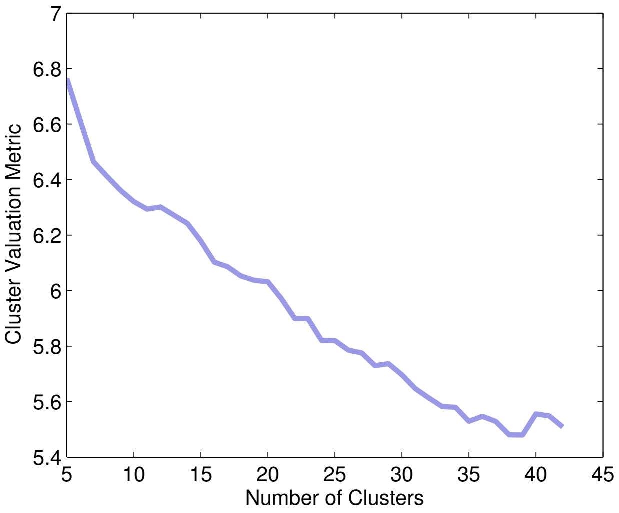


Figure 3.3: Cluster Valuation Metric for Given Numbers of Clusters Using $k$-means

and we measure the distance from each point in the cluster to its centroid. We then find the mean of these distances among all clusters, say the overall-cluster mean. Since $k$-means clustering is a randomized algorithm, we perform 100 runs of the algorithm on different fleets of transport loads $t_j$, and take the statistical mean of the overall-cluster mean from each run; the logarithm of this value is our cluster valuation metric on the $y$-axis of Figure 3.3, for the range of 5 to 45 clusters.

As Figure 3.3 shows, the valuation metric of within-cluster means stops decreasing at $k = 37$ and no more than 43 clusters can be formed. Thus, in our simulations we form 37 base driving profiles. (Note that an aggregator could use a similar metric to determine the number of clusters to form given data from a previous similar period.)

For simulation purposes, each transport load from the NHTS dataset was assigned to either a PHEV or a BEV: if the daily transport load amounted to less than 70 miles, the corresponding vehicle was assumed to be a BEV; otherwise, it was assumed to be a PHEV. (Realistically, the vehicle type would be information inherent to the vehicle.) BEVs and PHEVs were separated before clustering so that vehicles corresponding to the same base driving profile are either all BEVs or all PHEVs. As a side note, 91.6% of vehicles in the subset of data we used for simulation (the subset of driving patterns from urban California) were driven less than 70 miles in the 24-hour period.

To determine the expected density of each cluster, we assumed that the portion of vehicles associated with each base driving profile would be the same as the portion of vehicles from the sampled 400 in each cluster.

## 3.4 Linear Programming Formulation

Let $k$ be the number of clusters and $n$ be the time horizon in hours. The vector $t_\ell \in \mathbb{R}^n$ represents the base driving profile of each cluster $\ell$, where the $h$th element $t_{h\ell}$ is the number of miles driven in hour $h$. Similarly, $\sigma_\ell \in \mathbb{R}^n$ is the variance of the driving patterns in cluster $\ell$. In our simulations, we let $k = 37$ and $n = 120$, but these values can easily be changed.

Table 3.1 lists the decision variables of the linear program (LP-fleet), and Table 3.2 lists the input parameters to (LP-fleet), which are established by either the aggregator or the vehicle driver (as listed).

Table 3.1: List of Variables

| **Variable** | **Description** | **Dimension** |
|---|---|---|
| $s_\ell$ | electricity storage | $n+1$ |
| $c_\ell$ | charging schedule | $n$ |
| $d_\ell$ | discharging schedule | $n$ |
| $s_\ell^g$ | gasoline storage | $n+1$ |
| $f_\ell$ | fueling schedule | $n$ |
| $g_\ell$ | generation schedule | $n$ |
| $c_{\text{cap}}$ | charge cap | 1 |

Table 3.2: List of Parameters

| **Variable** | **Description** | **Dimension** |
|---|---|---|
| | **Established by Vehicle** | |
| $\bar{s}_\ell$ | storage capacity | $n+1$ |
| $\bar{c}_\ell$ | maximum charging rate | $n$ |
| $d_\ell$ | maximum discharging rate | $n$ |
| $\bar{s^g}_\ell$ | gas tank capacity | $n+1$ |
| $\bar{f}_\ell$ | maximum fueling rate | $n$ |
| $\bar{g}_\ell$ | maximum generation rate | $n$ |
| $s_{\ell 0}, s_{\ell 0}^g$ | initial battery and gasoline storage | 1 |
| $\alpha, \beta$ | kWh/mile, gallon/kWh | 1 |
| | **Established by Aggregator** | |
| $t_\ell$ | cluster base driving profile | $n$ |
| $p, p^g$ | electricity and gasoline prices | $n$ |
| $L$ | electricity load | $n$ |
| $\text{c}_{\text{eff}}$, $\text{g}_{\text{eff}}$ | charging and generating efficiencies | 1 |
| $\rho$ | scaling factor of charge cap in objective | 1 |
| $\sigma_{t_\ell}$ | variance of cluster transport loads | $n$ |
| $\sigma_L$ | variance of electricity loads prediction errors | $n$ |
| $\kappa_{t_\ell}, \kappa_L$ | scaling values for cluster transport and electricity load variance | 1 |

We form a static linear program with the base driving schedules of each cluster for the $n$-hour period. The primal solution to this static linear program will provide the charging, discharging, fueling and generating schedules, $c_\ell, d_\ell, f_\ell, g_\ell \in \mathbb{R}^n$, for each base driving profile $\ell = 1, \ldots, k$, that minimize the total consumer cost while meeting the market obligation and ensuring the transport load of each base driving profile will be satisfied. The solution also includes the optimal charge cap value, $c_{\text{cap}} \in \mathbb{R}$, that will ensure the energy needs of the fleet are met while minimizing the impact on electricity load. This formulation acts as a proxy for the aggregate behavior of the actual fleet. We use Matlab vector notation, where $x_{a:b}$ represents the elements $\{a, a+1\ldots, b\}$ of $x$.

$$\begin{array}{lll}
\underset{c_{\text{cap}}, c_\ell, d_\ell, f_\ell, g_\ell}{\text{minimize}} & \sum_{\ell=1}^{k} b_\ell \left(p^T(c_\ell - d_\ell) + (p^g)^T f_\ell\right) + \rho \cdot c_{\text{cap}} & \\
\text{subject to} & s_{\ell,1:n} = s_{\ell,0:n-1} + c_{\text{eff}} \cdot c_\ell - d_{\text{eff}} \cdot d_\ell + g_{\text{eff}} \cdot g_\ell - \alpha(t_\ell + \kappa_{t_\ell} \cdot \sigma_{t_\ell}^2) & : \lambda_\ell^s \\
& s_{\ell,1:n}^g = s_{\ell,0:n-1}^g + f_\ell - \beta \cdot g_\ell & : \lambda_\ell^{s^g} \\
& t_{\ell h} = 0 \implies f_{\ell h} = 0, g_{\ell h} = 0 & : \eta_\ell^1, \eta_\ell^2 \\
& t_{\ell h} > 0 \implies c_{\ell h} = 0, d_{\ell h} = 0 & : \eta_\ell^3, \eta_\ell^4 \\
& 0 \le (s_\ell, c_\ell, d_\ell, s_\ell^g, f_\ell, g_\ell) \le (\bar{s}_\ell, \bar{c}_\ell, \bar{d}_\ell, \bar{s}_\ell^g, \bar{f}_\ell, \bar{g}_\ell) & : \nu_\ell, \gamma_\ell \\
& \sum_{\ell=1}^{k} b_\ell(c_\ell - d_\ell) \le c_{\text{cap}} \cdot 1 - (L + \kappa_L \cdot \sigma_L^2) & : \theta
\end{array} \qquad \text{(LP-fleet)}$$

The first five constraints model the dynamics of energy flow in the PEV battery, the last constraint ensures the fleet energy exchange meets the charge cap. The objective minimizes the consumer cost and the charge cap value.

The first constraint updates the battery storage amount in each hour based on the amount of energy the vehicle charges, discharges, generates and uses to drive. Similarly, the second constraint updates the amount of gasoline storage in each hour based on the amount of energy generated from fuel and the amount the vehicle is fueled. The third and fourth constraints ensure that the vehicle can only fuel the tank and generate electricity from gasoline when the vehicle is driving, and that the vehicle battery can only charge or discharge when the vehicle is parked for at least an hour. The second-to-last constraint ensures the variables stay within their physical bounds. And the last constraint ensures that the total amount of hourly net charge meets the market obligation, or is below the charge cap.

The first and last constraints take into account the uncertainty of the variables $t_\ell$ (cluster $\ell$ transport load) and $L$ (electricity load), respectively. For the first constraint, the scalar value $\kappa_{t_\ell}$ is chosen so that a specified percentage of the driving patterns in each cluster are less than $(t_\ell + \kappa_{t_\ell} \cdot \sigma_{t_\ell}^2)$. For example, $\kappa_{t_\ell}$ can be chosen large enough so that the battery state of charge at each hour is expected to be meet the demands of 95% of all driving behaviors in cluster $\ell$.

Similarly, $L$ is the *expected* electricity load for the next $n$ hours, but we also take into account prediction errors. The vector $\sigma_L$ is the daily variance of electricity load prediction errors for a previous similar month. The scalar value $\kappa_L$ is chosen so that with a given probability, the realized electricity load will be less than $(L + \kappa_L \cdot \sigma_L^2)$. For example, $\kappa_L$ can be chosen large enough that the realized electricity load is expected to be less than $(L + \kappa_L \cdot \sigma_L^2)$ with 95% probability.

The variables on the far right in (LP-fleet) are the dual variables associated with each constraint, where

$$\lambda_\ell^s, \lambda_\ell^{s^g} \in \mathbb{R}^{n+1}, \quad \eta_\ell^1, \eta_\ell^2, \eta_\ell^3, \eta_\ell^4 \in \mathbb{R}^n, \quad \theta \in \mathbb{R}^n,$$

and

$$\nu_\ell = (\nu_\ell^s, \nu_\ell^c, \nu_\ell^d, \nu_\ell^{s^g}, \nu_\ell^f, \nu_\ell^g) \in \mathbb{R}^{6n+2}, \quad \gamma_\ell = (\gamma_\ell^s, \gamma_\ell^c, \gamma_\ell^d, \gamma_\ell^{s^g}, \gamma_\ell^f, \gamma_\ell^g) \in \mathbb{R}^{6n+2}.$$

Note that if we were to form an analogous micro-linear program from tens of thousands of potential *individual* travel profiles over a 5-day period (i.e., $n = 120$), the solution would be a set of schedules $(s_j, c_j, d_j, s_j^g, f_j, g_j) \in \mathbb{R}^{6n+2}$ for $j = 1, \dots, m$, where $m \approx 10,000$. In our testing, solving a problem of this magnitude takes a few hours on a single workstation.

In contrast, the formulation (LP-fleet) that determines one schedule for each of the 37 clusters is considerably smaller. In our simulations, forming the set of clusters took around 0.3 seconds and solving the clustered linear program (LP-fleet) can be done in about 3 seconds on a single workstation. Moreover, (LP-fleet) requires relatively few representative samples of driving patterns and does not require that the aggregator keep track of what each vehicle may *possibly* do.

## *3.5 Properties of the Linear Program*

The solution to (LP-fleet) will provide the optimal charging and discharging schedules for the set of the base driving profiles. However, because individual vehicles will have distinct driving patterns, we use information from the dual variables of (LP-fleet) to determine the energy exchange schedules for individual vehicles in each cluster.

### 3.5.1 Composite Valuation

Note that allocating charge to each vehicle in the hours with the lowest electricity price (as used in [71]) will guarantee the lowest cost to each driver, but in general will not meet a market obligation. For example, it can result in an increased peak electricity demand.

Thus, we create a set of hourly *composite values* that measures the overall benefit to the system of fleet vehicle charging in each hour. That is, there will be a value assigned to each hour that represents the marginal cost of charging and discharging, given the electricity price and the charge cap for that hour. A ranking of these values will produce an ordering on the best times for a vehicle in the fleet to charge its battery. We use the dual variable $\theta \in \mathbb{R}^n$ from LP-fleet to construct such a set of values. $\theta$ is the Lagrange multiplier for the linear constraint

$$\sum_{\ell=1}^{k} b_\ell(c_\ell - d_\ell) \leq \hat{c}_{\text{cap}}, \tag{3.1}$$

where $\hat{c}_{\text{cap}} := c_{\text{cap}} \cdot 1 - (L + \kappa_L \cdot \sigma_L^2)$. The vector $\theta$ is the set of $n$ *shadow prices* for each of the $n$ constraints in (3.1).

**Definition 8.** *A* shadow price *is the improvement in the objective function value per unit relaxation in the associated constraint.*

Let $\theta^\star$ be the shadow price of the constraint (3.1) at the solution, so that the value of $\theta^\star_h$ for $h \in \{1, \ldots, n\}$ is the reduction in the objective function value if the charge cap in hour $h$, $(c_{\text{cap}})_h$, were increased by a unit of 1. In our formulation, this corresponds to $\theta^\star_h$ being the improvement in the objective when $(c_{\text{cap}})_h$ is increased by 1 MW.

Define the function $u(\cdot)$ as

$$u(s, c, d, s^g, f, g) := \sum_{\ell=1}^{k} b_\ell \left(p^T(c_\ell - d_\ell) + (p^g)^T f_\ell\right),$$

and let $(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star, c^\star_{\text{cap}})$ represent an optimal solution to the problem (LP-fleet), given certain input parameters. Define $\hat{c}_{\text{cap}} := c^\star_{\text{cap}} \cdot 1 - (L + \kappa_L \cdot \sigma_L^2)$ to be the augmented charge cap value at the solution. Then the following equation defines $\theta^\star_h$:

$$\begin{aligned}\theta^\star_h &= \left(\frac{\partial u(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)}{\partial(\hat{c}_{\text{cap}})_h}\right) \Bigg/ \left(\frac{\partial\left(\sum_{\ell=1}^{k} b_\ell(c^\star_{\ell h} - d^\star_{\ell h})\right)}{\partial(\hat{c}_{\text{cap}})_h}\right) \\ &= \frac{\Delta u(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)}{\Delta\left(\sum_{\ell=1}^{k} b_\ell(c^\star_{\ell h} - d^\star_{\ell h})\right)}. \end{aligned} \tag{3.2}$$

That is, $\theta^\star_h$ is the relative change in the objective function value to change in the constraint (3.1), given a relaxation of the charge cap in hour $h$.

Intuitively, the value of $\theta^\star_h$ will be relatively large if there is a great value in increasing the upper bound value $(\hat{c}_{\text{cap}})_h$. That is, a large value of $\theta^\star_h$ implies a large demand for charging and a small supply (or a small amount of net charge allowed) in hour $h$. Therefore, the values of $\theta^\star$ represent the value (or marginal price) of fleet vehicle charging in each hour.

In the following Chapter 4, we prove that $\theta^\star$ can be used to in conjunction with the electricity price to obtain a set of values that factors in the benefits to both consumers and electric utilities.

# 4 Composite Valuation

The formulation (LP-fleet) in Chapter 3 finds charging, discharging, fueling, and generating schedules $(c_\ell, d_\ell, f_\ell, g_\ell) \in \mathbb{R}^n$ for each base driving profile $\ell = 1, 2, \ldots, k$. These schedules ensure that a vehicle with a transport load of the base profile will obtain sufficient charge in the hours with the lowest possible cost, while staying within the charge cap.

To determine energy exchange schedules for individual vehicles, we use the dual variables of (LP-fleet). Specifically, the dual variable $\theta \in \mathbb{R}^n$ can be used to construct a *composite valuation* that will be used in a dynamic algorithm to determine energy exchange schedules.

**Definition 9.** *Consider a fleet that consists of $k$ vehicles with driving schedules equal to the base driving profiles $t_\ell$, for $\ell = 1, \ldots, k$. A* composite valuation *is a set of hourly values $q^\star \in \mathbb{R}^n$ such that if each fleet vehicle were to charge in the hours corresponding to the elements of $q^\star$ that are the lowest and discharge in the corresponding hours when $q^\star$ is the highest, then: the cumulative fleet charging will satisfy the market obligation, the lowest feasible cost will be attained, and each base profile will have enough energy to meet its transport load.*

To explain a composite valuation more clearly, let $(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)$ and $((\lambda^s)^\star, (\lambda^{s^g})^\star, (\eta^1)^\star, (\eta^2)^\star, (\eta^3)^\star, (\eta^4)^\star, \theta^\star, \nu^\star, \gamma^\star)$ represent the optimal primal solution and associated dual variables from (LP-fleet). We treat the charge cap value as a constant determined by (LP-fleet), and consider the related linear program (LP-comp) that results from removing the constraint (3.1) from the formulation (LP-fleet) and changing the electricity price vector in the objective function to $q^\star \in \mathbb{R}^n$.

$$\begin{array}{lll}
\underset{c_\ell, d_\ell, f_\ell, g_\ell, \forall \ell}{\text{minimize}} & \sum_{\ell=1}^{k} b_\ell \left((q^\star)^T(c_\ell - d_\ell) + (p^g)^T f_\ell\right) & \\
\text{subject to} & s_{\ell,1:n} = s_{\ell,0:n-1} + c_{\text{eff}} \cdot c_\ell - d_{\text{eff}} \cdot d_\ell + g_{\text{eff}} \cdot g_\ell - \alpha(t_\ell + \kappa_{t_\ell} \cdot \sigma_{t_\ell}^2) & : \lambda_\ell^s \\
& s^g_{\ell,1:n} = s^g_{\ell,0:n-1} + f_\ell - \beta \cdot g_\ell & : \lambda_\ell^{s^g} \\
& t_{\ell h} = 0 \implies f_{\ell h} = 0, g_{\ell h} = 0 & : \eta_\ell^1, \eta_\ell^2 \\
& t_{\ell h} > 0 \implies c_{\ell h} = 0, d_{\ell h} = 0 & : \eta_\ell^3, \eta_\ell^4 \\
& 0 \le (s_\ell, c_\ell, d_\ell, s^g_\ell, f_\ell, g_\ell) \le (\bar{s}_\ell, \bar{c}_\ell, \bar{d}_\ell, \bar{s}^g_\ell, \bar{f}_\ell, \bar{g}_\ell) & : \nu_\ell, \gamma_\ell
\end{array} \quad \text{(LP-comp)}$$

If the solution to (LP-comp) *also* solves (LP-fleet), which includes charge cap constraint (3.1), then $q^\star$ is a *composite valuation.*

## 4.1 $(p + \theta^\star)$ as a Composite Valuation

In this chapter, we prove that $(p+\theta^\star)$ is a composite valuation. That is, we prove that if $q^\star = (p+\theta^\star)$, the solution to (LP-comp) also solves (LP-fleet) and will satisfy the charge cap constraint (3.1).

First, we explore the properties of $\theta^\star$ in Lemma 1, which will be used in the proof that $(p + \theta^\star)$ is a composite valuation. Define $(\hat{s}, \hat{c}, \hat{d}, \hat{(s^g)}, \hat{f}, \hat{g})$ and $(\hat{\lambda^s}, \hat{\lambda^{s^g}}, \hat{\eta^1}, \hat{\eta^2}, \hat{\eta^3}, \hat{\eta^4}, \hat{\nu}, \hat{\gamma})$ to be the optimal primal solution and the associated dual variables of (LP-comp), respectively.

**Lemma 1.** *Let $q^\star = (p + \theta^\star)$. Given the primal solution to* (LP-comp)*, $(\hat{s}, \hat{c}, \hat{d}, \hat{(s^g)}, \hat{f}, \hat{g})$, and the primal solution to* (LP-fleet)*, $(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)$, assume that for some $\ell = 1, \dots, k$ there is an element pair $(h, j)$ such that*

$$(\hat{c}_{\ell h} - \hat{d}_{\ell h}) < (c^\star_{\ell h} - d^\star_{\ell h}) \quad \textit{and} \quad (\hat{c}_{\ell j} - \hat{d}_{\ell j}) > (c^\star_{\ell j} - d^\star_{\ell j}). \tag{4.1}$$

*That is, assume the net charge done in hour $h$ in the solution to* (LP-fleet) *is* instead *in hour $j$ in the solution to* (LP-comp)*. Then $(p_h + \theta^\star_h) \leq (p_j + \theta^\star_j)$.*

*Proof.* Define

$$u^\star := u(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star) = \sum_{\ell=1}^{k} b_\ell \left(p^T(c^\star_\ell - d^\star_\ell) + (p^g)^T f^\star_\ell\right),$$

as the optimal objective function value of (LP-fleet). Recall the definition of $\theta^\star_h$ from Section 3.5.1,

$$\theta^\star_h = \frac{\Delta u(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)}{\Delta\left(\sum_{\ell=1}^{k} b_\ell(c^\star_{\ell h} - d^\star_{\ell h})\right)}. \tag{3.2}$$

Observe that when there is no degeneracy, by strict complementarity,

$$\begin{cases} \theta^\star_h > 0 & \iff \sum_{\ell=1}^{k} b_\ell(c^\star_{\ell h} - d^\star_{\ell h}) = (\hat{c}_{\text{cap}})_h \\ \theta^\star_h = 0 & \iff \sum_{\ell=1}^{k} b_\ell(c^\star_{\ell h} - d^\star_{\ell h}) < (\hat{c}_{\text{cap}})_h. \end{cases} \tag{4.2}$$

That is, if the net charge of the fleet in hour $h$ is on its upper bound, then $\theta^\star_h > 0$; otherwise, $\theta^\star_h = 0$.

We consider 3 scenarios (using that $\theta^\star \geq 0$) to prove Lemma 1 holds. We ignore the case that the solution to (LP-comp) is distinct from the solution to (LP-fleet), but would also be an optimal solution to (LP-fleet). That is, we ignore the case that $p_h = p_j$, and $(\hat{c}_{\hat{\ell}} - \hat{d}_{\hat{\ell}}) - (c^\star_{\hat{\ell}} - d^\star_{\hat{\ell}}) = \omega(e_h - e_j)$, for some $\omega \in \mathbb{R}$.

1. Scenario 1: $\theta^\star_h = \theta^\star_j = 0$

   By strict complementarity (4.2), the charge cap upper bound in (LP-fleet) is not attained in hour $h$ nor in hour $j$, and we know that the schedules $\hat{c}_{\hat{\ell}}$ and $\hat{d}_{\hat{\ell}}$ are feasible for all constraints in (LP-fleet) except the charge cap constraint. If we assume the conditions of Lemma 1 do not hold, i.e., that $(p_j + \theta_j^\star) = p_j < p_h = (p_h + \theta_h^\star)$, then there exists a vector $z := e_j - e_h$ and a scalar $\epsilon > 0$ for which

$$\left(\sum_{\ell \neq \hat{\ell}} b_\ell (c_\ell^\star - d_\ell^\star)\right) + b_{\hat{\ell}} \left(c_{\hat{\ell}}^\star - d_{\hat{\ell}}^\star + \epsilon z\right) \leq c_{\text{cap}},$$

and

$$\left(\sum_{\ell \neq \hat{\ell}} b_\ell (p^T(c_\ell - d_\ell) + (p^g)^T f_\ell)\right) + b_{\hat{\ell}} \left(p^T(c_{\hat{\ell}} - d_{\hat{\ell}} + \epsilon z) + (p^g)^T f_{\hat{\ell}}\right)$$
$$< \sum_{\ell=1}^{k} b_\ell \left(p^T(c_\ell - d_\ell) + (p^g)^T f_\ell\right).$$

That is, there exists a feasible set of charging and discharging schedules distinct from $c_\ell^\star, d_\ell^\star$ that results in a lower objective function value. This contradicts the optimality of $c_\ell^\star$ and $d_\ell^\star$. Thus, $p_h \leq p_j$ and Lemma 1 holds.

2. Scenario 2: $\theta_h^\star = 0, \theta_j^\star > 0$

   If we assume the conditions of Lemma 1 do not hold, i.e., that $p_j + \theta_j^\star < p_h$. Then by (4.2),

$$p_j < p_h \quad \text{and} \quad \sum_{\ell=1}^{k} b_\ell (c_{\ell h}^\star - d_{\ell h}^\star) < (c_{\text{cap}})_h \quad \text{and} \quad \sum_{\ell=1}^{k} b_\ell (c_{\ell j}^\star - d_{\ell j}^\star) = (c_{\text{cap}})_j.$$

   And by the definition of $\theta_j^\star$ from (3.2),

$$\theta_j^\star = \frac{\Delta u^\star}{\Delta\left(\sum_{\ell=1}^{k} b_\ell (c_{\ell j}^\star - d_{\ell j}^\star)\right)} = \frac{b_{\hat{\ell}}(p_h - p_j)(\Delta(c_{\hat{\ell} j} - d_{\hat{\ell} j}))}{b_{\hat{\ell}}(\Delta(c_{\hat{\ell} j} - d_{\hat{\ell} j}))} = p_h - p_j, \tag{4.3}$$

   where $\hat{\ell} = (\text{argmax}_{\ell \in \mathcal{L}}\, b_\ell)$, for $\mathcal{L}$ the set of driving patterns for which it is feasible to switch charge from hour $j$. We use $(\Delta(c_{\hat{\ell} j} - d_{\hat{\ell} j}))$ to refer to the change in the net charge for driving profile $\hat{\ell}$ in hour $j$ when $(\hat{c}_{\text{cap}})_j$ is relaxed by one unit.

   In other words, if it is cheaper for a vehicle to charge in hour $j$, but due to the charge cap it is *not* feasible to charge more in hour $j$ and it *is* feasible to charge more in hour $h$, then the value of $\theta_j^\star$ will be the maximum possible price difference between charging in hour $h$ and charging in hour $j$.

   However, (4.3) implies $\theta^\star = p_h - p_j$, or $p_h = p_j + \theta_j^\star$, which contradicts the inequality $p_j + \theta_j^\star < p_h$. Thus, $p_h \leq p_j + \theta_j^\star$ and Lemma 1 holds.

3. Scenario 3: $\theta_h^\star > 0$

   By (4.2), the charge cap upper bound is attained in hour $h$. Since we are considering the case that the net charge in hour $h$ is switched to hour $j$, and $\theta_h^\star > 0$ implies a positive improvement in the objective function when the charge cap in hour $h$ is relaxed, then by (3.2),

$$\theta_h^\star = \frac{\Delta u^\star}{\Delta\left(\sum_{\ell=1}^{k} b_\ell (c_{\ell h}^\star - d_{\ell h}^\star)\right)} = \frac{b_{\hat{\ell}}(p_j - p_h)(\Delta(c_{\hat{\ell}h} - d_{\hat{\ell}h}))}{b_{\hat{\ell}}(\Delta(c_{\hat{\ell}h} - d_{\hat{\ell}h}))} = p_j - p_h. \tag{4.4}$$

   This implies $\theta^\star = p_j - p_h$, or

$$p_h + \theta_h^\star = p_h + (p_j - p_h) = p_j \leq p_j + \theta_j^\star.$$

Thus, $(p_h + \theta_h^\star) \leq (p_j + \theta_j^\star)$ and Lemma 1 holds. □

We now use Lemma 1 to prove Theorem 1.

**Theorem 1.** *The set of values $(p + \theta^\star) \in \mathbb{R}^n$ is a composite valuation.*

*Proof.* We prove that

$$(\hat{s}, \hat{c}, \hat{d}, (\hat{s^g}), \hat{f}, \hat{g}) \equiv (s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star). \tag{4.5}$$

That is, given the same input parameters, the solution to (LP-fleet) and the solution to (LP-comp) are the same. The main implication of this is significant: when given the set of values $(p + \theta^\star)$, the charge cap constraint (3.1) is not necessary to ensure the market obligation is met.

The solution to (LP-comp) will also solve (LP-fleet) if $(\hat{s}, \hat{c}, \hat{d}, (\hat{s^g}), \hat{f}, \hat{g})$ satisfies the feasibility and optimality conditions of (LP-fleet). That is, we must show that $(\hat{s}, \hat{c}, \hat{d}, (\hat{s^g}), \hat{f}, \hat{g})$ satisfies each of the following feasibility and optimality conditions of (LP-fleet).

The optimality conditions of (LP-fleet) are

$$\begin{aligned}
(\lambda^s)_{1:n,\ell}^\star - (\lambda^s)_{0:n-1,\ell}^\star + (\gamma^s)_\ell^\star - (\nu^s)_\ell^\star &= 0\\
b_\ell p + c_{\text{eff}} \cdot (\lambda^s)_{1:n,\ell}^\star + I_d^\ell (\eta^1)_\ell^\star + b_\ell \theta^\star + (\gamma^c)_\ell^\star - (\nu^c)_\ell^\star &= 0\\
-b_\ell p - d_{\text{eff}} \cdot (\lambda^s)_{1:n,\ell}^\star + I_d^\ell (\eta^2)_\ell^\star - b_\ell \theta^\star + (\gamma^d)_\ell^\star - (\nu^d)_\ell^\star &= 0\\
(\lambda^{s^g})_{1:n}^\star - (\lambda^{s^g})_{0:n-1}^\star + (\gamma^{s^g})^\star - (\nu^{s^g})^\star &= 0\\
b_\ell p_g + (\lambda^{s^g})_{1:n,\ell}^\star + I_{nd}^\ell (\eta^2)_\ell^\star + (\gamma^f)_\ell^\star - (\nu^f)_\ell^\star &= 0\\
g_{\text{eff}} \cdot (\lambda^s)_{1:n,\ell}^\star - \beta \cdot (\lambda^{s^g})_{1:n,\ell}^\star + I_{nd}^\ell (\eta^3)_\ell^\star + (\gamma^g)_\ell^\star - (\nu^g)_\ell^\star &= 0\\
\nu_\ell^\star, \gamma_\ell^\star, \theta^\star &\geq 0
\end{aligned} \tag{OPT}$$

and the feasibility conditions of (LP-fleet) are

$$\begin{aligned}
&s^{\star}_{\ell,1:n} = s^{\star}_{\ell,0:n-1} + c_{\text{eff}} \cdot c^{\star}_{\ell} + d_{\text{eff}} \cdot d^{\star}_{\ell} + g_{\text{eff}} \cdot g^{\star}_{\ell} - \alpha \cdot (t_{\ell} + \kappa_{t_{\ell}} \cdot \sigma^2_{t_{\ell}}) \\
&{s^g}^{\star}_{\ell,1:n} = {s^g}^{\star}_{\ell,0:n-1} + f^{\star}_{\ell} - \beta \cdot g^{\star}_{\ell} \\
&t_{\ell h} = 0 \implies f^{\star}_{\ell h} = 0, g^{\star}_{\ell h} = 0 \\
&t_{\ell h} > 0 \implies c^{\star}_{\ell h} = 0, d^{\star}_{\ell h} = 0 \\
&\textstyle\sum_{\ell=1}^{m} b_{\ell}(c^{\star}_{\ell} - d^{\star}_{\ell}) \leq \hat{c}_{\text{cap}} \\
&0 \leq (s^{\star}_{\ell}, c^{\star}_{\ell}, d^{\star}_{\ell}, {s^g}^{\star}_{\ell}, f^{\star}_{\ell}, g^{\star}_{\ell}) \leq (\bar{s}_{\ell}, \bar{d}_{\ell}, \bar{c}_{\ell}, \bar{s}^g_{\ell}, \bar{f}_{\ell}, \bar{g}_{\ell}).
\end{aligned} \tag{FEAS}$$

The optimality conditions of (LP-comp) are

$$\begin{aligned}
\widehat{\lambda^s}_{1:n,\ell} - \widehat{\lambda^s}_{0:n-1,\ell} + \widehat{\gamma^s}_{\ell} - \widehat{\nu^s}_{\ell} &= 0 \\
b_{\ell}(p + \theta^{\star}) + c_{\text{eff}} \cdot \widehat{\lambda^s}_{1:n,\ell} + I^{\ell}_{d}\widehat{\eta^1}_{\ell} + \widehat{\gamma^c}_{\ell} - \widehat{\nu^c}_{\ell} &= 0 \\
-b_{\ell}(p + \theta^{\star}) - d_{\text{eff}} \cdot \widehat{\lambda^s}_{1:n,\ell} + I^{\ell}_{d}\widehat{\eta^2}_{\ell} + \widehat{\gamma^d}_{\ell} - \widehat{\nu^d}_{\ell} &= 0 \\
\lambda^{\widehat{s^g}}{}_{1:n} - \lambda^{\widehat{s^g}}{}_{0:n-1} + \gamma^{\widehat{s^g}} - \nu^{\widehat{s^g}} &= 0 \\
b_{\ell}p_g + \lambda^{\widehat{s^g}}{}_{1:n,\ell} + I^{\ell}_{nd}\widehat{\eta^2}_{\ell} + \widehat{\gamma^f}_{\ell} - \widehat{\nu^f}_{\ell} &= 0 \\
g_{\text{eff}} \cdot \widehat{\lambda^s}_{1:n,\ell} - \beta \cdot \lambda^{\widehat{s^g}}{}_{1:n,\ell} + I^{\ell}_{nd}\widehat{\eta^3}_{\ell} + \widehat{\gamma^g}_{\ell} - \widehat{\nu^g}_{\ell} &= 0 \\
\hat{\nu}_{\ell}, \hat{\gamma}_{\ell} &\geq 0.
\end{aligned} \tag{EQ-OPT}$$

By definition, $\theta^{\star}$ in (EQ-OPT) is equivalent to $\theta^{\star}$ in (OPT) and the optimality conditions (OPT) and (EQ-OPT) are equivalent. That is, the dual variables of (LP-comp) satisfy the optimality conditions of (LP-fleet). The solution of (LP-comp) is optimal for (LP-fleet), *if* it is feasible, i.e., if it satisfies (FEAS).

Because (LP-comp) has exactly the same constraints as (LP-fleet), except for the market obligation (3.1), $(\hat{s}, \hat{c}, \hat{d}, (\widehat{s^g}), \hat{f}, \hat{g})$ satisfies each of the feasibility conditions in (FEAS) except

$$\sum_{\ell=1}^{k} b_{\ell}(\hat{c}_{\ell} - \hat{d}_{\ell}) \leq \hat{c}_{\text{cap}}. \tag{4.6}$$

Therefore, it remains to show that (4.6) holds. We prove this by contradiction. Assume that (4.6) does not hold, that is, assume

$$\sum_{\ell=1}^{k} b_{\ell}(\hat{c}_{\ell} - \hat{d}_{\ell}) \nleq c_{\text{cap}}.$$

Then there exists an hour, say hour $h$, such that

$$\sum_{\ell=1}^{k} b_{\ell}(\hat{c}_{\ell h} - \hat{d}_{\ell h}) > (c_{\text{cap}})_h.$$

Because $\sum_{\ell=1}^{k} b_\ell(c_\ell^\star - d_\ell^\star) \leq c_{\text{cap}}$, there must be an $\hat{\ell} \in \{1, 2, \ldots, k\}$ such that $(\hat{c}_{\hat{\ell}h} - \hat{d}_{\hat{\ell}h}) > (c_{\hat{\ell}h}^\star - d_{\hat{\ell}h}^\star)$. Moreover, because both solutions to (LP-comp) and (LP-fleet) satisfy

$$s_{\ell,1:n}^\star = s_{\ell,0:n-1}^\star + c_{\text{eff}} \cdot c_\ell^\star - d_{\text{eff}} \cdot d_\ell^\star + g_{\text{eff}} \cdot g_\ell^\star - \alpha(t_\ell + \kappa_{t_\ell} \cdot \sigma_{t_\ell}^2),$$

then

$$e^T(c_\ell^\star - d_\ell^\star) = e^T(\hat{c}_\ell - \hat{d}_\ell). \tag{4.7}$$

That is, the total net charge done for driving profile $\ell$ in (LP-fleet) is equal to the total net charge done for the driving profile $\ell$ in (LP-comp). Intuitively, (4.7) follows from the fact that in both formulations, each base profile will have exactly enough energy in the battery to meet the augmented transport load. This also implies that the fueling variables in both problems will be equal, i.e., $\hat{f}_\ell = f_\ell^\star$ for all $\ell = 1, \ldots, k$.

Define $z \neq 0$ to be the vector difference between $(c_\ell^\star - d_\ell^\star)$ and $(\hat{c}_\ell - \hat{d}_\ell)$,

$$z := (\hat{c}_\ell - \hat{d}_\ell) - (c_\ell^\star - d_\ell^\star).$$

By (4.7), $e^T z = 0$; that is, $z$ must have both positive and negative elements, and the sum of its elements is zero. Because we assumed the solution to (LP-comp) is not equal to the solution of (LP-fleet), i.e., $(\hat{c}_\ell - \hat{d}_\ell) \neq (c_\ell^\star - d_\ell^\star)$, then

$$(p + \theta^\star)^T(\hat{c}_\ell - \hat{d}_\ell) = (p + \theta^\star)^T(c_\ell^\star - d_\ell^\star + z) < (p + \theta^\star)^T(c_\ell^\star - d_\ell^\star). \tag{4.8}$$

In other words, $(\hat{s}, \hat{c}, \hat{d}, \widehat{(s^g)}, \hat{f}, \hat{g})$ obtains a lower objective function value for (LP-comp) than $(s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star)$, which implies $(p + \theta^\star)^T z < 0$.

Without loss of generality, assume $z_j > 0$ for some $j$. Then there exists a non-empty set

$$H = \left\{ h \;\middle|\; z_h < 0, \left( \sum_{h \in H} \alpha_h z_h \right) + z_j = 0 \right\}.$$

More generally, by (4.7), for each cluster $\hat{\ell}$ such that $(\hat{c}_{\hat{\ell}j} - \hat{d}_{\hat{\ell}j}) > (c_{\hat{\ell}j}^\star - d_{\hat{\ell}j}^\star)$, there is a set of corresponding elements $h \in H$ such that $(\hat{c}_{\hat{\ell}h} - \hat{d}_{\hat{\ell}h}) < (c_{\hat{\ell}h}^\star - d_{\hat{\ell}h}^\star)$, and the net charge done in hours $h \in H$ in the solution to (LP-fleet) is transferred to hour $j$ in the solution to (LP-comp). Due to the optimality of $(\hat{c}_{\hat{\ell}j} - \hat{d}_{\hat{\ell}j})$ in (LP-comp), this implies that for all $h \in H$,

$$(p_j + \theta_j^\star) < (p_h + \theta_h^\star). \tag{4.9}$$

That is, the adjusted-price in hour $j$ must be less than the adjusted-price in hour $h$, since otherwise (4.8) does not hold. However, by Lemma 1,

$$(p_h + \theta_h^\star) \leq (p_j + \theta_j^\star). \tag{4.10}$$

This contradicts the assumption that there exists a vector $z \neq 0$ such that (4.8) holds for $e^T z = 0$.

Therefore, the solution to (LP-comp) will satisfy (FEAS) and (EQ-OPT), and hence

$$(\hat{s}, \hat{c}, \hat{d}, \widehat{(s^g)}, \hat{f}, \hat{g}) = (s^\star, c^\star, d^\star, (s^g)^\star, f^\star, g^\star).$$

The set of values $q^\star = (p + \theta^\star) \in \mathbb{R}^n$ is a composite valuation. □

The implication of Theorem 1 is the following: Given a fleet of $m$ vehicles with base driving profiles $t_\ell$, for $\ell = 1, \ldots, k$, if each PEV were to charge in the hours when $(p + \theta^\star)$ is the lowest and discharge in the hours when $(p + \theta^\star)$ is the highest, then

1. the cumulative fleet charging will satisfy the market obligation,
2. the lowest feasible cost will be attained, and
3. each base profile will have enough energy to meet its transport load.

## 4.2 Composite Valuation Examples

The composite valuation takes into account the benefits to consumers and utilities for each hour. Figures 4.1 and 4.2 illustrate two examples of the composite valuation for sample fleets, for a given 24-hour period.

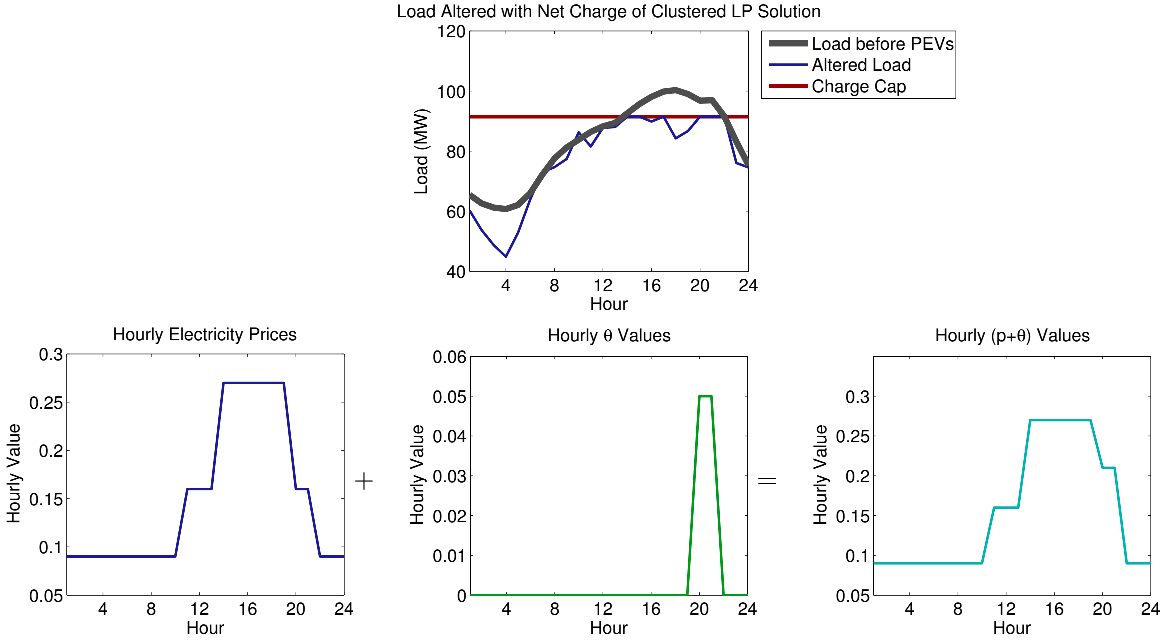


Figure 4.1: Example 1 of Composite Valuation for a Sample Fleet

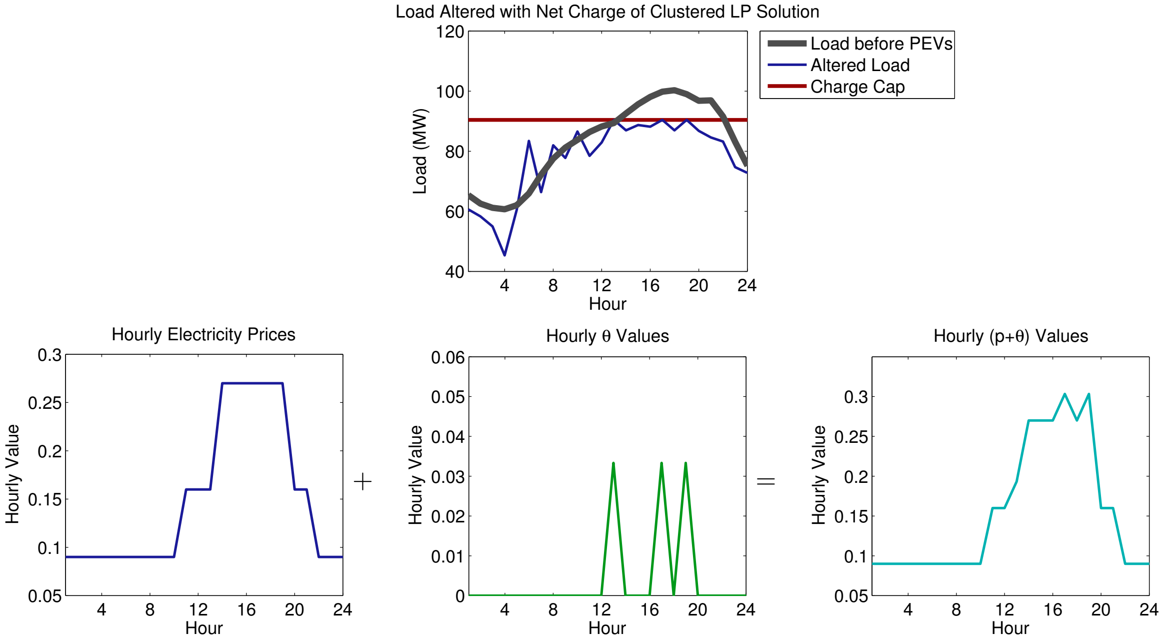


Figure 4.2: Example 2 of Composite Valuation for a Sample Fleet

Each example shows the construction of the composite valuation along with the base electricity load, the charge cap determined by (LP-fleet), and the altered load, which is the result of adding the net charge of the solution to (LP-fleet) to the base electricity load.

The vector $\theta$ represents the hourly value of fleet charging given the charge cap constraint and the fleet energy demand. Hence, as is shown in the figures, fleets with different charging needs will result in different composite valuations.

# 5 A Robust Dynamic Algorithm

We use the composite valuation from Chapter 4 to construct an algorithm that determines energy exchange schedules for individual vehicles that benefit both consumers and electric utilities.

The algorithm described in this chapter can be implemented by a device attached to each individual vehicle, and the aggregator can communicate with this device, as described in Chapter 2. The aggregator clusters driving patterns from a previous similar period, then solves (LP-fleet) to obtain the dual variable $\theta^\star$. The composite valuation $(p + \theta^\star)$ can be distributed to each vehicle's device, which can then determine the best energy exchange schedule without a back-and-forth communication with the aggregator or the remainder of the fleet.

The algorithm we construct is:

- *Dynamic:* an energy exchange schedule is determined at the moment the vehicle plugs-in
- *Distributed:* each vehicle will have enough information to determine its energy exchange schedule without back-and-forth communication between the aggregator or the remainder of the fleet
- *Robust:* each vehicle battery will have enough energy to meet the transport load, given the uncertainty in driving behavior.

Because the aggregator shares the composite valuation with each vehicle in the fleet, the algorithm is dynamic and distributed; when a vehicle plugs-in, an energy exchange schedule can be determined using this set of values without consulting the aggregator or the remainder of the fleet. In Section 5.1, we consider how to make the algorithm robust.

## *5.1 Robustness*

To make this algorithm robust, we consider the amount of energy each vehicle battery should have before driving. That is, we consider the question: if a vehicle expects to drive $x$ miles in hour $h$, what should the battery state of charge be before hour $h$? This question can be viewed in terms of inventory management.

**Definition 10.** Inventory Management *deals with the quantity and types of available goods kept in stock by a business, or a similar entity, in order to meet the consumer demand.*

We apply inventory management to determine the quantity of energy to be stored in the battery.

Each component of a business in the inventory management problem can be mapped to a component of a PEV. The *consumer demand* corresponds to the amount of energy the vehicle needs in order to drive the desired amount, which is an uncertain variable. The vehicle battery corresponds to the *storage location* for stocked goods, and the *amount of inventory* represents the battery state of charge.

The demand in our problem is uncertain because it is unknown exactly how much each vehicle will be driven. That is, we assume the PEV driver provides an expected driving schedule, but the realized driving schedule may be different.

In inventory management, it is standard to optimize the state of charge (or battery inventory) based on cost. The optimal inventory will minimize the following cost function:

$$\text{inventory cost} = \text{ordering costs} + \text{holding costs} + \text{shortage costs.}$$

The *ordering costs* are the costs of ordering the good to keep in inventory. In terms of our vehicle battery, this is the cost of obtaining electricity to store in the battery, i.e., the cost of charging. This cost is minimized when the vehicle battery is charged in the hours with the lowest cost, and discharged in the hours with the highest cost.

The *holding costs* are the result of keeping inventory in stock when there is not a current demand for it. This corresponds to the cost of storing energy in the vehicle battery that is not used in the imminent driving loads. Assuming the energy will eventually be used for driving, this cost is extremely small. It is non-zero because the state of charge will decrease naturally over time, without the vehicle being driven or battery discharging. For example, if a Lithium-Ion battery is completely idle at a temperature of 77° F (25° C), the state of charge will decrease 20% per month from a full charge, or 4% per month if the state of charge is between 40–60% [68].

The *shortage costs* are incurred when there is not enough inventory to meet the demand. In terms of our vehicle battery, this is the cost of insufficient energy in the battery to complete the driving load. The cost of a driver running out of charge before reaching his or her destination is extremely high for BEVs. For example, the car may need to be towed, or the driver may need to find alternate transportation to his or her destination. For PHEVs, the shortage cost is the cost of relying on gasoline, which is relatively high compared to electricity costs and holding costs.

Consequently, the inventory cost for a vehicle $j$ is:

$$\begin{aligned}\text{inventory cost} &= \quad \text{ordering costs} \quad + \quad \text{holding costs} \quad + \quad \text{shortage costs}\\ &= \; p^T(c_j - d_j) + p_g^T f_j \quad + \quad \epsilon \cdot \mathbf{1}^T(s_j - t_j)_+ \quad + \quad S \cdot \mathbf{1}^T(t_j - s_j)_+,\end{aligned}$$

where $(x)_+ = \{x_i\} = \{\max(x_i, 0)\}$ and $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones, for small $\epsilon > 0$ and large $S$.

Note that the ordering cost $(p^T(c_j - d_j) + p_g^T f_j)$ is proportional to the amount the vehicle charges (or potentially fuels). However, even if the vehicle charges (or fuels) enough to fill the battery (or tank) from empty to full capacity, it can still be safely assumed that

$$0 < \epsilon \ll p^T(c_j - d_j) + p_g^T f_j \ll S.$$

In order to minimize the overall inventory cost given an uncertain driving load, shortage costs should not be incurred. That is, there is a steep price of an energy shortage, and shortage costs should be avoided when possible. Because the holding costs are negligible, the expected inventory cost will be minimized by storing as much energy in the battery as possible, i.e., charging the vehicle battery to its full capacity in the hours with the lowest possible cost.

Therefore, our algorithm ensures each vehicle battery contains the maximum possible state of charge before the start of the transport load. This will also make our algorithm robust to an uncertain number of miles to be driven.

## *5.2 Rolling Horizon*

The preceding Section 5.1 establishes the *amount* that each vehicle should charge to account for uncertainty in the number of miles driven. In this section, we address the uncertainty in the *timing* of the transport load (i.e., the hours in which the vehicle is driven).

Let $h$ refer to the next hour in which a vehicle will be driving. It is reasonable to assume that as time approaches hour $h$, the accuracy of the predicted vehicle driving schedule will improve. For example, consider a driver who plans to drive at 8am and 5pm on the following day, say to and from work. Yet at 9am on the given day, the driver decides he will also be driving around lunchtime, at 12 noon. We want our algorithm to consider this change, and sufficiently charge the vehicle by noon instead of by 5pm.

To account for such changes in transport load, we implement our algorithm on a rolling horizon that re-runs the algorithm periodically. That is, at a given hour $h$, the time horizon will be hour $(n+h-1)$. The algorithm will be implemented at a series of hours: $h = 0, \tau, 2\tau, 3\tau, \ldots$, where $\tau > 0$ is a chosen increment parameter; our simulations use $\tau = 3$ hours.

At each hour $h = 0, \tau, 2\tau, 3\tau, \ldots$, the algorithm will have improved information to update the energy exchange schedules and account for changes in the expected driving schedule. The aggregator will solve (LP-fleet) at hour $h$ to create a new composite valuation from hours $h$ to $(n+h-1)$. Each composite valuation will be a vector of $n$ elements, and can be used to determine energy exchange schedules for the considered $n$-hour period.

## *5.3 Justification of Robustness in* (LP-fleet)

The formulation (LP-fleet) is also robust, because the constraints ensure that the state of charge for each cluster $\ell$ is sufficient to meet the needs of 95% of drivers in cluster $\ell$; this robust formulation has an advantage over a nonrobust formulation. Consider (LP-nonrobust), which does not account for uncertainty in the potential demand of the vehicles or in the base electricity load.

$$
\begin{array}{lll}
\underset{c_\ell, d_\ell, f_\ell, g_\ell, \forall \ell}{\text{minimize}} & \sum_{\ell=1}^{k} b_\ell \left( p^T (c_\ell - d_\ell) + (p^g)^T f_\ell \right) + \rho \cdot c_{\text{cap}} & \\
\text{subject to} & s_{\ell,1:n} = s_{\ell,0:n-1} + c_{\text{eff}} \cdot c_\ell - d_{\text{eff}} \cdot d_\ell + g_{\text{eff}} \cdot g_\ell - \alpha t_\ell & : \lambda_\ell^s \\
& s_{\ell,1:n}^g = s_{\ell,0:n-1}^g + f_\ell - \beta \cdot g_\ell & : \lambda_\ell^{s^g} \\
& t_{\ell h} = 0 \implies f_{\ell h} = 0, g_{\ell h} = 0 & : \eta_\ell^1, \eta_\ell^2 \\
& t_{\ell h} > 0 \implies c_{\ell h} = 0, d_{\ell h} = 0 & : \eta_\ell^3, \eta_\ell^4 \\
& \sum_{\ell=1}^{k} b_\ell (c_\ell - d_\ell) \leq c_{\text{cap}} \cdot 1 - L & : \theta \\
& 0 \leq (s_\ell, c_\ell, d_\ell, s_\ell^g, f_\ell, g_\ell) \leq (\bar{s}_\ell, \bar{c}_\ell, \bar{d}_\ell, \bar{s}_\ell^g, \bar{f}_\ell, \bar{g}_\ell) & : \nu_\ell, \gamma_\ell.
\end{array}
\qquad \text{(LP-nonrobust)}
$$

Let $\theta_{\text{rob}} \in \mathbb{R}^n$ and $\theta_{\text{nonrob}} \in \mathbb{R}^n$ represent the dual variables of (LP-fleet) and (LP-nonrobust) respectively, given the same input. Figure 5.1 shows an example of $\theta_{\text{rob}}$ and $\theta_{\text{nonrob}}$, and loads altered with the net charge amounts resulting from (LP-fleet) and (LP-nonrobust), given the same input. As can be seen in this figure, the net charge resulting from (LP-fleet) is more than the net charge resulting from (LP-nonrobust).

A simulation of 300 different clusters of fleets showed that 59% of the time, the peak demand value of the altered load from (LP-fleet) is less than the peak demand of the altered load from (LP-nonrobust). That is, on average the fleet of vehicles charges *more* with a robust formulation, but the maximum peak demand is lower. Thus, the robust formulation (LP-fleet) accounts for uncertainty in vehicle demands, while reducing the average impact on peak demand.

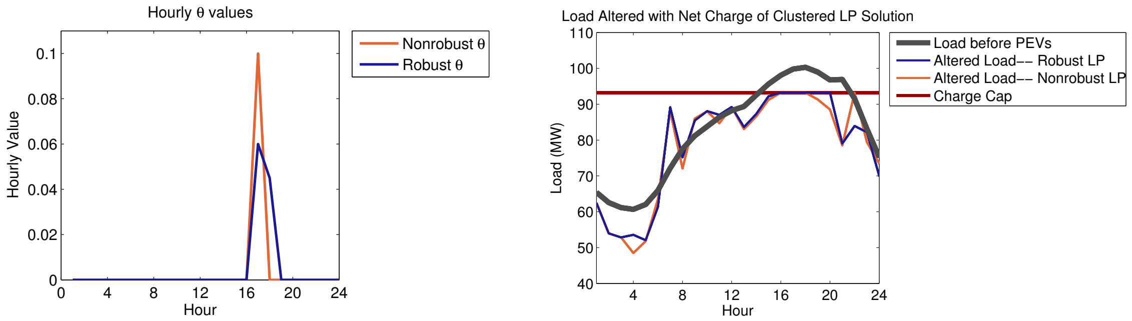


Figure 5.1: Example of $\theta$ and Altered Load for a Fleet using a Robust and Nonrobust Formulation

## 5.4 Algorithm Description

We now describe Algorithm 1, which uses the composite valuation to determine energy exchange schedules for PEVs. It includes the actions of both the aggregator and the devices on the PEVs. The algorithm is run on a rolling horizon and re-evaluates every $\tau$ hours. At each re-evaluation, the aggregator solves the robust formulation (LP-fleet) to find the marginal hourly price of electricity, $\theta^\star \in \mathbb{R}^n$, and constructs the composite valuation $q^\star = (p + \theta^\star) \in \mathbb{R}^n$. When a PEV connects to the

**Algorithm 1** Electricity Trading with a fleet of PEVs using a Composite Valuation

```
for $h = 0, \tau, 2\tau, 3\tau, \ldots$ do
  Cluster past driving patterns and solve the static linear program to find the composite valuation $q^\star := (p + \theta^\star) \in \mathbb{R}^n$, for hours $h$ to $h + n - 1$
  for each vehicle that connects to the grid do
    if the vehicle can charge for $h_c$ hours then
      Charge the battery in the $h_c$ hours when $q^\star$ is lowest
    end if
    if the number of hours spent idle is positive, $h_i > 0$ then
      Discharge the battery in the $(h_i/2)$ hours when $q^\star$ is highest
      Charge the battery in the $(h_i/2)$ hours when $q^\star$ is lowest
    end if
    if the vehicle has a fuel tank and can rely on fuel then
      Alert driver to fill the fuel tank
    end if
  end for
end for
```

grid, it uses the composite valuation to instantly determine an energy exchange schedule.

To determine the best energy exchange schedule, the vehicle battery is maximally charged in the hours when $q^\star$ is lowest. If there are hours in which the vehicle battery is spent idle (i.e., not charging, discharging or driving), then the battery charges and discharges equal amounts of energy during the idle time period; this charging is done in the hours with the lowest possible values of $q^\star$ and discharging is done in the hours with the highest possible values of $q^\star$. If the vehicle has a gasoline tank that is near empty, the gasoline tank is filled during the next driving period.

## *5.5 Implementation*

We implemented this algorithm in Matlab. The details of the implementation including data analysis and input parameters are discussed below in Sections 5.5.1 and 5.5.2. For consistency in our simulations, we use data from urban California. The results are in Section 5.7.

### 5.5.1 Data Analysis

We simulated our algorithm using real data on: driving behaviors, electricity loads, electricity and gasoline pricing, day-ahead electricity load prediction errors, and vehicle characteristics.

#### Driving Behavior

We obtained real driving patterns from the 2009 National Household Travel Survey [64], as discussed in Section 3.3.1. The data for each trip is extremely detailed, including over 80 fields of information about the trip and household. Each trip is also given a weighting so that the *weighted* version of the data is a representative sample.

Our simulations use the subset of trips from urban California, around 17,000 households. The data was cleaned and the relevant fields were used, including the following data on each trip: the start and end time, the number of miles traveled, and the starting and ending location. The day of the week on which the travel occurred was also recorded as well as the size of the household, the type of vehicle, and the miles per gallon of each vehicle.

These driving patterns were grouped into 37 clusters, as described in Section 3.3.1. We found the variance of the driving patterns in each cluster, $\sigma_{t_\ell}$, and used a binary search method to determine the scalar $\kappa_{t_\ell}$ so that 95% of the transport loads in the cluster are less than $\kappa_{t_\ell} \cdot \sigma_{t_\ell}$.

We assume in our simulations that at a given hour $h$, the exact driving schedules are known for hours $\{h, \ldots, h+\tau\}$, and an expected (but possibly inexact) driving schedule is known for the driving schedules between hours $\{h+\tau+1, \ldots, h+n-1\}$. As time progresses, we assume that PEV drivers update their vehicle devices according to the predicted driving load.

Specifically, let $\bar{z}$ be the actual number of miles driven in a given hour $\hat{h} \in \{h+\tau+1, \ldots, h+n-1\}$, and $z$ be the predicted number of miles to be driven in hour $\hat{h}$. Then we assume $z \sim \mathcal{N}(\bar{z}, 5)$. Similarly, if $\bar{y}$ is the actual starting hour of a given trip occurring between hours $\{h+\tau+1, \ldots, h+n-1\}$, and $y$ is the predicted starting hour of the trip, then we assume $y \sim \mathcal{N}(\bar{y}, 2)$.

### Electricity and Gasoline Pricing

The electricity prices used are the PG&E summer baseline time-of-use (TOU) rates [50], which take on three values throughout a 24-hour period, as shown in Figure 5.2. The gasoline price used is \$3.82/gallon, which was the mean price of regular gasoline in California for the week of August 25, 2011 [18].

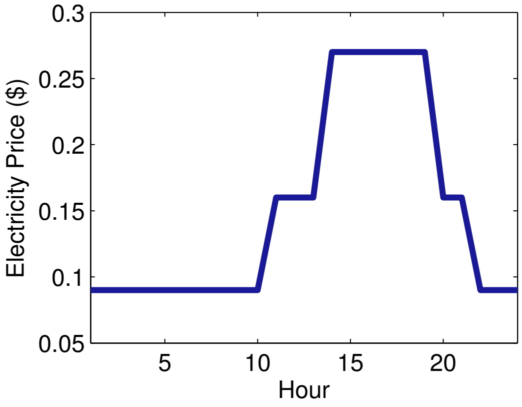


Figure 5.2: PG&E Summer Baseline Time-of-Use Rates

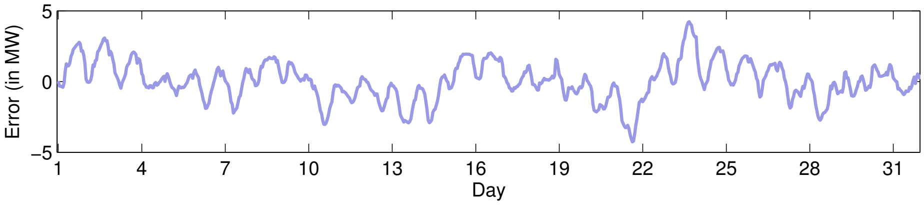


Figure 5.3: Day-Ahead Prediction Market Errors for the month of August 2011

### Electricity Load

Electricity loads were obtained from the California Independent System Operator (CAISO) in the PG&E transmission area. This data is available on the CAISO Open Access Same-time Information System (OASIS) [11]. The PG&E system loads for the month of August 2011 were used, and scaled using the assumption that PG&E serves 5.1 million households [51].

### Day-ahead Electricity Load Prediction Errors

A distribution of load prediction errors was obtained from the CAISO OASIS website, for the PG&E transmission access area. We obtained the predicted day-ahead demand and compared it to the realized demand. From this, we derived the variance $\sigma_L$ on the prediction load error, and used binary search to determine the corresponding value $\kappa_L$ so that the realized load will be less than $(L + \kappa_L \cdot \sigma_L^2)$ with 95% probability. Figure 5.3 shows the PG&E day-ahead prediction error in MW for August 2011.

### Vehicle Characteristics

Two types of PEVs were simulated: Battery Electric Vehicles and Plug-in Hybrid Electric Vehicles. The BEVs were modeled after the specifications of the Nissan Leaf [48], and the PHEVs were modeled after the Chevrolet Volt [47]. Table 5.1 shows the specifications of each vehicle type.

Table 5.1: Vehicle Model Specifications

| | BEV | PHEV |
|---|---|---|
| Battery Capacity | 20 kWh | 10.4 kWh |
| Fuel Tank Capacity | 0 | 8.6 gallons |
| Charging Rate | 6.6 kWh/h | |
| Discharging Rate | 6.6 kWh/h | |
| Fueling Rate | 0 | 3 gallons/min |
| Generation Rate | 0 | 55 kWh/h |

Table 5.2: Algorithm Input Parameters

| Parameter Name | Default Value |
|---|---|
| Number of Days | 5 days |
| Discrete Time Periods per Day | 24 hours |
| Time Horizon ($n$) | 120 hours |
| Re-evaluation Time Step ($\tau$) | 3 hours |
| Charging Efficiency ($c_{\text{eff}}$) | 81% |
| Discharging Efficiency ($d_{\text{eff}}$) | 81% |
| Generating Efficiency ($g_{\text{eff}}$) | 30% |
| kWh per Mile ($\alpha$) | 36/100 at 20 miles/hour<br>37/100 at 48 miles/hour |
| kWh per Gallon Gasoline ($1/\beta$) | 33.7 |
| Number of Clusters ($k$) | 37 |
| Optimization Solver | MOSEK |

### 5.5.2 Input Parameters

There are several different parameters input to our simulation that are designed for easy adjustment. Table 5.2 lists these parameters and the default values used for the results in Section 5.7.

Note that the last input parameter is the Optimization Solver, which can be specified to be one of two different solvers: MOSEK or SeDuMi (through CVX). MOSEK is an optimization software package that includes interior-point, simplex and mixed-integer optimizers. This solver is particularly efficient with sparse large-scale problems, such as (LP-fleet) [4]. SeDuMi is an optimization software package for Matlab that uses Self-Dual Minimization techniques, and also efficiently solves sparse large-scale problems [62]. We use SeDuMi via the modeling system CVX, which allows the optimization problem input to be written in a simply understood modeling language [32].

## *5.6 Comparison Algorithms*

In general, the *baseline* for comparison is the estimate of electricity usage in the absence of a DR mechanism [20]. Since we are simulating a fleet of vehicles, we compare the results of our algorithm to the outcome of vehicles charging at the maximum rate each time they plug-in for at least an hour (Algorithm 2, Standard Charging). We also compare the outcome of our algorithm to that of vehicles charging during the hours with the lowest possible cost (Algorithm 3, Lowest-Cost Charging). We briefly describe each algorithm used for comparison before summarizing the results.

### 5.6.1 Standard Charging

For a standard baseline, we simulate the outcome of vehicles charging at the maximum rate each time they plug-in, until either the battery is full or the connection is ended; we assume that a vehicle is plugged-in to the grid each time it is parked for at least an hour. This is Standard Charging, as described in Algorithm 2, and simulates PEV charging in the absence of DR intervention.

**Algorithm 2** Standard Charging

```
for each vehicle that connects to the grid do
  while the battery is not full and the vehicle is connected do
    Charge the battery at the maximum rate
  end while
end for
```

### 5.6.2 Lowest-Cost Charging

For a comparison to other potential DR mechanisms, we simulate the outcome if vehicles were to charge in the hours with the lowest possible electricity price, as in Algorithm 3. As described in [19], this is an Economic-based DR program, which is used in practice for DR resources in the Commercial and Industrial sector.

**Algorithm 3** Lowest-Cost Charging

```
for each vehicle that connects to the grid do
  if the vehicle can charge for $h_c$ hours then
    Charge the battery in the $h_c$ hours when $p$ is lowest
  end if
  if the vehicle has a fuel tank and can rely on fuel then
    Alert driver to fill the fuel tank
  end if
end for
```

## *5.7 Results*

The following results use the default input parameters (as listed in Table 5.2) unless otherwise specified. We compare two standard measurement and benefit features [29] from each charging scenario: cost to the consumer and increase in peak electricity demand. We also include the total amount of money spent on powering the fleet (with both electricity and fuel).

Figures 5.4 and 5.5 show the baseline electricity load, along with the augmented load when PEVs charge according to each of three Algorithms 1, 2 and 3. Ideally, the goal is for the altered load to decrease the peak demand and increase the off-peak demand, as in Figure 3.1.

Figure 5.5 shows the resulting augmented electricity loads, along with the charge cap as determined by (LP-fleet). As planned, the load augmented with net charge of the fleet using Algorithm 1 stays within the charge cap, where as the altered loads using Algorithms 2 and 3 do not.

Table 5.3: Results of Fleet in Figure 5.4

| | Standard Charging | Low-Cost Charging | Composite Valuation |
|---|---|---|---|
| Increase in Peak (%) | 5.1 | 1.4 | −0.15 |
| Total Fleet Cost ($) | 97,678 | 83,695 | 65,349 |
| Mean Cost Per Mile (¢) | 6.8 | 4.4 | .54 |

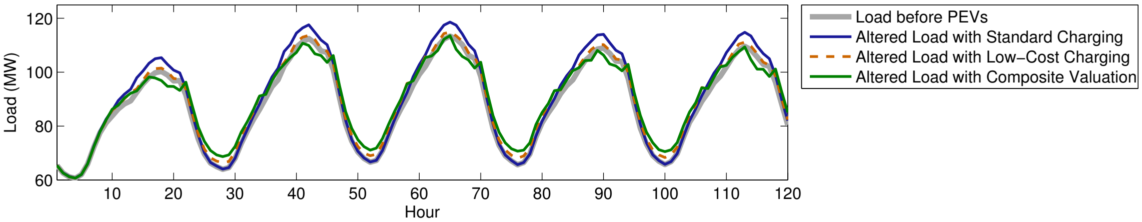


Figure 5.4: Example with a Fleet of 10,000 Vehicles for 5 days

Table 5.4: Results of Fleet in Figure 5.5

| | Standard Charging | Low-Cost Charging | Composite Valuation |
|---|---|---|---|
| Increase in Peak (%) | 4.3 | 1.4 | −0.16 |
| Mean Cost Per Mile (¢) | 6.7 | 4.5 | 0.32 |
| Total Fleet Cost ($) | 101,310 | 88,596 | 70,371 |

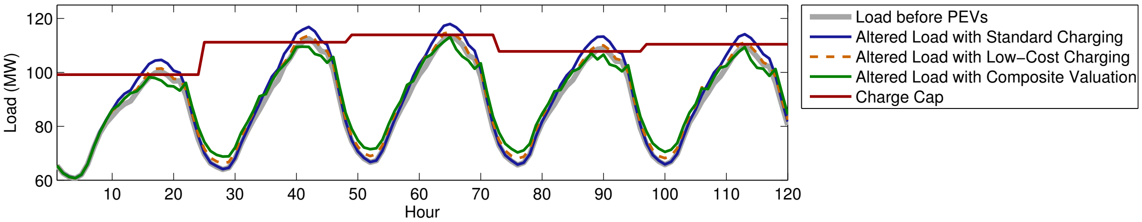


Figure 5.5: Example with a Fleet of 10,000 Vehicles for 5 days

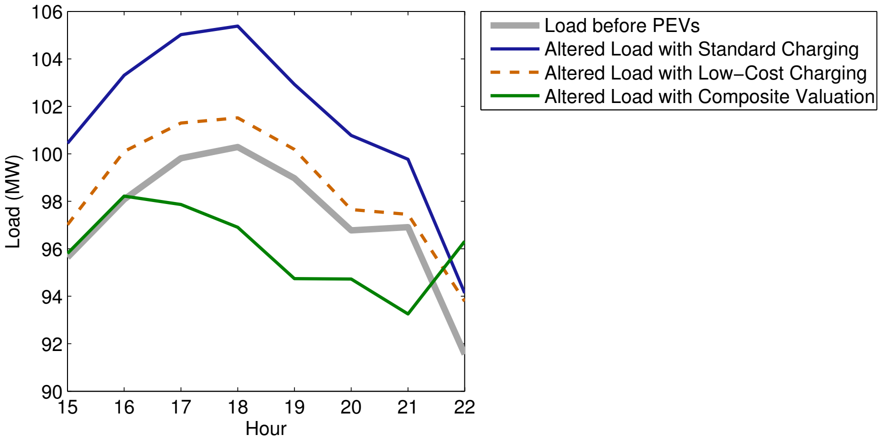


Figure 5.6: A Close-up of the Peak Demand in Figure 5.4

Figure 5.6 shows the peak demand during the first day of the example above. Specifically, it shows a close-up of hours 15 to 22 in Figure 5.4. This plot shows the drastic differences in the impact each of algorithms has on the peak electricity demand. Algorithm 1 clearly decreases the peak demand value, whereas the other algorithms may result in a need to build new generators.

Table 5.5 presents the mean statistics over all algorithms, for 200 simulated fleets of 10,000 vehicles over a 5-day period. These results, along with Figures 5.4, 5.5, and 5.6, show that our algorithm actually *decreases* the peak demand value for each day considered. Both standard charging and low-cost charging will result in an increase in peak demand, by 1.1% and 4.7%, respectively, whereas our algorithm results in a decrease in peak demand.

The results in Table 5.5 state that the cost per mile of a PEV can range from 0.67¢ to 6.5¢. For comparison, using the same pricing, a Toyota Prius that gets 50 miles per gallon would cost about 7.8¢ per mile. Similarly, a Volkswagen Golf with 32 miles per gallon would cost about 12.1¢ per mile. Thus, even when PEVs charge without a DR mechanism, it costs less to power a PEV than a standard internal combustion engine vehicle.

Table 5.5: Mean Values taken over 200 Simulated Fleets of 10,000 Vehicles for 2 days

| Mean Values | | | |
|---|---|---|---|
| | Standard Charging | Low-Cost Charging | Composite Valuation |
| Increase in Peak (%) | 4.7 | 1.1 | −1.1 |
| Mean Cost Per Mile (¢) | 6.5 | 4.1 | 0.67 |
| Total Fleet Cost ($) | 91,987 | 76,653 | 61,580 |

However, the cost per mile for a PEV when using our composite valuation algorithm is about one-tenth the cost as implementing standard charging, and less than one-sixth the cost of lowest-cost charging. Given that the average person drives around 13,500 miles per year [49], the yearly price difference between standard charging and using our composite valuation is almost $800 per vehicle (on average); the average yearly price difference between lowest charging and using our composite valuation would be more than $450 per vehicle. This provides motivation for consumers to participate in a DR program that uses a similar method to Algorithm 1.

Table 5.5 also shows that an implementation of Algorithm 1 could *decrease* the peak load by over 1%, whereas both Algorithms 2 and 3 result in an increase in peak electricity demand. This is motivation for utilities to implement a DR mechanism similar to our composite valuation algorithm.

Therefore, we have created an algorithm to assign energy exchange schedules to individual PEVs in a fleet that benefits both electricity consumers and utilities. The resulting altered load meets a market obligation that is determined by the aggregator as the optimal load curtailment. Consumers will incur the lowest possible cost, and their vehicle batteries will have the maximum state of charge before their expected driving loads. Our mechanism, Algorithm 1, is dynamic, distributed and robust and was tested on realistic data.

Part II

# Network of Sensor-equipped Vehicles

# 6 Background

Sensor-equipped vehicles will use developing sensor technology to detect the distance from themselves to nearby objects. In the next few years, SVs could improve standards such as safety and traffic flow; in the next few decades, they could be implemented in potential autonomous vehicles [61].

SVs may be equipped with GPS systems, which can determine the location of vehicles on the road with an accuracy of 7.8 meters at a 95% confidence level [31]. However, we explore the possibility of vehicles using sensors to detect the distance from the vehicle to its surroundings more accurately, i.e., with an accuracy of ±1cm. These sensors could also determine the relative *locations* of nearby vehicles, if a sufficient set of pair-wise distances between vehicles is known.

The aim of Part II in this thesis is to establish conditions on sensor density and structure in a given area to ensure sensors can accurately determine their relative location. We use a semidefinite programming relaxation of the problem formulation, and explore two properties of the sensors. In Chapter 7, we find a lower bound on the radio range of randomly distributed sensors to ensure the SDP provides the correct solution with high probability. In Chapter 8, we establish conditions on the structure, or framework, of $n$ sensors in order for the corresponding stress matrix to have rank $n$, and hence for there to be exactly one realizable localization of the sensors in all dimensions.

## 6.1 Problem Formulation

We consider a network of $n$ sensors whose locations are unknown, and $m$ anchors whose locations are known. The sensors can lie in any dimension $d \geq 1$, but we focus on the practical cases that $d = 2$ or $d = 3$.

The sensor network can be represented by a graph $G(V, E)$, where both the vertex set and the edge set consist of two distinct subsets. The vertices in $V = V_x \cup V_a$ are partitioned into the set $V_x = \{x_1, \ldots, x_n\}$ of $n$ sensors and the set $V_a = \{a_1, \ldots, a_m\}$ of $m$ anchors, for $m \geq d+1$. The edges in $E = E_a \cup E_x$ are partitioned into the set $E_x = \{(i,j) : i, j \in V_x\}$ of edges between sensors, and $E_a = \{(k,j) : k \in V_a, j \in V_x\}$ of edges between an anchor and a sensor. For each $(i,j) \in E_x$ (or $(k,j) \in E_a$) the Euclidean distance between sensor $i$ and sensor $j$ (respectively anchor $k$ and sensor $j$) is known as $d_{ij}$ (respectively $\bar{d}_{kj}$). The problem of finding the locations of the sensors can be formulated as finding points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ that satisfy the set of quadratic equations:

$$\begin{aligned} \| x_i - x_j \|^2 &= d_{ij}^2, \ \forall \ (i,j) \in E_x \\ \| a_k - x_j \|^2 &= \bar{d}_{kj}^2, \ \forall \ (k,j) \in E_a. \end{aligned} \tag{SNL-norm}$$

The question of whether there exists a realization of $x_j$s that solves (SNL-norm), or if such a realization is unique, has been extensively studied in the graph rigidity community from a theoretical perspective (see [14,35] and references therein). However, the question of whether there is an efficient computational algorithm to numerically answer some of these questions has only recently received attention, following the formulation of its relaxation as a semidefinite program [2, 7, 39, 60].

### 6.1.1 Semidefinite Programming Relaxation

The relaxation (SNL-SDP) is an effective computational method to find a set of points $\{x_1, x_2, \ldots, x_n\}$ that solve the system (SNL-norm) [7, 60]. We use the Matlab notation $Z_{(1:d,1:d)}$ to represent the upper-left $d$-dimensional principle submatrix of the decision variable $Z$, and the matrix dot-product to refer to the sum of element-wise products: $A \bullet B = \sum_{ij} A_{ij} B_{ij}$.

Let $e_i \in \mathbb{R}^n$ be the $i$th column of the identity matrix in $\mathbb{R}^{n\times n}$, and $\mathbf{0} \in \mathbb{R}^d$ be the vector of all zeros. Define the symmetric matrices $A_{ij} := (\mathbf{0}; e_i - e_j)(\mathbf{0}; e_i - e_j)^T$ and $\bar{A}_{kj} := (a_k; -e_j)(a_k; -e_j)^T$. We consider the following SDP relaxation of (SNL-norm):

$$\begin{array}{lll} \underset{Z}{\text{maximize}} & 0 & \\ \text{subject to} & A_{ij} \bullet Z = d_{ij}^2, & \forall (i,j) \in E_x \\ & \bar{A}_{kj} \bullet Z = \bar{d}_{kj}^2, & \forall (k,j) \in E_a \\ & Z \succeq 0,\ Z_{(1:d,1:d)} = I_d & \end{array} \tag{SNL-SDP}$$

Problem (SNL-SDP) is a convex semidefinite program and can be approximately solved in polynomial time by interior-point algorithms. The solution matrix to (SNL-SDP), $Z \in \mathbb{R}^{(d+n)\times(d+n)}$, can be decomposed into submatrices,

$$Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix},$$

such that the constraint $Z \succeq 0$ implies $Y \succeq X^T X$. If $Y = X^T X$ in the solution to (SNL-SDP), then the columns of the submatrix $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$ satisfy all quadratic equations in (SNL-norm).

## *6.2 Conditions for Unique Localizability*

In Chapters 7 and 8, we determine conditions that ensure the solution to (SNL-SDP) will provide a set of sensor locations that satisfy (SNL-norm). So et al. proved in [60] that the SDP relaxation will find the exact locations of the sensors if the sensor network is *uniquely localizable.*

**Definition 11.** *A sensor network is* uniquely localizable *if there is a unique $X \in \mathbb{R}^{d\times n}$ with columns that satisfy* (SNL-norm)*, and there is no $\bar{X} \in \mathbb{R}^{h\times n}$, for $h > d$, with columns that satisfy* (SNL-norm) *and $\bar{X} \neq (X; \mathbf{0})$. In other words, there is no non-trivial extension of $X \in \mathbb{R}^{d\times n}$ into higher dimension $h > d$ that also satisfies* (SNL-norm) *[60].*

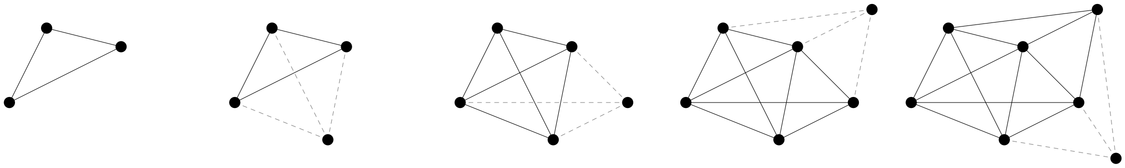


Figure 6.1: Construction of $(d+1)$-lateration graph in $d = 2$

The notion of unique localizability is stronger than the notion of global rigidity: a network can be globally rigid and also have a non-trivial extension of $X \in \mathbb{R}^{d\times n}$ into a higher dimension space that satisfies (SNL-norm) [1, 60].

In this thesis, we consider sensor networks that induce a certain class of graphs. Specifically, we consider networks that induce $(d+1)$*-lateration graphs*.

**Definition 12.** *For $d, n \geq 1$, the graph $G(V, E)$ is a* $(d{+}1)$-lateration graph *if there exists a permutation of the vertices, $\{\pi(1), \pi(2), \ldots, \pi(n)\}$, such that the edges of the sub-graph on $\{\pi(1), \ldots, \pi(d{+}1)\}$ form a complete graph, and each successive vertex $\pi(j)$ for $j \geq d+2$ is connected to $d+1$ vertices in the set $\{\pi(1), \ldots, \pi(j-1)\}$. The permutation $\pi$ of the vertices is called a* $(d+1)$-lateration ordering*. The graph $G(V, E)$ contains a* spanning $(d+1)$-lateration graph *if there is a subgraph $H(\hat{V}, \hat{E}) \subseteq G(V, E)$ such that $\hat{V} = V$, $\hat{E} \subseteq E$ and $H$ is a $(d+1)$-lateration graph.*

Figure 6.1 illustrates the construction of a sample $(d+1)$-lateration graph in dimension 2.

If the graph induced from a sensor network contains a *spanning* $(d+1)$*-lateration graph* and the sensor points are in general position, then it is uniquely localizable and (SNL-SDP) is a computationally tractable method of finding the sensor locations. Zhu et al. [74] provide a rigorous proof of this. Intuitively, if all distances between $(d+1)$ points in general position are known, then there are unique locations for these points. Similarly, any point with known distances to at least $(d+1)$ points with known locations also has a unique location.

**Definition 13.** *A group of sensors in dimension $d$ are in* general position *if there are no $(d+1)$ sensors whose locations are affinely dependent.*

That is, the sensor locations $\{x_1, \ldots, x_{(d+1)}\}$ are in general position if the only solution to the system of equations

$$\sum_{i=1}^{d+1} \alpha_i x_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^{d+1} \alpha_i = 0$$

is $\alpha_i = 0$ for all $i = 1, \ldots, d+1$. In other words, if we define the matrix

$$A = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & & 1 \end{pmatrix} \in \mathbb{R}^{(d+1)\times n}, \tag{6.1}$$

the columns of the matrix $X$, or the sensor locations, are in general position if and only if every $(d+1) \times (d+1)$ square sub-matrix of $A$ has rank $(d+1)$.

## 6.3 Related Work

Our work throughout Chapters 7 and 8 relies on that of Zhu et al. [74], which proves that a sensor network inducing a trilateration graph with points in generic position is generically universally rigid, and constructs an algorithm to reduce a graph with a spanning trilateration graph into a trilateration graph. We also use a result of So's PhD thesis [58], which proves that the following are equivalent:

1. The maximum-rank solution matrix $Z$ of (SNL-SDP) has rank $d$.
2. Given the set of known distances, there is a unique set of sensor locations in dimension $d$.
3. The submatrices $X, Y$ of the solution matrix $Z = \Big(I,\ X;\ X^T,\ Y\Big)$ satisfy $Y = X^T X$.

Chapter 7 finds a lower bound on the radio range of sensors in a network to ensure (SNL-SDP) provides their correct locations. So et al. [59] attempted to create such a bound by analyzing the area within the range of each sensor; this work illustrates the difficulty in such a method. Thus, our analysis divides the sensor network area into sub-regions, a method similar to those in [6, 30, 73].

In 2005, Aspnes et al. [6] used graph rigidity theory to construct an asymptotic lower bound on the sensor radio range to ensure there is a unique set of sensor locations when the network induces a trilateration graph. Because this bound is asymptotic, it only holds as the number of sensors in the given area approaches infinity. More recently in 2010, Javanmard et al. [36] also constructed a bound on the sensor radio range; this is a weaker bound than ours, but considers noisy measurements.

We note that the localization of sensors in a network of vehicles is a distinctive challenge, because vehicles (or sensors) may be continually moving. We refer the curious reader to Jin's PhD thesis [37], which creates software (SpaseLoc) to solve (SNL-SDP) for large-scale problems by breaking a sensor network into smaller subproblems. Jin's thesis also develops a dynamic version of SpaseLoc to estimate the locations of moving sensors in a real-time environment and applies the method to bus transit systems.

### 6.3.1 Our Contribution

Chapters 7 and 8 construct conditions for unique localizability based on $(d+1)$-lateration graphs.

In Chapter 7, we assume each sensor (or anchor) can detect the distance from itself to each sensor (or anchor) within a given *radio range*. Our result is a lower bound on this range to ensure the induced graph contains a spanning $(d+1)$-lateration graph, with high probability. Determining this range may be a key component in designing sensors of SVs; given the density of vehicles (and static objects) equipped with sensors in an area, the radio range of the sensors will control the number of known distances and hence the localizability of the vehicles.

In Chapter 8, we create an algorithm that constructively proves the stress matrix of the network has rank $n$, if the induced graph on $n$ sensors in dimension $d$ has points in *general position* that form a $(d+1)$-lateration graph. This proof provides a certificate of unique localizability for networks that induce this class of graphs.

# 7 Sensor Radio Range

In this chapter, we establish a lower bound on the required radio range of randomly distributed sensors in order for the network to be uniquely localizable with high probability. Commonly, sensors are able to detect the distance to other sensors within a given connectivity (or radio) range. The graph induced from such a network is called a *unit-disk graph*.

**Definition 14.** *A sensor network induces a* unit-disk graph *if the distance between two sensors is known if and only if the sensors are within a distance of $r$ of each other.*

For a sensor network that induces a unit-disk graph, the number of sensors that can be correctly localized increases with the radio range. Angluin et al. use an asymptotic analysis to explain this phenomenon in [5] when the sensor locations are uniformly distributed in a unit-square. We present a *non-asymptotic* bound on the radio range requirement of the sensors in order to ensure the network is uniquely localizable.

We consider a unit-disk graph induced by a given concentration of sensors uniformly distributed throughout the specified area. Our non-asymptotic bound is constructed by deconstructing the area into sub-regions and analyzing the network using these sub-regions.

Define $r(p)$ to be the radio range of the sensors that ensures the network is uniquely localizable with probability at least $p$. We find a lower bound on $r(p)$ such that the induced unit-disk graph will have a spanning $(d+1)$-lateration graph.

## 7.1 Decomposition of Network Area

Let $\mathcal{H}$ be the area in dimension $d$ that contains the sensor network. Without loss of generality, we can assume $\mathcal{H}$ is a unit hypercube of dimension $d$, i.e., $\mathcal{H} = [0,1]^d$. We divide $\mathcal{H}$ into $M$ sub-hypercubes in dimension $d$, say $h_1, h_2, \ldots, h_M \subset \mathcal{H}$, where each sub-hypercube $h_i$ will have a volume of $1/M$, and the length of each of its edges will be $\ell := 1/\sqrt[d]{M}$. This partition will allow us to analyze the localizability of sub-hypercubes in the region and their contained points, as opposed to analyzing the localizability of each point individually. By breaking the region into small enough regions, we can use a divide-and-conquer method to analyze the entire network.

Without loss of generality, we can assume $M = b^d$, where $b$ is a positive integer and $b \geq 3$. Similarly, if the region considered is a hyper-rectangle in dimension $d$, we can assume $M = b_1 \cdot b_2 \cdots b_d$, where $b_i \geq 3$ for $i = 1, \ldots, d$ positive integers. Figure 7.1 shows a unit hypercube in dimension $d$ split into $M = 25$ sub-hypercubes.

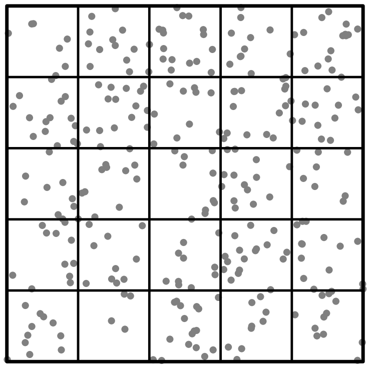


Figure 7.1: Example of a Partitioned Hypercube

Let the number of sensors in each sub-hypercube be binomially distributed. Specifically, let $Y_i \sim B\left(n, \frac{1}{M}\right)$ be the number of sensors placed in sub-hypercube $h_i$, for $i = 1, \ldots, M$. Assume the $Y_i$ sensors in sub-hypercube $h_i$ are arbitrarily placed in general position.

Let $S = \sum_{i=1}^{M} Y_i$ denote the total number of points in $\mathcal{H}$. Because the $Y_i$ values are independently and identically distributed and each sub-hypercube has equal area, the total number of points will be approximately uniformly distributed in $\mathcal{H}$. By properties of the binomial distribution,

$$\begin{aligned} \mathrm{E}\left[S\right] &= M \cdot \mathrm{E}\left[Y_1\right] = M\left(\frac{n}{M}\right) = n \\ \mathrm{Var}\left(S\right) &= M \cdot \mathrm{Var}\left(Y_1\right) = M \cdot \left[\frac{n}{M}\left(1 - \frac{1}{M}\right)\right] = n\left(1 - \frac{1}{M}\right). \end{aligned}$$

Thus, $(S/n) \to 1$ almost surely. For sufficiently large $M$, this binomial distribution of sensor points throughout each sub-hypercube is statistically equivalent to a uniform distribution of $n$ points throughout the region $\mathcal{H}$.

## *7.2 Radio Range Bound*

In this section, we construct a bound on the radio range $r(p)$ that ensures the graph induced from the sensor network is a $(d+1)$-lateration graph, and hence is uniquely localizable. This lower bound on $r(p)$ depends on the number of sub-hypercubes $M$, which implicitly creates a bound on $r(p)$, given the density of the network. That is, our bound on $r(p)$ will be in terms of $M$, where $M$ will be chosen to ensure a sufficient number of sensors lie in each sub-hypercube with high probability.

We establish an ordering on the sub-hypercubes $h_i$ for $i = 1, \ldots, M$ that extend to an ordering on the sensor points in each of the hypercubes. This resulting ordering will produce a $(d+1)$-lateration ordering on the points, with probability $p$.

For simplicity, we prove the following lemmas for the case of $d = 2$, and we refer to the sub-hypercubes as sub-squares. We also refer to a $(d+1)$-lateration in dimension $d = 2$ as a *trilateration.*

However, we note that the same analysis can be applied to hypercubes in higher dimensions, and our bound $r(p) \geq 2\ell\sqrt{2}$ in Lemmas 2–4 is analogous to the bound $r(p) \geq 2\ell\sqrt{d}$ in dimension $d$, where $\ell$ is the edge length of each hypercube.

A trilateration ordering starts with a clique on 3 vertices, and extends to the remaining vertices and their respective edges. Thus, we first establish a sufficient condition for the graph induced on the sensor network to contain a 3-clique, then determine sufficient conditions for there to be a trilateration ordering on the remaining vertices.

Given a lower bound on $r(p)$ to ensure theres is a clique in the graph, we extend the bound to account for two possible scenarios:

- each sub-square contains at least one sensor point
- there are sub-squares that do not contain any sensor points.

We determine the probability of a $(d+1)$-lateration on the induced graph for both of these cases. Combining these two probabilities allows us to determine a lower bound on the radio range $r(p)$ to ensure the induced graph contains a spanning $(d+1)$-lateration graph.

### 7.2.1 A Clique in the Graph

We first determine the number of sub-squares $M$ such that, for a given number of sensors, there will be a 3-clique contained in a single sub-square. We then construct a lower bound on $r(p)$ such that all points in a sub-square are connected to each other. This 3-clique in the graph can represent to the permutation indices $\{\pi(1), \pi(2), \pi(3)\}$ of the trilateration ordering.

**Proposition 1.** *Let $\mathcal{H}$ contain $n$ points, and let $r(p) \geq \ell\sqrt{2} = \frac{\sqrt{2}}{\sqrt{M}}$ and $M \leq \frac{n-1}{2}$ (or equivalently $r \geq \frac{\sqrt{2}\sqrt{d}}{\sqrt{n-1}}$). Then there exists at least one clique of 3 points in the unit-disk graph $G(V, E)$.*

*Proof.* The length of the diagonal of each sub-square is $\frac{\sqrt{2}}{\sqrt{M}}$. If $r(p)$ is lower bounded by the given value, each point in sub-square $h_i$ will be connected to every point in the same sub-square $h_i$.

If there are at most $\frac{n-1}{2}$ sub-squares, then by the pigeon-hole principle, at least one sub-square contains at least 3 points. Therefore, when $M \leq \frac{n-1}{2}$, there is a clique of 3 points in at least one sub-square in the unit-disk graph with a radius $r(p) \geq \ell\sqrt{2}$. □

### 7.2.2 A $(d+1)$-lateration ordering

Given the density of the sensors, we establish conditions on $r(p)$ to ensure each remaining point can be assigned a permutation index $\pi(j)$ and is connected to at least 3 points in $\{\pi(1), \pi(2), \ldots, \pi(j-1)\}$.

**Definition 15.** *For an integer $k > 1$, we say $k$ sub-squares are* sequential *if they are: a) in the same row and $k$ consecutive columns, or b) in the same column and $k$ consecutive rows.*

**Definition 16.** *Sub-squares that share an edge or a vertex are called neighbors. If two sub-squares share an edge, they are called* simple *neighbors. If two sub-squares do not share any edges, but share a vertex, they are called* adjacent *neighbors.*

Define $\Pi_j := \{\pi(1), \pi(2), \ldots, \pi(j)\}$ to be a permutation of a *subset* of the vertices in $G(V, E)$ that form a trilateration ordering. By Proposition 1, if $r(p) \geq \ell\sqrt{2}$, then this permutation contains at least 3 points that form a clique; that is, there is a permutation $\Pi_j$ on a subset of $j \geq 3$ of the points that is a trilateration ordering.

In the following lemmas, we take the ordering $\Pi_j$ on a subset of the points, and build on it to construct a trilateration ordering on all points in the induced graph, with probability $p$.

**Lemma 2.** *Let $\Pi_j := \{\pi(1), \pi(2), \ldots, \pi(j)\}$ be a permutation of a subset of the points in $G(V, E)$ that forms a trilateration ordering. Assume that each sub-square $h_i \in \mathcal{H} \in \mathbb{R}^2$ contains at least one point, and $r(p) \geq 2\ell\sqrt{2}$. If three sequential sub-squares in a row (or column) contain points in $\Pi_j$, then there is a permutation of all points in the corresponding columns (or rows) of the three sequential sub-squares, $\{\pi(j+1), \pi(j+2), \ldots, \pi(k)\}$, such that the augmented permutation of the vertices $\Pi_k = \{\pi(1), \ldots, \pi(j), \pi(j+1), \ldots, \pi(k)\}$ is also a trilateration ordering.*

*Proof.* We prove this lemma by induction on the number of corresponding columns (or rows) whose points are also in the trilateration ordering $\Pi_j = \{\pi(1), \ldots, \pi(j)\}$, and use Figure 7.2 as an example. The lower bound $2\ell\sqrt{2}$ is twice the length of the diagonal of each sub-square, so that $r(p) \geq 2\ell\sqrt{2}$ ensures all points in a given sub-square are connected to all points in neighboring sub-squares.

Consider the example in Figure 7.2 as the base case: if the points in the three sequential sub-squares $\{h_{11}, h_{12}, h_{13}\}$ are in the trilateration ordering $\Pi_j$, then there is an augmented permutation that includes the points in $\{h_{21}, h_{22}, h_{23}\}$ and is also a trilateration ordering. Specifically, if we permute the points in $\{h_{21}, h_{22}, h_{23}\}$ via the ordering on their containing sub-squares:

$$\{\pi(j+1), \ldots, \pi(k)\} := \{h_{22}, h_{21}, h_{23}\},$$

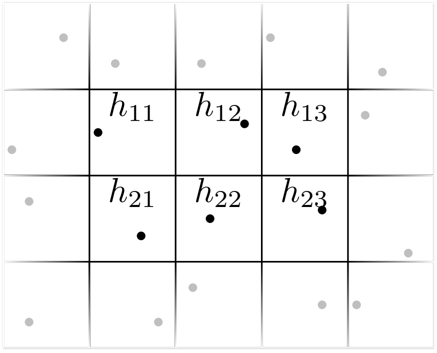


Figure 7.2: A Base-Case Example of a Network with a Trilateration Ordering

then the augmented permutation $\Pi_k := \{\pi(1), \ldots, \pi(j), \pi(j+1), \ldots, \pi(k)\}$ is also a trilateration ordering on the points.

Thus, there is an augmented permutation $\Pi_k$ that includes the points in the sequential sub-squares in a single neighboring row (i.e., $\{h_{21}, h_{22}, h_{23}\}$), and $\Pi_k$ is also a trilateration ordering.

Inductively, such an ordering can continue throughout successive rows. Again, we use Figure 7.2 as an example. Assume that there is an augmented permutation $\Pi_k$ that includes the points in $\{h_{11}, h_{12}, h_{13}, \ldots, h_{\ell,1}, h_{\ell,2}, h_{\ell,3}\}$, for some $\ell \geq 2$. If we permute the points in sub-squares $\{h_{\ell+1,1}, h_{\ell+1,2}, h_{\ell+1,3}\}$ via the ordering on their containing sub-squares:

$$\{\pi(k+1), \ldots, \pi(k+3)\} := \{h_{\ell+1,2}, h_{\ell+1,1}, h_{\ell+1,3}\},$$

then the augmented permutation $\Pi_{k+3} := \{\pi(1), \ldots, \pi(k), \pi(k+1), \ldots, \pi(k+3)\}$ is also a trilateration ordering on the points.

An analogous argument using three sequential sub-squares in the same column would show that if a trilateration $\Pi_j$ contains points in three sequential sub-squares in consecutive rows, then $\Pi_j$ can be augmented with all points in the corresponding rows to form a permutation $\Pi_k$ that is also a trilateration ordering.

Therefore, we have proved via induction that if a trilateration ordering contains points in three sequential sub-squares, then there is an augmented trilateration ordering that contains the points in all sub-squares in the corresponding columns (or rows) of the sequential sub-squares. □

Lemma 3 below analyzes the cases depicted in Figure 7.3, and builds on the augmented trilateration constructed in Lemma 2. Under the assumption that each sub-square contains at least one point, Lemma 3 connects Lemma 2 and Proposition 1 to construct a trilateration on the entire induced graph.

**Lemma 3.** *Assume that each sub-square $h_i \in \mathcal{H}$ contains at least one point, and $r(p) \geq 2\ell\sqrt{2}$. Then the associated unit-disk graph contains a spanning trilateration graph if either:*

1. *there is a non-corner sub-square that contains 3 points*

2. *there is corner sub-square that contains 3 points and one of its neighbor sub-squares has at least two points.*

*Proof.* As in Lemma 2, the lower bound $r(p) \geq 2\ell\sqrt{2}$ ensures all points in a given sub-square are connected to all points in neighboring sub-squares. Using Figure 7.3 as an example, we prove that if either condition of Lemma 3 is satisfied, then there exists a trilateration ordering on the points in the graph.

1. Consider the example in the left grid of Figure 7.3 with a 3-clique in non-corner sub-square $h_{22}$; let this 3-clique form the trilateration ordering $\Pi_3 = \{\pi(1), \pi(2), \pi(3)\}$. If the points in

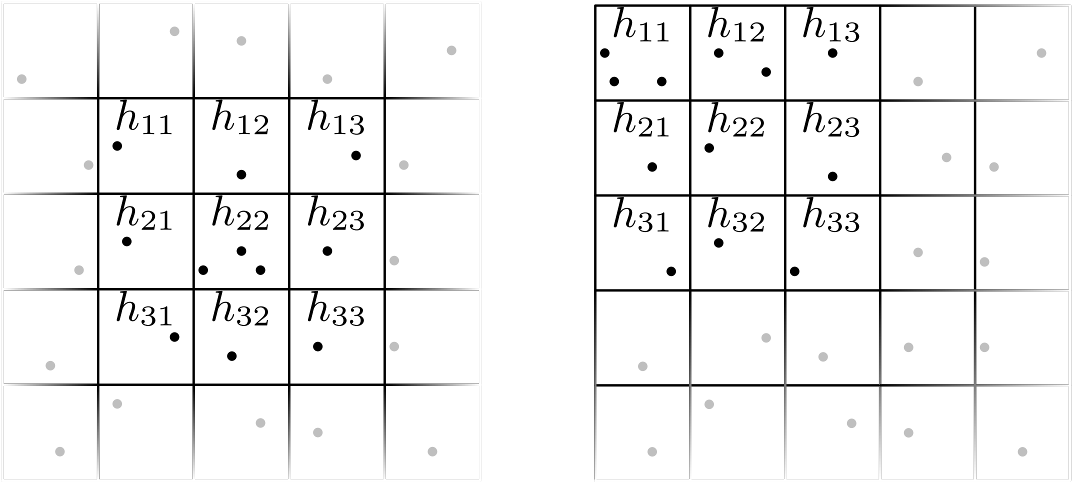


Figure 7.3: Conditions as described in Lemma 3 to ensure trilateration

the sub-squares $\{h_{ij}\}$, for $i, j \in \{1, 2, 3\}$, are augmented to the permutation $\Pi_3$ in *any* order, then the resulting permutation of the points $\Pi_9$ is a trilateration ordering.

By Lemma 2, there is a permutation of the points, say $\{\pi(10), \ldots, \pi(k)\}$, in the corresponding three rows and three columns of sub-squares such that the augmented permutation $\Pi_k := \{\pi(1), \ldots, \pi(9), \pi(10), \ldots, \pi(k)\}$ is a trilateration ordering. Because $\Pi_k$ contains points in three sequential rows and three sequential columns of *every* row and column of the grid, by Lemma 2, the points in every sub-square in the grid can be augmented to $\Pi_k$ to construct an extended trilateration ordering on every point in $\mathcal{H}$.

2. Now consider the right grid of Figure 7.3, where the corner sub-square $h_{11}$ contains a 3-clique, and there are two points in a neighboring sub-square of $h_{11}$. Let the 3-clique in sub-square $h_{11}$ form the trilateration ordering, $\Pi_3 = \{\pi(1), \pi(2), \pi(3)\}$. If the points in the remaining sub-squares are permuted via the ordering on their sub-squares:

$$\{\pi(4), \ldots, \pi(13)\} = \{h_{12}, h_{22}, h_{21}, h_{13}, h_{23}, h_{32}, h_{31}, h_{33}\},$$

then the augmented permutation on the vertices $\{\pi(1), \ldots, \pi(13)\}$ is a trilateration ordering.

Using an analogous argument to part 1 of this proof, by Lemma 2, there is a permutation $\Pi_k$ that contains every point in $\mathcal{H}$ and is a trilateration ordering.

Therefore, if the conditions of Lemma 3 hold, the associated unit-disk graph contains a spanning trilateration graph. □

Lemma 3 provides sufficient, but not necessary, conditions for a trilateration ordering on all the points to exist with high probability, which would imply unique localizability of the sensor network.

However, these are strict conditions for a sensor network, since a random distribution of sensors may not always ensure that each sub-region contains at least one sensor. Thus, we extend these conditions to a more general case, and allow for the possibility of empty sub-squares.

Intuitively, too many empty sub-squares will result in a graph that is not uniquely localizable; if empty sub-squares exist, there must be restricting conditions to ensure the graph is not too sparse to ensure localizability. Thus, we establish additional properties of the graph that ensure a trilateration with high probability, but allow for empty sub-squares.

**Definition 17.** *A sub-square is* densely surrounded *if all its simple neighbors have at least two points, and one of the simple neighbors has at least 3 points.*

**Lemma 4.** *Assume every empty sub-square is densely surrounded and $r(p) \geq 2\ell\sqrt{2}$. Then the associated unit-disk graph contains a spanning trilateration graph if there is a 3-clique in a non-corner sub-square.*

*Proof.* We know that if there is at least one point in every sub-square, then there is a permutation of the vertices (via a permutation on the sub-squares) that is a trilateration ordering. It remains to be shown that if there is an empty sub-square in the grid that satisfies the conditions of Lemma 4, then there is *still* a permutation on the vertices that is a trilateration ordering.

Consider the sample grid in Figure 7.4 as a counterexample, and define a permutation of the points via the ordering on their sub-squares:

$$\Pi_7 := \{h_{11}, h_{12}, h_{21}\}. \tag{7.1}$$

The permutation $\Pi_7$ is the *only* trilateration ordering on a subset of points in the sample grid.

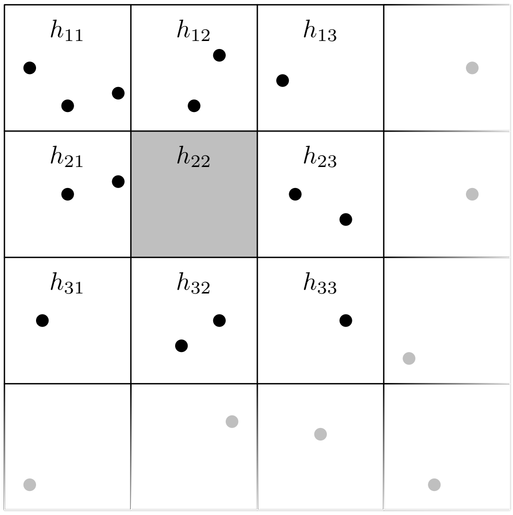


Figure 7.4: Example of a grid that does *not* satisfy the conditions in Lemma 4

That is, $\Pi_7$ cannot be augmented with any other points in the sample grid and still be a trilateration ordering, because none of the sub-squares $\{h_{13}, h_{23}, h_{31}, h_{32}, h_{33}\}$ neighbor a subset of sub-squares in $\Pi_7$ that contain at least 3 points combined. Thus, empty sub-squares must be densely surrounded to ensure a trilateration ordering on the points exists.

The grid in Figure 7.5 shows the worst-case example of the conditions of Lemma 4. The shaded sub-square $h_{22}$ is empty and densely surrounded, and the 3-clique is along the edge of the area $\mathcal{H}$. We prove that a trilateration ordering exists on the points in this sample grid with a densely surrounded sub-square. This proves Lemma 4 holds in the worst-case; the proof that Lemma 4 holds in every case is a generalized extension of this.

Define the permutation on the points in Figure 7.5 via the ordering on their sub-squares:

$$\Pi_{13} := \{h_{12}, h_{11}, h_{13}, h_{21}, h_{23}, h_{32}, h_{31}, h_{33}\}. \tag{7.2}$$

This permutation is a trilateration ordering on the points in the sample grid. Note that $\Pi_{13}$ is a trilateration ordering only because the sub-squares $\{h_{21}, h_{23}\}$ contain at least three points combined; that is, a permutation containing the points in the sub-squares $\{h_{32}, h_{31}, h_{33}\}$ can only be a trilateration if $\{h_{21}, h_{23}\}$ together contain at least three points.

By Lemmas 2 and 3, if there are no other empty sub-squares in $\mathcal{H}$, then there is trilateration ordering on all the points in $\mathcal{H}$. However, if there *are* other densely surrounded empty sub-squares in $\mathcal{H}$, then by a similar construction as (7.2), there is still a trilateration ordering on all points in $\mathcal{H}$.

Therefore, if the condition of Lemma 4 holds, the associated graph contains a spanning trilateration graph. □

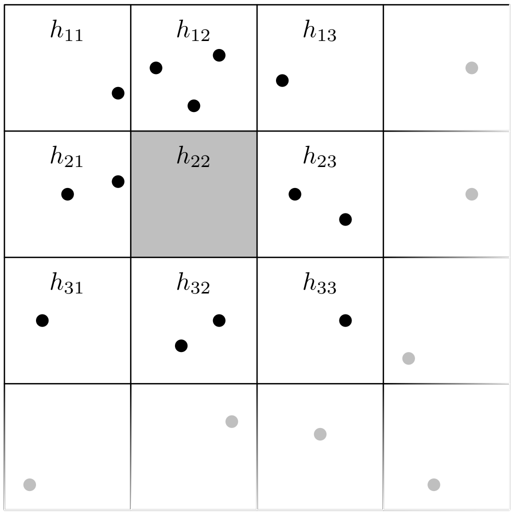


Figure 7.5: Example of a grid that satisfies the conditions in Lemma 4

Lemmas 3 and 4 utilize Lemma 2 to establish sufficient conditions for the graph induced on a sensor network in dimension 2 to contain a spanning trilateration graph. Because such a graph is uniquely localizable [74], these are also sufficient conditions to ensure the solution to the SDP relaxation formulation (SNL-SDP) will provide the correct locations of the sensors. Thus, we use the conditions of Lemmas 3 and 4 to establish a lower bound on the probability that a sensor network that induces a unit-disk graph with radius $r(p) \geq 2\ell\sqrt{2}$ is uniquely localizable.

Define the events:

$A := \{$A network of sensors binomially distributed in its sub-regions is uniquely localizable$\}$,

$B := \{$All empty sub-squares are densely surrounded$\}$,

$C := \{$There is a sub-square containing a 3-clique that is not in a corner$\}$,

$\widehat{C} := C^c = \{$Any sub-square containing a 3-clique is in a corner$\}$.

The probability that a sensor network with points binomially distributed throughout its sub-regions is uniquely localizable is lower-bounded by

$$\mathrm{P}\{A\} = \mathrm{P}\left\{A \mid \widehat{C}\right\}\mathrm{P}\left\{\widehat{C}\right\} + \mathrm{P}\{A \mid C\}\,\mathrm{P}\{C\} \geq \mathrm{P}\left\{A \mid \widehat{C}\right\}\mathrm{P}\left\{\widehat{C}\right\}.$$

We find a lower bound on $\mathrm{P}\{A\}$ by lower bounding the probabilities $\mathrm{P}\{A \mid C\}$ and $\mathrm{P}\{C\}$.

For an area $\mathcal{H}$ divided into $M = b^2$ sub-squares (for some integer $b \geq 3$), there are $(M-4)$ sub-squares in $\mathcal{H}$ that are not in a corner. The probability that there is a sub-square containing a 3-clique that is *not* in a corner is

$$\mathrm{P}\{C\} = 1 - \left(\sum_{i=0}^{2}\binom{n}{i}\left(\frac{1}{M}\right)^{i}\left(1-\frac{1}{M}\right)^{n-i}\right)^{M-4}.$$

To find $\mathrm{P}\{A \mid C\}$, consider an area $\mathcal{H}$ with $k \geq 0$ empty sub-squares. By Lemma 3,

$$\mathrm{P}\{A \mid k=0, C\} = 1,$$

and by Lemma 4, for $i > 0$,

$$\mathrm{P}\{A \mid k=i, C\} \geq \mathrm{P}\{B \mid k=i, C\}.$$

The conditions of Lemma 4 require that empty sub-squares be densely surrounded. To find the probability that an empty sub-square is densely surrounded, we first find the probability that a sub-square does not have empty simple neighbors, and then find the probability that each simple neighbor has at least two points and one simple neighbor has more than two points.

Assume there are $k$ empty sub-squares, say $s_1, s_2, \ldots, s_k$. Because of the independence assumption, these empty sub-squares are uniformly distributed. Given the empty sub-square $s_1$, the probability that $s_2$ is not a simple neighbor of $s_1$ is at least $\left(1 - \frac{4}{M-1}\right)$; the probability that $s_3$ is not a simple neighbor of $s_1$ or $s_2$ is at least $(1 - 2 \cdot \frac{4}{M-2})$; and so on, so that the probability that no two empty sub-squares are neighbors is at least $\prod_{j=1}^{k-1}(1 - \frac{4j}{M-j})$.

The probability that all simple neighbors of an empty sub-square have at least two points and at least one of them has more than two points is

$$\begin{aligned}\hat{p} &= \mathrm{P}\{\text{All simple neighbors have at least two points}\}\\ &\quad - \mathrm{P}\{\text{All simple neighbors have exactly two points}\}\\ &= \left[1 - \binom{n}{1}\left(\frac{1}{M}\right)\left(1 - \frac{1}{M}\right)^{n-1}\right]^4 - \left[\binom{n}{2}\left(\frac{1}{M}\right)^2\left(1 - \frac{1}{M}\right)^{n-2}\right]^4.\end{aligned}$$

Thus, the probability that all empty sub-squares are densely surrounded is

$$\mathrm{P}\{B \mid \ k = i, C\} \geq (\hat{p})^i \cdot \prod_{j=1}^{i-1}(1 - \frac{4j}{M-j}). \tag{7.3}$$

Because the right-hand side of (7.3) is only a valid probability when $i < M/5$, we only consider grids with less than $u := \lfloor M/5 \rfloor - 1$ empty sub-squares. Moreover, for any $i < M - 4$,

$$\mathrm{P}\{C \mid k = i\} \geq 1 - \left(\sum_{j=0}^{2}\binom{n}{j}\left(\frac{1}{M}\right)^j\left(1 - \frac{1}{M}\right)^{n-j}\right)^{M-4-i} := p_{C,i}.$$

The probability that a given sub-square is empty is $p_0 = (1 - \frac{1}{M})^n$, and the probability that $\mathcal{H}$ has $k$ empty sub-squares is

$$\mathrm{P}\{k = i\} = \binom{M}{i} p_0^i \left(1 - p_0\right)^{M-i}.$$

Finally, given a radio range $r(p) \geq 2\ell\sqrt{2}$, and points binomially distributed throughout each sub-square, the probability that a sensor network in $\mathcal{H}$ is uniquely localizable is lower-bounded by

$$\begin{aligned}\mathrm{P}\{A\} &\geq \mathrm{P}\{A|C\}\,\mathrm{P}\{C\} = \sum_{i=0}^{u}\mathrm{P}\{A|k=i,C\}\,\mathrm{P}\{C|k=i\}\,\mathrm{P}\{k=i\}\\ &\geq \sum_{i=0}^{u}\mathrm{P}\{A \mid k=i,C\}\,\mathrm{P}\{k=i\}\,p_{C,i}\\ &\geq p_{C,0}\,\mathrm{P}\{k=0\} + \sum_{i=1}^{u}p_{C,i}\,\mathrm{P}\{k=i\}\times\mathrm{P}\{B \mid k=i,C\}\\ &\geq p_{C,0}\,\mathrm{P}\{k=0\} + \sum_{i=1}^{u}\hat{p}^i \cdot p_{C,i}\,\mathrm{P}\{k=i\}\times\prod_{j=1}^{i-1}(1 - \frac{4j}{M-j}).\end{aligned} \tag{7.4}$$

This result leads to Theorem 2. Note that we can treat sensors that form a 3-clique as anchors, because their locations are known given the distances between them.

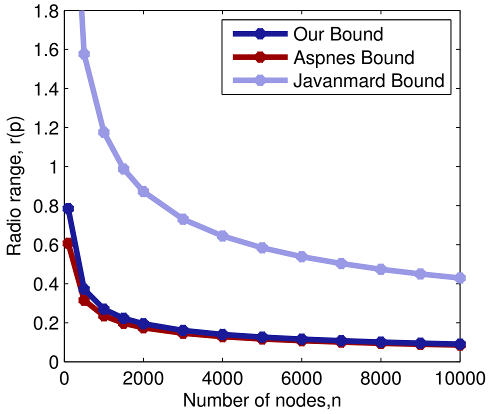


Figure 7.6: Comparison of our Radio Range to that of Javanmard [36] and Aspnes [5]

**Theorem 2.** *Let $\mathcal{H} \in [0,1]^d$ be the unit hypercube in dimension $d$ and partition $\mathcal{H}$ into $M = b^d$ equal sub-hypercubes, say $h_1, h_2, \ldots, h_M \subset \mathcal{H}$, where $\ell = 1/b$ is the edge length of each sub-hypercube. Let the number of sensor points in each sub-hypercube be independently and binomially generated according to $B\left(n, \frac{1}{M}\right)$ where $n \geq M+1$, and let one of the sub-hypercubes contain $d+1$ anchors. Then, if the connectivity radius satisfies $r(p) \geq 2\ell\sqrt{d}$, the probability $p$ that the sensor network is uniquely localizable is given by expression* (7.4).

*Proof.* This follows from the probability analysis above and Lemmas 2, 3, and 4. □

Observe that this lower bound on $\mathrm{P}\{A\}$ is inversely proportional to $M$, which is inversely proportional to $n$. This implies that the probability a network in a given area $\mathcal{H}$ is uniquely localizable is proportional to the number of distributed sensors $n$, which is intuitive.

For a given number of sensors $n$, we can find value of $M$ such that the right-hand side of Equation (7.4) is at least 0.99. Figure 7.6 shows $r(0.99)$ versus the number of sensors $n$ in the unit square such that the SDP relaxation will correctly determine the locations of the sensors.

We compared our radio range lower bound against two other bounds in Figure 7.6. Aspnes' bounds are almost identical to ours for any value of $n$. Thus, our result shows that the bound of Aspnes et al. in [5] (of $r > \frac{2\sqrt{2}\sqrt{\log n}}{\sqrt{n}}$ for $d = 2$) is true even when $n$ is small, although it was initially proved to be an asymptotic bound for $n$ sufficiently large. Our bound, while not in an analytical form, is proved for any value of $n$. Javanmard et al. also constructed an asymptotic bound in [36], which is weaker than ours and Aspnes'.

Therefore, we have constructed a lower bound on the required sensor radio range to ensure the corresponding network is uniquely localizable with probability 0.99. This result can be applied to determining the range of vehicle sensors, or of anchors on a surrounding a road, to determine the vehicles' locations. If sensors and anchors in the area have a sufficient communication range, the relaxation (SNL-SDP) will correctly determine the sensor (or vehicle) locations.

# 8 General Frameworks of Sensor Networks

In this chapter, we construct a matrix purification algorithm that provides a certificate of unique localizability for sensor networks inducing a $(d+1)$-lateration graph with points in general position.

## 8.1 A Sensor Network Framework

Chapter 7 establish conditions on the sensor radio range so that, with high probability, enough pair-wise distances are known to determine the sensor locations. However, the number of known pair-wise distances between sensor is not the only factor that determines unique localizability; the structure of the network also plays a role. For example, consider the extreme case illustrated in Figure 8.1, where the points represent sensors and the edges represent known distances between neighboring sensors. In this example, the sensors are densely placed in an area in a linear formation, and even though a large number of distances may be known, there is more than one set of sensor locations that satisfy the distance constraints (SNL-norm).

We define a *framework* of a sensor network, which allows us to define sufficient properties on the network structure to ensure unique localizability. Let $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{d\times n}$ be the given location matrix of $n$ sensors in dimension $d$.

**Definition 18.** *A graph $G$ and its corresponding location matrix $X$ are called a* framework*, $(G, X)$.*

We again consider a sensor network that consists of $m \geq d+1$ anchor points whose locations $\{a_1, \ldots, a_m\} \in \mathbb{R}^d$ are known, and $n$ sensor points whose locations $\{x_1, \ldots, x_n\} \in \mathbb{R}^d$ are yet to be determined. We assume only a subset of the distances between sensors and anchors are known, and we want to solve (SNL-SDP) to obtain a set of locations that satisfies equations (SNL-norm).

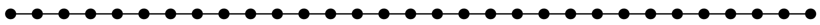


Figure 8.1: An Example of a Sensor Placement that is Not Uniquely Localizable

### 8.1.1 Strong Localizability

To establish conditions on the framework of the sensor network, we need to observe properties of the dual of (SNL-SDP):

$$\begin{array}{ll} \text{minimize} & I_d \cdot V + \sum_{(i,j)\in E_x} w_{ij} d_{ij}^2 + \sum_{(k,j)\in E_a} \bar{w}_{kj} \bar{d}_{kj}^2 \\ \text{subject to} & S := \begin{pmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{(i,j)\in E_x} w_{ij} A_{ij} + \sum_{(k,j)\in E_a} \bar{w}_{kj} \bar{A}_{kj} \succeq \mathbf{0}. \end{array} \qquad \text{(SDP-dual)}$$

Observe that (SDP-dual) is always feasible, because

$$V = \mathbf{0}, \quad w_{ij} = 0 \ \forall (i,j) \in E_x, \quad \text{and} \quad \bar{w}_{kj} = 0 \ \forall (k,j) \in E_a$$

is a feasible solution. Note that since there are no constraints on $V$ in (SDP-dual), any principal $d \times d$ submatrix of a matrix $S$ is feasible as long as $S$ remains positive semidefinite. As discussed in Chapter 6, the problem is *uniquely localizable* if

$$Z = \begin{pmatrix} I_d & X \\ X^T & X^T X \end{pmatrix} \succeq \mathbf{0}$$

is the only solution to (SNL-SDP). In this chapter, we extend the notion of localizability by observing the framework of the network.

**Definition 19.** *The SDP relaxation* (SNL-SDP) *is* strongly localizable *if there is a feasible dual matrix $S$ such that*

$$ZS = \mathbf{0}, \quad S \succeq \mathbf{0} \quad \textit{and} \quad \textit{rank}(S) = n. \tag{8.1}$$

If a sensor network is strongly localizable, then it is uniquely localizable, and a certificate of its unique localizability can be computed in polynomial time. In graph rigidity theory, strong localizability is equivalent to the corresponding stress matrix of the network being positive semidefinite and having rank $n$; unique localizability is equivalent to universal rigidity [3].

In the remainder of this chapter, we construct an algorithm to prove that for a sensor network framework $(G, X)$, if the induced graph $G$ contains a spanning $(d+1)$-lateration graph and the locations $X$ are in general position, then (SNL-SDP) is strongly localizable and the solution can be computed in polynomial time.

## *8.2 An Algorithm to Prove Strong Localizability*

The following algorithm constructs a matrix that is feasible for the dual problem (SDP-dual) and satisfies (8.1). We start with a matrix that satisfies (8.1), and then modify each column so that the resulting matrix is also feasible for (SDP-dual).

### 8.2.1 Format of a Feasible Dual Matrix

We separate an optimal solution matrix of (SDP-dual) into the submatrices

$$S = \begin{pmatrix} V & \bar{W} \\ \bar{W}^T & W \end{pmatrix}, \quad W \in \mathbb{R}^{n \times n}, \quad \bar{W} \in \mathbb{R}^{d \times n},$$

where the elements of $W$ correspond to weights on the sensor-to-sensor edges:

$$W_{ij} = \begin{cases} -w_{ij}, & (i,j) \in E_x \\ 0, & (i,j) \notin E_x \\ \sum_{(i,j)\in E_x} w_{ij} + \sum_{(k,j)\in E_a} \bar{w}_{kj}, & i = j \end{cases} \tag{8.2}$$

and the elements of $\bar{W}$ correspond to weights on the sensor-to-anchor edges:

$$\bar{W}_{ij} = -\sum_{(k,j)\in E_a} \bar{w}_{kj} (a_k)_i. \tag{8.3}$$

Thus, the $j$th diagonal element of $S$ is the negative sum of the weights of $w_{ij}$ and $\bar{w}_{kj}$ on sensor $j$.

### 8.2.2 Matrix Modification Algorithm

In $\mathcal{O}(n^3)$ operations, we construct a matrix $S^\star \in \mathbb{R}^{(d+n)\times(d+n)}$ that is feasible for (SDP-dual) and also satisfies (8.1). The algorithm starts by modifying an initial matrix that satisfies (8.1), and progressively alters each column so that the resulting matrix will be feasible for (SDP-dual).

We assume that the sensors are in general position and that the induced graph is a $(d+1)$-lateration graph. For simplicity, we also assume there are exactly $(d+1)$ anchors that are the first $(d+1)$ points in the $(d+1)$-lateration ordering; all other points are sensors, ordered $1, \ldots, n$, where only edges $(i < j, j)$ are included in $E_x$. The primal solution matrix $Z$ can be written as

$$Z = \begin{pmatrix} I_d & X \\ X^T & X^T X \end{pmatrix} = \begin{pmatrix} I_d \\ X^T \end{pmatrix} \begin{pmatrix} I_d & X \end{pmatrix},$$

which makes it easy to see that $L = \begin{pmatrix} -X; & I_n \end{pmatrix} \in \mathbb{R}^{(d+n)\times n}$ is in the nullspace of $Z$. Moreover,

$$P^n := LL^T = \begin{pmatrix} XX^T & -X \\ -X^T & I_n \end{pmatrix} \succeq \mathbf{0}$$

is in the nullspace of $Z$, and $\operatorname{rank}(P^n) = n$. The matrix $P^n$ satisfies (8.1) and will be the initial matrix for our algorithm, Algorithm 4 on page 65.

Algorithm 4 modifies the elements of $P^n$ to create a matrix $S^\star$ in $\mathcal{O}(n^3)$ operations. There will be $n$ modifications steps (one for each column $\{d+1, \ldots, d+n\}$), each of which create a new matrix,

$P^{n-1}, P^{n-2}, \ldots, P^0$, where $P^\ell$, for $\ell = n-1, \ldots, 0$, will satisfy (8.1). Each alteration requires $\mathcal{O}(n^2)$ operations and modifies a single column, starting with the last column $(d+n)$ and continuing until column $(d+1)$. The resulting matrix $S^\star := P^0$ will satisfy (8.1) and be a feasible solution to (SDP-dual).

### Modification of the last column

We use the modification of column $(d+n)$ of $P^n$ as an example, then generalize the modification for the remainder of the columns. Define the vector

$$p^n := \begin{pmatrix} x_n - \sum_{(k,n)\in E_a} \bar{w}_{kn} a_k \\ e_n - \sum_{(i,n)\in E_x} w_{in} e_i \end{pmatrix} \in \mathbb{R}^{d+n}, \tag{8.4}$$

for some weight variables $(\bar{w}_{kn}, w_{in})$ that will be determined later. The first modification of $P^n$ will create the new matrix

$$P^{n-1} := P^n + p^n (p^n)^T.$$

Let $P^n_j$ denote the $j$th column of $P^n$. Because $p^n_n = 1$, the last column of $p^n(p^n)^T$ is simply $p^n$, and the last column of the modified matrix $P^{n-1}$ is

$$P^{n-1}_{(d+n)} = P^n_{(d+n)} + p_n = \begin{pmatrix} -x_n \\ e_n \end{pmatrix} + \begin{pmatrix} x_n - \sum_{(k,n)\in E_a} \bar{w}_{kn} a_k \\ e_n - \sum_{(i,n)\in E_x} w_{in} e_i \end{pmatrix} = \begin{pmatrix} -\sum_{(k,n)\in E_a} \bar{w}_{kn} a_k \\ 2e_n - \sum_{(i,n)\in E_x} w_{in} e_i \end{pmatrix}.$$

The values of the of the weight variables $\bar{w}_{kn}, \forall (k,n) \in E_a$, and $w_{in}, \forall (i,n) \in E_x$, need to be chosen such that $P^{n-1}_{(d+n)}$ is the last column of a *feasible* matrix for (SDP-dual). That is, the weight variables $(\bar{w}_{kn}, w_{in})$ in $p^n$ must satisfy (8.2) and (8.3). Specifically, the off-diagonal elements in the last column must sum to the diagonal element:

$$\sum_{(k,n)\in E_a} \left( \bar{w}_{kn} \sum_i (a_k)_i \right) + \sum_{(i,n)\in E_x} w_{in} = P^{n-1}_{(d+n),(d+n)} = 2. \tag{8.5}$$

The last column of an *optimal* solution matrix to (SDP-dual) will be in the nullspace of $\begin{pmatrix} I_d & X \end{pmatrix}$, and also satisfy (8.1), if

$$\begin{aligned} \begin{pmatrix} I_d & X \end{pmatrix} p_n &= \begin{pmatrix} I_d & X \end{pmatrix} \begin{pmatrix} x_n - \sum_{(k,n)\in E_a} \bar{w}_{kn} a_k \\ e_n - \sum_{(i,n)\in E_x} w_{in} e_i \end{pmatrix} \\ &= 2x_n - \sum_{(k,n)\in E_a} (\bar{w}_{kn} a_k) - \sum_{(i,n)\in E_x} (w_{in} x_i) = \mathbf{0}. \end{aligned} \tag{8.6}$$

Because $(\bar{X}, X)$ are in general position and $G$ is a $(d+1)$-lateration graph, equations (8.5) and (8.6) combined place exactly $(d+1)$ linearly independent constraints on $(d+1)$ weight variables. Thus, there is a set of unique $(d+1)$ weight variables $\bar{w}_{kn}$ and $w_{in}$ that satisfy this set of equations.

It follows from $ZP^{n-1} = 0$ and $\text{rank}(Z) = d$ that $\text{rank}(P^{n-1}) \leq n$. And $P^{n-1} \succeq P^n \succeq 0$ implies that $\left|\mathcal{N}(P^{n-1})\right| \leq |\mathcal{N}(P^n)| = d$, or equivalently that $\text{rank}(P^{n-1}) \geq n$. Thus, $\text{rank}(P^{n-1}) = n$.

Therefore, this modification has accomplished the objective:

- $ZP^{n-1} = ZP^n + Zp^n(p^n)^T = \mathbf{0}$,
- $P^{n-1} = P^n + p^n(p^n)^T \succeq P^n \succeq \mathbf{0}$,
- $\text{rank}(P^{n-1}) = n$,
- and the last column of $P^{n-1}$ satisfies the conditions to be the last column of a feasible and optimal matrix solution of (SDP-dual).

This modification of last column of $P^n$ creates a new matrix $P^{n-1}$ that satisfies (8.1) and the last column is a feasible and optimal last column of a matrix solution to (SDP-dual).

### General modification of each column

Each remaining column $(d+n-1), \ldots, (d+1)$ of $P^n$ can be modified so that the resulting matrix $S^\star$ satisfies (8.1) and is feasible for (SDP-dual). The remaining steps are slightly different than the first, because the modification of each column $(d+\ell)$ for $\ell = n, \ldots, 1$ alters the entries $P^\ell_{ij}$ for $i, j \leq \ell$, while keeping the entries $P^\ell_{ij}$ for $i, j > \ell$ unchanged. That is, the modification step to obtain the matrix $P^{\ell-1}$ from $P^\ell$ will alter the principal $(d+\ell) \times (d+\ell)$ submatrix of $P^\ell$ so that $P^{\ell-1}_{(d+\ell)}$ is a feasible and optimal column of a matrix solution to (SDP-dual), and $P^{\ell-1}$ will satisfy (8.1). The modification of column $P^{\ell-1}_{(d+\ell)}$ will *not* alter the $(d+\ell+1), \ldots, (d+n)$ columns or rows of $P^\ell$.

For the modification of column $(d+\ell)$ for $\ell = n-1, \ldots, 1$, define the vector $p^\ell \in \mathbb{R}^{d+n}$ as

$$p^\ell = \begin{pmatrix} p^\ell_{1:(d+\ell-1)} \\ p^\ell_{d+\ell} \\ p^\ell_{(d+\ell+1):(d+n)} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} -\sum_{(k,\ell)\in E_a} \bar{w}_{k\ell} a_k \\ -\sum_{(i,\ell)\in E_x} w_{i\ell} e_i \end{pmatrix} - P^\ell_{1:(d+\ell-1),(d+\ell)} \\ 1 \\ \mathbf{0} \end{pmatrix}, \tag{8.7}$$

for weight variables $\bar{w}_{k\ell}$ and $w_{i\ell}$ yet to be determined.

Note that when $\ell = n$, this definition of $p^\ell$ is the same as the definition of $p^n$ in (8.4). Moreover, observe that the updated matrix $P^{\ell-1} := P^\ell + p^\ell(p^\ell)^T$ will be equivalent to the matrix $P^\ell$ in all columns $j$ and rows $i$, for $i, j > (d+\ell)$. The $(d+\ell)$th column of $P^{\ell-1}$ is

$$P^{\ell-1}_{(d+\ell)} = \begin{pmatrix} P^{\ell-1}_{1:(d+\ell-1),(d+\ell)} \\ \\ P^{\ell-1}_{(d+\ell),(d+\ell)} \\ \\ P^{\ell-1}_{(d+\ell+1):(d+n),(d+\ell)} \end{pmatrix} . = \begin{pmatrix} \begin{pmatrix} -\sum_{(k,\ell)\in E_a} \bar{w}_{k\ell} a_k \\ -\sum_{(i,\ell)\in E_x} w_{i\ell} e_i \end{pmatrix} \\ \\ 1 + P^\ell_{(d+\ell),(d+\ell)} \\ \\ P^\ell_{(d+\ell+1):(d+n),(d+\ell)} \end{pmatrix}$$

Again, for $P^{\ell}_{(d+\ell)}$ to be the $(d+\ell)$th column of a *feasible* matrix for the dual problem (SDP-dual), the weight variables $(\bar{w}_{k\ell}, w_{i\ell})$ in $p^{\ell}$ must satisfy (8.2) and (8.3). That is, the off-diagonal elements in the last column must be equal to the diagonal element:

$$\sum_{(k,\ell)\in E_a} \left(\bar{w}_{k\ell} \sum_i (a_k)_i\right) + \sum_{(i,\ell)\in E_x} w_{i\ell} = 1 + P^{\ell}_{(d+\ell),(d+\ell)} + \sum_{i=d+\ell+1}^{d+n} P^{\ell}_{i,(d+\ell)}. \tag{8.8}$$

Similarly, the $(d+\ell)$th column of an *optimal* solution matrix to (SDP-dual) will be in the nullspace of $\begin{pmatrix} I_d & X \end{pmatrix}$:

$$-\sum_{(k,\ell)\in E_a} \bar{w}_{k\ell} a_k - \sum_{(i,\ell)\in E_x} w_{i\ell} x_i + \left(1 + P^{\ell}_{(d+\ell),(d+\ell)}\right) x_{\ell} = P^{\ell}_{1:d,(d+\ell)} + \sum_{i=d+1}^{d+\ell-1} P^{\ell}_{i,\ell} x_i. \tag{8.9}$$

Again, because $(\bar{X}, X)$ are in general position and $G$ is a $(d+1)$-lateration graph, Equations (8.8) and (8.9) combined place exactly $(d+1)$ linearly independent constraints on the $(d+1)$ weight variables, and thus there is a unique solution that determines the weight variables $(\bar{w}_{k\ell}, w_{i\ell})$. Therefore, for each column $(d+\ell)$ this modification accomplishes the objective:

- $ZP^{\ell-1} = ZP^{\ell} + Zp^{\ell}(p^{\ell})^T = \mathbf{0}$,
- $P^{\ell-1} = P^{\ell} + p^{\ell}(p^{\ell})^T \succeq P^{\ell} \succeq \mathbf{0}$,
- $\operatorname{rank}(P^{\ell-1}) = n$,
- and the $(d+\ell)$th column of $P^{\ell-1}$ satisfies the conditions to be the $(d+\ell)$th column of a feasible and optimal matrix solution of (SDP-dual).

Repeating this process on each column, for columns $(d+n), \ldots, (d+1)$, will result in an optimal dual solution matrix $S^{\star} := P^0$ that satisfies (8.1). This algorithm is formalized in Algorithm 4.

**Algorithm 4** Matrix Modification Algorithm

```
Given the primal solution matrix $Z \in \mathbb{R}^{(d+n)\times(d+n)}$ to (SNL-SDP) and the sensor locations
$X \in \mathbb{R}^{d\times n}$, define $L := \begin{pmatrix} -X \\ I_n \end{pmatrix}$ and $P^n := LL^T$.
for $\ell = n, \ldots, 1$ do
    Choose weight variables $\bar{w}_{k\ell}$ and $w_{i\ell}$ that satisfy the set of $(d+1)$ equations (8.8) and (8.9).
    Define $p^{\ell} \in \mathbb{R}^{(d+n)}$ as in (8.7)
    Set $P^{\ell-1} := P^{\ell} + p^{\ell}(p^{\ell})^T$
end for
$S^{\star} \leftarrow P^0$.
```

### 8.2.3 Proof of Strong Localizability

We now prove that a $(d+1)$-lateration graph with points in general position admits a positive semidefinite stress matrix with rank $n$ that can be computed in strongly polynomial time.

**Theorem 3.** *Take a graph $G$ of $m \geq d+1$ anchor points and $n$ sensor points with edges given in $E_x$ and $E_a$, and let $G$ be a $(d+1)$-lateration graph with $(\bar{X}, X)$ in general position. The sensor network localization problem on $G$ is strongly localizable, and a rank-$n$ dual optimal matrix can be computed in strongly polynomial time from $(\bar{X}, X)$, using $\mathcal{O}(n^3)$ arithmetic operations, if the lateration ordering is known.*

*Proof.* Each of the $n$ modification steps in Algorithm 4 performs $\mathcal{O}\left(n^2\right)$ operations, for a total of $\mathcal{O}\left(n^3\right)$ operations. The constructed matrix $S^\star$ satisfies:

- $ZS^\star = \mathbf{0}$,
- $S^\star \succeq \mathbf{0}$,
- $\operatorname{rank}(S^\star) = n$,
- and $S^\star$ will be an optimal solution matrix to the dual problem (SDP-dual).

Because the graph $G$ induced from the sensor network is a $(d+1)$-lateration graph and the locations of the sensors are in general position, there is exactly one possible modification procedure for each column in order to construct $S^\star$. For the sensor network framework $(G, X)$, the matrix modification process described in Algorithm 4 constructs a rank-$n$ dual optimal matrix in strongly polynomial time. Therefore, the sensor network localization problem on $(G, X)$ is strongly localizable. □

We also have the following corollary that generalizes the result of Theorem 3.

**Corollary 1.** *Take a graph $G$ of $m$ $(\geq d+1)$ anchor points and $n$ sensor points with edges given in $E_x$ and $E_a$, and let $G$ contain a $(d+1)$-lateration spanning subgraph, with $(\bar{P}, P)$ in general position. The sensor network localization problem on $G$ is strongly localizable, and a rank-$n$ dual optimal matrix can be computed in polynomial time by solving* (SNL-SDP) *and* (SDP-dual).

This secondary result holds because we can ignore all edges outside the spanning $(d+1)$-lateration subgraph to prove the existence of a positive semidefinite stress matrix with rank $n$. Finding a spanning $(d+1)$-lateration subgraph requires at least $\mathcal{O}(n^{d+2})$ operations, and hence we cannot actually construct such a stress matrix in $\mathcal{O}(n^3)$ operations. However, the interior-point semidefinite programming algorithms will always compute a maximum-rank stress matrix in polynomial time, although not strongly polynomial.

# 9 Conclusion

Future personal vehicles will most likely collaborate to improve standards such as safety, traffic flow, and environmental benefits [15]. This thesis focuses on two types of future vehicles, Plug-in Electric Vehicles and Sensor-equipped Vehicles. We use linear optimization methods to construct algorithms that facilitate communication or energy flow among future vehicle networks.

For Plug-in Electric Vehicles, we use online linear programming techniques to design and simulate an algorithm that utilizes vehicle batteries in a way that benefits both consumers and utilities. We consider a fleet of PEVs with V2G capabilities, and show that the vehicle batteries can be used as a beneficial resource for the electricity grid, while being of minimal cost to the consumer. Our algorithm could dynamically assign robust charging and discharging schedules for each vehicle in a fleet of thousands.

For Sensor-equipped Vehicles, we apply sensor network localization techniques to establish sufficient conditions under which SVs could determine their relative locations given the distances between some of them. Specifically, we establish a lower bound on the radio range of the sensors and conditions on the framework of a network in a given area that ensure a semidefinite programming relaxation of the problem will produce a correct localization of the sensors.

## 9.1 Plug-in Electric Vehicle Network

We construct a dynamic, distributed and robust mechanism to determine energy exchange schedules for a fleet of PEVs. The resulting allocation of electricity trading can meet a market obligation and provide each vehicle with enough energy for its driving load. Our algorithm depends only on the knowledge of a few hundred driving behaviors from a previous similar day and instantly assigns energy exchange schedules to vehicles as they plug-in to the grid, without requiring a back-and-forth communication with the aggregator or other vehicles in the fleet.

We proved existence of a *composite valuation* that equally factors in the benefits to the electricity market and to the consumers. This set of hourly values allows us to create a distributed algorithm, removing the need for a centralized mechanism that requires constant communication with an aggregator and the electricity market.

Our results show that our algorithm using this composite valuation, Algorithm 1, out-performs each comparison algorithm in terms of consumer cost, increase in peak power demand, and total cost for the fleet.

### 9.1.1 Practical Significance

According to the United States Environmental Protection Agency (EPA), personal vehicles contributed nearly 17% of all greenhouse gas emissions in the U.S. in 2009 [67]. The same study by the EPA states that the electric power industry contributes about 33% of all greenhouse gas emissions, making it the largest contributor of greenhouse gases. These emissions are a result of power plants fueled by coal, petroleum, and natural gas.

Renewable energy generation creates electricity from natural resources, such as wind, sun or geothermal heat. Such power generation has a number of benefits: it does not emit greenhouse gases during operation; natural resources are unlimited in quantity and lack the potential for scarcity, unlike fossil fuels; and after capital investment, sustainable resources are cost-free due to their infinite supply.

However, sustainable power is variable and unpredictable; supply of renewable generation cannot be controlled to meet user demand and storage devices are needed that store surplus electricity to be used in times of deficit. Storing surplus power for later use can allow utilities to meet the demand using electricity obtained at the most efficient and cheapest times [70]. A main concern among energy storage analysts is the capital cost of large energy storage devices for this purpose, and the idea of using PEVs as distributed energy storage has become popular in the energy storage field.

Plug-in electric vehicles will run off electricity provided by the grid, which can be generated sustainably and used to run the vehicle without releasing greenhouse gases. The data analysis in [16] shows that on any given day 54% of vehicles are not driven at all. The same work also shows that 90% of vehicles start and end their days at home, implying that there is a substantial window of time every night when the vehicle can be plugged-in with the potential for both charging and discharging while benefiting the consumer. Thus, if PEVs can be used as distributed energy storage devices that facilitate the integration of renewable energy generation into the grid, such a system has the potential to drastically decrease greenhouse gas emissions.

We have designed a mechanism specifically for PEVs that could be used by demand response aggregators to utilize a fleet of PEVs as distributed energy storage devices and promote the integration of renewable energy into the grid. An implementation of our algorithm would provide substantial monetary profits and environmental offsets to motivate consumer participation, while also using PEV batteries to provide grid services.

## *9.2 Sensor-equipped Vehicle Network*

The benefits and implications of sensor-equipped vehicles are abundant, but their implementation presents a number of challenges. SVs will detect the distance from themselves to nearby objects, and if certain conditions are true, this set of known pair-wise distances can be used to compute the relative locations of nearby vehicles. We establish the conditions under which an SDP relaxation is an efficient computational method to correctly solve the sensor network localization problem.

We provide the first nonasymptotic bound on the connectivity radius of randomly distributed sensors to ensure the graph induced from the network is uniquely localizable, with high probability. From our result, the strength of vehicles sensors can be determined. This could have large implications in designing sensors for sensor-equipped vehicles, and in the placement of anchors on surrounding roads.

We also prove that if the graph induced from a network of $n$ sensors in dimension $d$ is a $(d+1)$-lateration graph of points in general position, then the graph admits a positive semidefinite stress matrix with rank $n$, and there is a unique localization of the points. This was a long open question in graph rigidity theory. We show that a network with the specified framework will have a unique set of sensor locations, and a certificate of this uniqueness can be constructed in strongly polynomial time. This result establishes conditions on the framework of the sensors in order for there to be exactly one localization of the sensors in all dimensions.

These results establish the conditions on a sensor network to ensure the solution to the SDP relaxation provides the exact sensor locations. We can apply these conditions to determining the density, radio range, and placement of anchors for SVs and the surrounding road, so that vehicles in the given area can establish their relative location.

### 9.2.1 Practical Significance

Sensor-equipped vehicles may detect situations that a driver may not and operate accordingly to avoid collisions or prevent traffic congestion, which could lead to a number of benefits on an individual and societal level. SVs could drastically decrease the frequency of accidents caused by human error, which cause around 33,000 deaths and 1.2 million injuries each year [44]. The vehicles could also improve fuel efficiency by optimizing driving speeds, and could cooperate to improve traffic flow.

Such intelligent systems in vehicles may also bring us a step closer to autonomous vehicles. General Motors (GM) predicts that autonomous vehicles, or driverless vehicles, will be on the road by 2018 [61]. Google has designed, built and tested a small fleet of driverless vehicles on California highways [44]. By providing guidelines on the design of these sensors and the structure of the networks, we may also be assisting to further the development of driverless vehicle technology.

# Bibliography

[1] Abdo Y. Alfakih. On the universal rigidity of generic bar frameworks. *Contribution to Discrete Mathematics*, 5(3):7–17, 2010.

[2] Abdo Y. Alfakih, Amir Khandani, and Henry Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. In *Computational Optimization and Applications*, volume 12, pages 13–30, 1999.

[3] Abdo Y. Alfakih, Nicole Taheri, and Yinyu Ye. On stress matrices of $(d+1)$-lateration frameworks in general position. *Mathematical Programming*, pages 1–17, 2011.

[4] Erling D. Andersen. The homogeneous and self-dual model and algorithm for linear optimization: MOSEK technical report. 2009.

[5] Dana Angluin, James Aspnes, Melody Chan, Michael J. Fischer, Hong Jiang, and René Peralta. Stably computable properties of network graphs. In *Distributed Computing in Sensor Systems: First IEEE International Conference, DCOSS 2005, Marina del Rey, CA, USE, June/July, 2005, Proceedings*, volume 3560 of *Lecture Notes in Computer Science*, pages 63–74. Springer-Verlag, 2005.

[6] James Aspnes, Tolga Eren, David K. Goldenberg, A. Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian D. O. Anderson, and Peter N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.

[7] Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless network localization. In *IPSN*, pages 46–54, 2004.

[8] Blink Network. Smart charging solutions at home, 2012. `http://www.blinknetwork.com/chargers-residential.html`.

[9] Ward Bower. International guideline for the certification of photovoltaic system components and grid-connected systems. 2002. Report IEA PVPS T5-06.

[10] Bureau of Transportation Statistics: Research and Innovative Technology Administration. Number of U.S. aircraft, vehicles, vessels, and other conveyances, 2007. `http://www.bts.gov/publications/national_transportation_statistics/html/table_01_11.html`.

[11] CAISO. California ISO OASIS, 2011. `http://www.oasis.caiso.com`.

[12] Sharon Silke Carty. Blind spot monitoring coming to mainstream cars, 2010. `http://content.usatoday.com/communities/driveon/post/2010/07/blind-spot-monitoring-systems-coming-to-mainstream-cars/1#.T8hxFYlR2lg`.

[13] Andrew Chen, Behrooz Khorashadi, Chen-Nee Chuah, Dipak Ghosal, and Michael Zhang. Smoothing vehicular traffic flow using vehicular-based ad hoc networking & computing grid (VGrid). In *Proceedings of the IEEE ITSC 2006*. IEEE Intelligent Transportation Systems Conference, 2006.

[14] Robert Connelly. Generic global rigidity. *Discrete and Computational Geometry*, 33(4):549–563, 2005.

[15] Tyler Cowen. Can I see your license, registration and C.P.U? *The New York Times*, 2011. `http://www.nytimes.com/2011/05/29/business/economy/29view.html`.

[16] Barbara Morgan Davis. Understanding the effects and infrastructure needs of plug-in electric vehicle (PEV) charging. Master's thesis, Colorado State University, 2010.

[17] General Electric. Charging ahead. 2012. `http://www.ge.com/innovation/electric_vehicles/index.html`.

[18] Energy Almanac. California average weekly retail gasoline prices. `http://energyalmanac.ca.gov/gasoline/retail_gasoline_prices.html`.

[19] EnerNOC. Demand response: A multi-purpose resource for utilities and grid operators. Technical report, 2009.

[20] EnerNOC. The demand response baseline. Technical report, 2011.

[21] Environmental Protection Agency. Emission facts: Average annual emissions and fuel consumption for passenger cars and light trucks. 2007. `http://www.epa.gov/oms/consumer/f00013.htm`.

[22] EPRI. Environmental assessment of plug-in hybrid electric vehicle: Volume 2: United States air quality analysis based on AEO-2006 assumptions for 2030. 2007. 1015326.

[23] EPRI. Impact of plug-in electric vehicle technology diffusion on electricity infrastructure: Preliminary analysis of capacity and economic impacts. 2008. 1016853.

[24] EPRI. Impact of plug-in hybrid electric vehicles on utility distribution systems. 2009. 1018698.

[25] EPRI. The power to reduce $CO_2$ emissions: The full portfolio: 2009 technical report. 2009. 1020389.

[26] EPRI. Vehicle-to-grid (V2G) data dictionary: A resource for acquiring data on V2G and related fields. 2011. 1022461.

[27] Eric Evarts. Average age of cars is increasing. 2009. `http://news.consumerreports.org/cars/2009/03/average-car-age-increasing-older-cars-on-the-road.html`.

[28] Eric Evarts. Leaf, Volt tests show electric cars cost less per mile to operate. 2011. `http://news.consumerreports.org/cars/2011/12/leaf-volt-tests-show-electric-cars-cost-less-per-mile-to-operate.html`.

[29] Federal Energy Regulatory Commission. 2008 assessment of demand response & advanced metering. Technical report, 2008.

[30] Aram Galstyan, Bhaskar Krishnamachari, Kristina Lerman, and Sundeep Pattem. Distributed online localization in sensor networks using a moving target. In *Proceedings of the third international symposium on information processing in sensor networks*, pages 61–70. ACM Press, 2003.

[31] GPS.gov. GPS accuracy: Official U.S. government information about the global positioning system (GPS) and related topics, 2011. `http://www.gps.gov/systems/gps/performance/accuracy/`.

[32] Michael Grant. CVX: Matlab software for disciplined convex programming, 2012. `http://cvxr.com/cvx/`.

[33] Bengt Halvorson. Affordable cars could soon offer collision-avoidance systems, 2012. `http://www.thecarconnection.com/news/1050012_affordable-cars-could-soon-offer-collision-avoidance-systems`.

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

[35] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM Journal on Computing*, 21(1):65–84, 1992.

[36] Adel Javanmard and Andrea Montanari. Localization from incomplete noisy distance measurements. 2011. `http://arxiv.org/abs/1103.1417v3`.

[37] Holly Hui Jin. *Scalable Sensor Localization Algorithms for Wireless Sensor Networks.* PhD thesis, University of Toronto, 2005.

[38] Ben Klayman and John Crawley. Electric vehicle battery costs coming down: Chu. *Reuters*, 2012. `http://www.reuters.com/article/2012/01/11/us-autoshow-batteries-idUSTRE80A1FA20120111`.

[39] Nathan Krislock and Henry Wolkowicz. Explicit sensor network localization using semidefinite representations and clique reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010.

[40] National Renewable Energy Laboratory. Learning about renewable energy: Distributed energy basics. 2009. `http://www.nrel.gov/learning/eds_distributed_energy.html`.

[41] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[42] Zhongjing Ma, Duncan Callaway, and Ian Hiskens. Decentralized charging control for large populations of plug-in electric vehicles: Application of the Nash certainty equivalence principle. *2010 IEEE International Conference on Control Applications*, pages 191–195, 2010.

[43] John Markoff. Google cars drive themselves, in traffic. *New York Times*, 2010. `http://www.nytimes.com/2010/10/10/science/10google.html?pagewanted=all`.

[44] John Markoff. Collision in the making between self-driving cars and how the world works. *The New York Times*, 2012. `http://www.nytimes.com/2012/01/24/technology/googles-autonomous-vehicles-draw-skepticism-at-legal-symposium.html`.

[45] Mercedes-Benz. Mercedes-Benz: PARKTRONIC, 2012. `http://www4.mercedes-benz.com/e/home/innovation/recentdevelopments/safety/fahrzeug/parktronic.htm`.

[46] Jim Motavalli. In a blackout, Nissan, Mitsubishi and Toyota E.V.'s could function as generators. 2011. `http://wheels.blogs.nytimes.com/2011/09/01/in-a-blackout-nissan-mitsubishi-and-toyota-e-v-s-could-function-as-generators/`.

[47] General Motors. 2012 Chevy Volt electric car features and specs. `http://www.chevrolet.com/volt-electric-car/features-specs/`.

[48] Nissan. The new car: Features and specifications, 2012. `http://www.nissanusa.com/leaf-electric-car/specs-features/index#/leaf-electric-car/specs-features/index`.

[49] United States Department of Transportation: Federal Highway Administration. Average annual miles per driver by age group, 2011. `http://www.fhwa.dot.gov/ohim/onh00/bar8.htm`.

[50] Pacific Gas and Electric. Electric schedule E-6: Residential time-of-use service, 2011. `http://www.pge.com/tariffs/`.

[51] Pacific Gas and Electric. Company profile, 2012. `http://www.pge.com/about/company/profile/`.

[52] Matthew J. Rutherford and Vahid Yousefzadeh. The impact of electric vehicle battery charging on distribution transformers. *Applied Power Electronics Conference and Exposition 2011*, pages 396–400, 2011.

[53] SICK AG. Distance measurement: From micron to mile– SICK covers the distance. 2006.

[54] Ramteen Sioshansi. Welfare impacts of electricity storage and the implications of ownership structure. *The Energy Journal*, 31(2):189–214, 2010.

[55] Ramteen Sioshansi and Paul Denholm. Emissions impacts and benefits of plug-in hybrid electric vehicles and vehicle to grid services. *Environmental Science and Technology*, 43(4):1199–1208, 2009.

[56] Ramteen Sioshansi and Paul Denholm. The value of plug-in hybrid electric vehicles as grid resources. *The Energy Journal*, 31(3):1–22, 2010.

[57] Ramteen Sioshansi and Jacob Miller. Plug-in hybrid electric vehicles can be clean and economical in dirty power systems. *Energy Policy*, 39(10):6151–6161, 2011.

[58] Anthony Man-Cho So. *A Semidefinite Programming Approach to the Graph Realization Problem: Theory, Applications and Extensions.* PhD thesis, Stanford University, 2007.

[59] Anthony Man-Cho So and Yinyu Ye. On solving coverage problems in a wireless sensor network using Voronoi diagrams. pages 584–593, 2005.

[60] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. *Symposium on Discrete Algorithms*, pages 405–414, 2005.

[61] Chuck Squatriglia. GM says driverless cars could be on the road by 2018. *Autopia*, 2008. http://www.wired.com/autopia/2008/01/gm-says-driverl.

[62] Jos R. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. 2001.

[63] The University of Delaware. The grid-integrated vehicle with vehicle to grid technology. 2011. http://www.udel.edu/V2G/index.html.

[64] United States Department of Transportation, Federal Highway Administration. 2009 National Household Travel Survey, 2009. http://nhts.ornl.gov.

[65] United States Energy Information Administration. Electricity Data, 2009. http://www.eia.gov/electricity/data.cfm.

[66] United States Energy Information Administration. How much electricity does an American home use? 2011. http://www.eia.gov/tools/faqs/faq.cfm?id=97&t=3.

[67] United States Environmental Protection Agency. Inventory of U.S. greenhouse gas emissions and sinks: 1999–2009. 2011.

[68] Battery University. The secrets of battery runtime. http://batteryuniversity.com/learn/article/the_secrets_of_battery_runtime.

[69] John Voelcker. November plug-in car sales: Volt sparks, Leafs fall. 2011. `http://www.greencarreports.com/news/1070075_november-plug-in-car-sales-volt-sparks-leafs-fall`.

[70] Todd Woody. PG&E's battery power plans could jump start electric car market. 2007. `http://thegreenwombat.com/2007/06/12/pges-battery-power-plans-could-jump-start-electric-car-market/`.

[71] Di Wu, Dionysios C. Aliprantis, and Lei Ying. Load scheduling and dispatch for aggregators of plug-in electric vehicles. *IEEE Transactions on Smart Grid*, Special Issue on Transportation Electrification and Vehicle-to-Grid Applications, 2011.

[72] Xiaomin Xi, Ramteen Sioshansi, and Vincenzo Marano. A stochastic dynamic programming model for co-optimization of distributed energy storage. *Submitted to Management Science*, 2012.

[73] Shuhui Yang, Jie Wu, and Fei Dai. Localized movement-assisted sensor deployment in wireless sensor networks. In *Mobile Adhoc and Sensor Systems*, pages 753–758, 2006.

[74] Zhisu Zhu, Anthony Man-Cho So, and Yinyu Ye. Universal rigidity: Towards accurate and efficient localization of wireless networks. In *IEEE INFOCOM*, 2010.