University of California
Santa Barbara

Safe and Secure Optimization in Human-Cyber-Physical Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Berkay Turan

Committee in charge:

Professor Mahnoosh Alizadeh, Chair
Professor Ramtin Pedarsani
Professor João Hespanha
Professor Francesco Bullo
Professor César Uribe

December 2023

The Dissertation of Berkay Turan is approved.

Professor Ramtin Pedarsani

Professor João Hespanha

Professor Francesco Bullo

Professor César Uribe

Professor Mahnoosh Alizadeh, Committee Chair

December 2023

Safe and Secure Optimization in Human-Cyber-Physical Systems

by

Berkay Turan

To my mother, Muradiye Turan.

Acknowledgements

I am deeply grateful as I convey my heartfelt appreciation to my research advisor, Mahnoosh Alizadeh. She's much more than a mentor; she's a cherished friend. Along this path, her unwavering support for my research and ideas, coupled with her steadfast guidance, has played a pivotal role in my growth within her group. Thanks to her, I now complete my Ph.D. studies, armed with invaluable skills and a profound research vision, ready to guide me through my career. I couldn't have hoped for a more exceptional advisor.

I'd like to express my sincere appreciation to Ramtin Pedarsani for the invaluable insights and wisdom that have illuminated my academic journey. As a collaborator, committee member, and esteemed professor, he has been a tremendous source of inspiration for me, combining remarkable intelligence with a delightful sense of humor.

I'd also like to extend my gratitude to Francesco Bullo and Joao Hespanha for being part of my committee and being generous with their expertise and time. Special thanks to Ali Emre Pusane for being a driving force of motivation during my undergraduate studies, which ultimately led me to apply for a Ph.D.

I want to thank my dedicated collaborators, Cesar Uribe and Hoi-To Wai. Their generous sharing of their extensive knowledge in the realm of theoretical research has been invaluable to me. Their mentorship and perspectives have enriched my intellectual accumulation, and I am truly fortunate to have had the privilege of working alongside such exceptional mentors and colleagues.

My sincere thanks go also to my labmates and fellow ECE students at the University of California, Santa Barbara for their support and camaraderie.

Being far from my family, I was extremely lucky to have found another Turkish family in Santa Barbara. I would like to thank Furkan, Asutay, Mert, Metehan, Arınç, Zeki, İlayda, Anıl, and Oya for the great memories and their lifelong friendships.

Lastly, words cannot capture the profound depth of gratitude and love I hold for my beloved mother, Muradiye Turan, for the unwavering love and unconditional support she has given me in pursuit of my dreams. Her sacrifice, her enduring love, and the countless years spent apart from her beloved son weigh on my heart. Losing her life partner at a very young age, she dedicated herself to raising her only two sons. Amid life's relentless challenges, she, as a teacher, consistently placed education above all else, steering me toward the path of becoming the best version of myself. I will never truly repay her boundless efforts and selflessness, but I will persistently strive to keep her heart full of pride and happiness.

Curriculum Vitæ

Berkay Turan

Education

2023	Ph.D. in Electrical and Computer Engineering (Expected), University of California, Santa Barbara.
2020	M.Sc. in Electrical and Computer Engineering, University of California, Santa Barbara.
2018	B.Sc. in Electrical and Electronics Engineering, Boğaziçi University.
2018	B.Sc. in Physics, Boğaziçi University.

Publications

Journal Publications

1. **B. Turan**, S. Hutchinson, and M. Alizadeh, "A Safe First-Order Method for Pricing-Based Resource Allocation in Safety-Critical Networks", *submitted to the IEEE Transactions on Control of Network Systems*, 2023 [1].
2. **B. Turan** and M. Alizadeh, "Competition in electric autonomous mobility-on-demand systems", *IEEE Transactions on Control of Network Systems*, 2022 [2].
3. **B. Turan**, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "Robust distributed optimization with randomly corrupted gradients", *IEEE Transactions on Signal Processing*, 2022 [3].
4. **B. Turan**, R. Pedarsani, and M. Alizadeh, "Dynamic pricing and fleet management for electric autonomous mobility on demand systems", *Transportation Research Part C: Emerging Technologies*, 2020 [4].
5. **B. Turan**, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "Resilient primal-dual optimization algorithms for distributed resource allocation", *IEEE Transactions on Control of Network Systems*, 2020 [5].

Conference Proceedings

1. S. Hutchinson, **B. Turan**, and M. Alizadeh, "The impact of the geometric properties of the constraint set in safe optimization with bandit feedback", *Learning for Dynamics and Control Conference*, 2023 [6].
2. S. Hutchinson, **B. Turan**, and M. Alizadeh, "A safe pricing mechanism for distributed resource allocation with bandit feedback", *IEEE 61st Conference on Decision and Control*, 2022 [7].
3. A. Moradipari, **B. Turan**, Y. Abbasi-Yadkori, M. Alizadeh, and M. Ghavamzadeh, "Feature and parameter selection in stochastic linear bandits", *International Conference on Machine Learning*, 2022 [8].

4. **B. Turan** and M. Alizadeh, "Safe dual gradient method for network utility maximization problems", *IEEE 61st Conference on Decision and Control*, 2022 [9].
5. **B. Turan**, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "On robustness of the normalized random block coordinate method for non-convex optimization", *60th IEEE Conference on Decision and Control*, 2021 [10].
6. **B. Turan**, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "On robustness of the normalized subgradient method with randomly corrupted subgradients", *IEEE American Control Conference*, 2021 [11].
7. N. Tucker, **B. Turan**, and M. Alizadeh, "Online charge scheduling for electric vehicles in autonomous mobility on demand fleets", *IEEE Intelligent Transportation Systems Conference*, 2019 [12].
8. **B. Turan**, N. Tucker, and M. Alizadeh, "Smart charging benefits in autonomous mobility on demand systems", *IEEE Intelligent Transportation Systems Conference*, 2019 [13].

Abstract

Safe and Secure Optimization in Human-Cyber-Physical Systems

by

Berkay Turan

In our rapidly evolving technological landscape, the proliferation of enabling technologies for autonomous systems has given rise to a burgeoning realm of societal-scale smart systems. One noteworthy category within this domain is Human-Cyber-Physical Systems (H-CPS), which encompass physical systems controlled by a blend of computer-based algorithms and human inputs. Examples of H-CPS include the smart grid and autonomous transportation systems. These systems harness the potential of distributed computing units, fast communication channels, and real-time data collection, offering efficient mechanisms for their management. This requires a synthesis of tools from distributed optimization, machine learning, game theory, and stochastic control.

However, the advent of H-CPS also presents novel challenges. Human decisions, often stochastic and beyond direct control, must be factored into the developed mechanisms. Moreover, the dependable operation of H-CPS hinges on secure communication between physical systems and computing units, raising concerns regarding user data privacy and system security. The growing number of humans and devices generates copious amounts of data from sensing units, necessitating computationally efficient data processing to ensure seamless H-CPS operation.

This thesis aims to design network control, optimization, and learning frameworks that enhance safety, robustness, and efficiency in H-CPS, with practical applications in smart infrastructure systems like the power grid and transportation networks. Additionally, its relevance extends to diverse Internet of Things applications, emphasizing user data privacy, such

as the development of language models from text data. The thesis unfolds in three interconnected chapters. In the first chapter, we introduce provably efficient and adversarially robust multi-agent optimization algorithms tailored for distributed resource allocation and distributed learning scenarios in the presence of malicious agents. Moving forward to the second chapter, we aim to design prices for shared resources that do not violate hard (mainly physical) constraints of the system, without any two-way communications with the users as common in distributed optimization based methods. The third chapter focuses on crafting and analyzing joint ride pricing and fleet management policies for the control of autonomous urban mobility fleets. Throughout these chapters, we not only analyze the theoretical performance of our proposed mechanisms but also substantiate their effectiveness through extensive simulations on real-world problems.

Contents

Chapter 1

Introduction

In our age of swiftly advancing technology led by artificial intelligence, the fusion of Human-Cyber-Physical Systems (H-CPS) has paved the way for the development of transformative societal-scale smart infrastructure systems. H-CPS are a class of complex systems that integrate physical components, computational elements, and human decision-making. These systems are designed to operate and interact in the real world, connecting the digital and physical realms, such as smart infrastructure systems, intelligent transportation systems, and the Internet of Things. In H-CPS, the computational elements in the digital realm play a crucial role, as their overarching goal is to continuously collect data from the physical realm and implement efficient optimization algorithms for decision-making and learning to improve the performance of the system.

The integration of optimization frameworks within H-CPS brings up novel challenges that need to be addressed for the efficient and secure operation of such systems. As mentioned earlier, efficient operation heavily depends on optimization, however, H-CPS are often multi-agent systems respecting user privacy. This necessitates the optimization of the system to be performed collaboratively and in a distributed fashion, where the users and the system operator try to find the optimal operation parameters through back-and-forth communication with min-

imum private information exchange. Accordingly, the success of optimization algorithms in distributed setups relies on 1) trustworthy user behavior and 2) reliable communication channels. In case either user behavior or communication channels become corrupted (e.g., due to corrupted user data in distributed learning setups or man-in-the-middle attacks to hack the communication channels), then the operation of the H-CPS becomes compromised and security becomes a major concern.

Furthermore, stochastic components of H-CPS (e.g., due to unpredictable and uncertain human behavior) raise concerns about reliability and safety when optimizing the efficiency of the systems. In systems where the operation has to abide by safety constraints but is also impacted by human decisions that can not directly be controlled (e,g., demand response programs where power grid safety is critical but power generation/consumption can only be impacted through incentives), the optimization algorithms have to predict user behavior and account for the uncertainties in order to ensure safety. Another application where uncertain behavior impacts performance is ride-sharing systems, where the reliability of the system is measured by customer demand fulfillment rate. The efficient and reliable operation of such transportation systems depends on coupled optimization of ride prices and idle vehicle relocation decisions while accounting for stochastic customer demand.

Given the aforementioned challenges inherently present in H-CPS, the purpose of this thesis is to develop state-of-the-art optimization, control, and learning frameworks for promoting efficiency and security in societal-scale H-CPS, with applications in resource management in smart infrastructure systems, intelligent transportation systems, and multi-user distributed learning.

1.1 Summary of Contributions

Safe and Robust Pricing for Distributed Resource Allocation

Security of optimization algorithms for distributed resource allocation is important for the seamless operation of many safety-critical systems such as the electric power grid. The shared goal in these systems is to allocate the resources to the users such that the total utility of the users is maximized. However, the user-specific utilities for accessing the resources are private information, and therefore a direct allocation of the resources by the provider is not possible. Accordingly, distributed optimization methods have become suitable tools, where users determine their resource consumption based on prices, and the prices are updated based on all the users' resource consumption levels. Because these resource allocation schemes are actual physical systems with physical constraints (e.g., the capacity of a power line), the prices should ensure that the users' resource consumption will not overload the system (e.g., exceeding the capacity of a power line might cause a fire). Without any two-way communication with the users to access their private information (e.g., related to their private utility functions), designing prices for resources that allow the users to freely determine their profit-maximizing resource demand, while simultaneously meeting the hard constraints of the system, becomes challenging. As such, this thesis proposes a safe pricing mechanism that induces user demand always satisfying the constraints, while promoting efficient utilization of the resources.

In addition, such distributed optimization methods require reliable communication between the users and the resource provider and can be affected severely when this communication becomes unreliable (e.g., the electricity meter of a household always reads 0 while the house consumes a lot of power). In such cases, the resource provider might think that there are under-utilized resources (while there are none) due to miscommunication and can decide to reduce the prices, which in turn will cause the users to consume more resources. This will cause an overload of the system and damage the physical hardware, which might lead to a seri-

ous disaster. To prevent this, this thesis establishes robust distributed optimization algorithms for resource allocation that are resilient to attacks and manipulations on the communication channels.

Adversarially Robust Distributed Learning

Many learning applications such as image classification, which is one of the significant problems for autonomous driving, or language models, which are used applied for text prediction, rely on user data. In such applications, the goal is to develop a model that optimizes a certain performance metric on a given dataset. Due to privacy, it is not feasible to collect the user data and create the globally optimal model in a centralized manner. Instead, the models are optimized in a distributed fashion, where each user communicates the information on how to update the global model to optimize the performance for their own data to a central machine, which then aggregates the collective information to update the global model.

Given the distributed nature of this framework that heavily relies on trustworthy communication between the users and the central machine, security and robustness become major concerns. If some of the users communicate corrupted information intentionally or unintentionally due to corrupted data or cyber-attacks, the global model can become arbitrarily corrupted. Accordingly, this thesis proposes distributed optimization algorithms that are robust to corruptions on the communicated information in multi-agent learning problems.

Fleets of Autonomous Electric Vehicles for Urban Mobility

Shared use of autonomous electric vehicles is a promising candidate for future urban mobility thanks to the advancements in electric vehicle and autonomous driving technologies. With the urban population projected to reach 60 percent of the world population by 2030, private cars are widely recognized as unsustainable for the future of personal urban mobility. Unlike personal vehicles that are operated by a driver to provide mobility services, shared autonomous vehicles

can be controlled centrally, and idle vehicles can be efficiently relocated to locations with demand without depending on the drivers' decisions. Additionally, electric vehicles provide opportunities for cheap and environment-friendly energy resources (e.g., solar energy). By exploiting the temporal and geographical differences in electricity prices, the operation costs of this fleet can be minimized. All in all, the deployment of a fleet of autonomous electric vehicles enables more efficient transportation services at potentially lower costs compared to the current services, which would benefit both the customers and the service provider. However, these benefits would heavily depend on optimally controlling the fleet for routing and charging and optimally determining the ride prices. Therefore, it is crucial to develop controllers that can make real-time decisions by adapting to the uncertainties and changes in the environment. To this end, this thesis proposes reinforcement-learning based control policies for control of such fleets.

Additionally, this thesis offers a rigorous theoretical analysis of duopolistic competition involving two firms offering transportation services through autonomous electric vehicle fleets. The derived closed-form mathematical expressions quantify the competition's effects on ride prices, firm profits, aggregate demand served, and consumer surplus. These findings can assist investors in making informed decisions about competing platforms and efficient urban mobility technology investments.

1.2 Chapter Overviews

Chapter 1

Chapter 1 presents the motivations for this thesis, summary of main contributions, and chapter overviews.

Chapter 2

Chapter 2 presents results on robust distributed optimization algorithms in presence of malicious agents.

Section 2.2 studies attack-resilient distributed algorithms for resource allocation systems based on primal-dual optimization when Byzantine attackers are present in the system. In particular, we design attack-resilient primal-dual algorithms for static and dynamic impersonation attacks by means of robust statistics. For static impersonation attacks, we formulate a robustified optimization model and show that our algorithm guarantees convergence to a neighborhood of the optimal solution of the robustified problem. On the other hand, a robust optimization model is not required for the dynamic impersonation attack scenario and we are able to design an algorithm that is shown to converge to a near-optimal solution of the original problem. We analyze the performances of our algorithms through both theoretical and computational studies.

Section 2.3 proposes a first-order distributed optimization algorithm tailored to learning applications that is provably robust to Byzantine failures—arbitrary and potentially adversarial behavior, where all the participating agents are prone to failure. We model each agent's state over time as a two-state Markov chain that indicates Byzantine or trustworthy behaviors at different time instants. We set no restrictions on the maximum number of Byzantine agents at any given time. We design our method based on three layers of defense: 1) temporal robust aggregation, 2) spatial robust aggregation, and 3) gradient normalization. We study two

settings for stochastic optimization, namely Sample Average Approximation and Stochastic Approximation. We provide convergence guarantees of our method for strongly convex and smooth non-convex cost functions and provide numerical evidence demonstrating the efficacy and robustness of RANGE in the proposed setting.

Chapter 3

Chapter 3 introduces a novel algorithm for solving network utility maximization (NUM) problems that arise in resource allocation schemes over networks with known safety-critical constraints, where the constraints form an arbitrary convex and compact feasible set. Inspired by applications where customers' demand can only be affected through posted prices and real-time two-way communication with customers is not available, we require an algorithm to generate "safe prices". This means that at no iteration should the realized demand in response to the posted prices violate the safety constraints of the network. Thus, in contrast to existing distributed first-order methods, our algorithm, called safe pricing for NUM (SPNUM), is guaranteed to produce feasible primal iterates at all iterations. At the heart of the algorithm lie two key steps that must go hand in hand to guarantee safety and convergence: 1) applying a projected gradient method on a shrunk feasible set to get the desired demand, and 2) estimating the price response function of the users and determining the price so that the induced demand is close to the desired demand. We ensure safety by adjusting the shrinkage to account for the error between the induced demand and the desired demand. In addition, by gradually reducing the amount of shrinkage and the step size of the gradient method, we prove that the primal iterates produced by the SPNUM achieve a sublinear static regret of $\mathcal{O}(\log(T))$ after T time steps.

Chapter 4

Chapter 4 presents studies on the use of fleets of autonomous electric vehicles for urban mo-

bility.

Section 4.2 considers the joint routing, battery charging, and pricing problem faced by a profit-maximizing transportation service provider that operates a fleet of autonomous electric vehicles. We first establish the static planning problem by considering time-invariant system parameters and determine the optimal static policy. While the static policy provides stability of customer queues waiting for rides even if consider the system dynamics, we see that it is inefficient to utilize a static policy as it can lead to long wait times for customers and low profits. To accommodate for the stochastic nature of trip demands, renewable energy availability, and electricity prices and to further optimally manage the autonomous fleet given the need to generate integer allocations, a real-time policy is required. The optimal real-time policy that executes actions based on full state information of the system is the solution of a complex dynamic program. However, we argue that it is intractable to exactly solve for the optimal policy using exact dynamic programming methods and therefore apply deep reinforcement learning to develop a near-optimal control policy. The two case studies we conducted in Manhattan and San Francisco demonstrate the efficacy of our real-time policy in terms of network stability and profits, while keeping the queue lengths up to 200 times less than the static policy.

Section 4.3 investigates the impacts of competition in autonomous mobility-on-demand systems. By adopting a network-flow based formulation, we first determine the optimal strategies of profit-maximizing platform operators in monopoly and duopoly markets, including the optimal prices of rides. Furthermore, we characterize the platform operator's profits and the consumer surplus. We show that for the duopoly, the equilibrium prices for rides have to be symmetric between the firms. Then, in order to study the benefits of introducing competition in the market, we derive universal theoretical bounds on the ratio of prices for rides, aggregate demand served, profits of the firms, and consumer surplus between the monopolistic and the duopolistic setting. We discuss how consumers' firm loyalty affects each of the aforementioned metrics. Finally, using the Manhattan network and demand data, we quantify the efficacy of

static pricing and routing policies and compare them to real-time model predictive policies.

Chapter 5

Chapter 5 presents the conclusions of this thesis and future directions.

Chapter 2

Adversarially Robust Multi-Agent Distributed Optimization Algorithms

2.1 Introduction

This chapter delves into the intricate intersection of distributed optimization and security, with a specific emphasis on its robustness within the context of cyber-physical systems. Our exploration encompasses a wide spectrum of multi-agent distributed optimization frameworks, including resource allocation systems that fall under the general umbrella of Network Utility Maximization (NUM) problems as well as distributed learning.

The NUM problems manifest in diverse domains, ranging from the classic example of congestion control in data networks [14] to the optimization of electricity pricing and demand-supply balancing in smart power distribution networks [15, 16]. Additionally, they extend to user transmission management in wireless cellular networks [17, 18], optimal caching policies by content delivery networks [19], and power optimization in energy-restricted wireless sensor networks [20, 21], along with congestion control systems in urban traffic networks [22]. In these contexts, the common objective is to minimize the aggregate cost functions of N users

while adhering to coupling constraints, all while preserving the privacy of user-specific cost functions and coordination constraints. Scholars have advocated the use of primal-dual optimization methods in these distributed scenarios, as they naturally facilitate decomposable algorithms conducive to distributed implementation [23]. Notably, these methods enjoy both practical success and strong theoretical foundations, ensuring rapid convergence to near-optimal solutions [24]. In addition, distributed optimization has emerged as an attractive tool for scholars in the field of distributed learning, owing to its applicability in large-scale data processing, privacy preservation, and the potential for parallel algorithm execution [25, 26, 27].

Nevertheless, the inherent distributed nature of these methods, involving physically separated servers or agents connected over networks, exposes the system to vulnerabilities distinct from their traditional centralized counterparts [28]. Ensuring the robustness and security of distributed methods becomes imperative when evaluating algorithmic performance [26]. In a centralized system, one can typically rely on data cleanliness, faultless computation, and the presence of reliable hardware, with minimal communication requirements. Conversely, distributed algorithms often make assumptions of trustworthy data, error-free computation, and dependable communication channels. Additionally, privacy constraints may preclude data corruption checks, and the distributed computing infrastructure can increase the likelihood of encountering faulty hardware, such as personal devices [29]. Moreover, unreliable communication can occur due to factors like noisy wireless communication or, more critically, due to man-in-the-middle adversarial attacks. In man-in-the-middle attacks, adversaries can take control of network sub-systems and manipulate the information exchanged between machines, potentially obstructing convergence to optimal solutions, as seen in Byzantine attacks [30].

In this chapter, we focus on designing robust distributed optimization algorithms for scenarios where communication channels are susceptible to adversarial attacks. These attacks can compromise the stability of distributed systems, potentially leading to hardware damage and system-wide disruptions. Our objective is to illuminate strategies for enhancing the resilience

and security of distributed optimization algorithms for resource allocation (Section 2.2) and learning (Section 2.3) while bridging the gap between theoretical foundations and practical implementation in the face of these multifaceted challenges.

2.2 Resilient Primal-Dual Optimization Algorithms for Distributed Resource Allocation

In this section, our goal is to design attack-resilient primal-dual algorithms in order to solve multi-agent resource allocation problems in presence of Byzantine attackers. If a communication channel is attacked and becomes compromised, the attacker can modify messages and/or inject fresh messages into the network on the agents' behalf. We consider two scenarios with different attacker capabilities. A static impersonation attack scenario considers the set of agents communicating through compromised channels to be the same for the duration of the algorithm, whereas a dynamic impersonation attack scenario considers the case where all agents are susceptible to attacks and hence communicate through compromised channels for a *limited fraction* of the algorithm's runtime. Our main contributions are as follows:

- We propose resilient distributed resource allocation algorithms under the two aforementioned attack scenarios that rely on robust mean estimation.
- We provide convergence guarantees of the proposed algorithms. We show that our algorithm for the dynamic impersonation attack scenario converges to the optimal solution of the regularized problem, while our algorithm for the static impersonation attack scenario converges to an $\mathcal{O}(\alpha_1^2)$ neighborhood of the optimal solution of a robustified and regularized optimization model, where $\alpha_1 \in [0, \frac{1}{2})$ is a known upper bound on fraction of attacked channels.
- We provide empirical evidence that supports our theoretical results on convergence and preventing constraint violation. We do so via computational simulations on electric vehicle charging and power distribution applications.

Related work: Vulnerabilities of various types of distributed algorithms have been identified and addressed in a number of recent studies. Relevant examples can be found in [31, 32, 33, 34,

35, 36, 37, 38, 39] which study secure decentralized algorithms on a general network topology but consider consensus-based optimization models. There are two fundamental differences between distributed resource allocation and consensus problems that make these algorithms inapplicable in our case:

- In resource allocation problems, each agent is solving for their own optimal level of resource consumption, i.e., each agent is solving for their own parameter, whereas consensus problems focus on all agents solving for a shared (global) parameter.
- Unlike resilient consensus algorithms, in resource allocation problems pertaining to access to critical infrastructure systems such as power or transportation networks, one cannot simply block a set of users' access to the network even if they are deemed likely to be attackers.

A recently popular line of works in [40, 41, 42, 43, 44] focuses on building resilient algorithms for distributed statistical learning. A crucial difference from the work presented in this section is that they assume identical functions across the agents. In fact, we employ robust statistics [45, 46] to develop our resilient algorithms, and particularly, we develop novel results for robust mean estimation, a topic that is recently rekindled in [47, 48, 49].

Organization: In Subsection 2.2.1, we provide an overview of the basic primal-dual algorithm for resource allocation. In Subsection 2.2.2, we formally define two Byzantine attack models and demonstrate how Byzantine attacks can alter the primal-dual optimization procedure. In Subsection 2.2.3, we present two attack-resilient primal-dual algorithms corresponding to the different attack scenarios along with their convergence analysis. In Subsection 2.2.4, we provide numerical results for our algorithms.

Notations. Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm. For any $N \in \mathbb{N}$, $[N]$ denotes the finite set $\{1, ..., N\}$. Given $\boldsymbol{\theta}$, $\boldsymbol{\theta}_i$ indicates the i'th block/entry of $\boldsymbol{\theta}$ that corresponds to the parameter of agent i. $\boldsymbol{\theta}_{i,j}$ denotes the j'th element of vector $\boldsymbol{\theta}_i$.

Algorithm 1: PD-DRA Procedure.

1: **for** $k = 1, 2, ...$ **do**

2: *(Communication stage)*:

(a) Central coordinator receives $\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N$ from agents and computes $\overline{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}$, $\{\nabla_{\boldsymbol{\theta}} g_t(\overline{\boldsymbol{\theta}}^{(k)})\}_{t=1}^T$.

(b) Central coordinator broadcasts the vector $\overline{\boldsymbol{g}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} g_t(\overline{\boldsymbol{\theta}}^{(k)})$.

3: *(Computation stage)*:

(a) Agent i computes the update for $\boldsymbol{\theta}_i^{(k+1)}$ according to (2.4a) using the received $\overline{\boldsymbol{g}}^{(k)}$.

(b) The central coordinator computes the update for $\boldsymbol{\lambda}^{(k+1)}$ according to (2.4b).

4: **end for**

2.2.1 Overview of Primal-Dual Algorithm for Resource Allocation

We consider the following multi-agent optimization problem with an objective to minimize the average cost incurred by the agents, subject to a set of constraints that are functions of the average of the agents' parameters:

$$\begin{aligned} \min_{\boldsymbol{\theta}_i \in \mathbb{R}^d, \forall i} \quad & f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^N f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & g_t\left(\tfrac{1}{N}\textstyle\sum_{i=1}^N \boldsymbol{\theta}_i\right) \leq 0, \quad t = 1, \ldots, T, \\ & \boldsymbol{\theta}_i \in \mathcal{C}_i, \qquad\qquad\quad i = 1, \ldots, N, \end{aligned} \tag{2.1}$$

where $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is the continuously differentiable and convex cost function of agent i and $g_t(\cdot) : \mathbb{R}^d \to \mathbb{R}$ are continuously differentiable and convex set of constraints. The parameter $\boldsymbol{\theta}_i$ of agent i is constrained to be in a compact convex set $\mathcal{C}_i \in \mathbb{R}^d$.

Running Example 2.2.1 (Resource Allocation Problem) *Throughout the rest of this section, we use the following toy example as a running example to clarify the concepts and the meth-*

ods: We consider an EV charging example with 5 agents. The cost function $f_i(\cdot)$ is monotone decreasing and is the same for all agents. As an example, we set $f_i(\boldsymbol{\theta}_i) = (\boldsymbol{\theta}_i - 10)^2$ as the quadratic cost function which is monotonically decreasing for $0 \leq \boldsymbol{\theta}_i \leq 10$. There is a charging station with 5 EV charging points, three of which have a maximum charging rate of 7kW, and two have a rate of 10kW. The total rate at which the charging station is able to deliver electricity is determined by the grid, and let it be upper bounded by 25kW (hence, the average rate is upper bounded by $\frac{25}{5} = 5$kW). Accordingly, the constraints of this system are stated as:

$$
\begin{aligned}
& g\big((1/5)\textstyle\sum_{i=1}^{5} \boldsymbol{\theta}_i\big) := (1/5)\textstyle\sum_{i=1}^{5} \boldsymbol{\theta}_i - 5 \leq 0, \\
& 0 \leq \boldsymbol{\theta}_i \leq 7, \quad i = 1, 2, 3, \\
& 0 \leq \boldsymbol{\theta}_i \leq 10, \quad i = 4, 5.
\end{aligned}
$$

Note that $\boldsymbol{\theta}$ is a real number, hence dimension $d = 1$. The optimal solution in this example is to deliver electricity at a rate of 5kW to all agents due to symmetry.

The optimization problem in (2.1) can not be solved centrally, because the utility functions $f_i(\cdot)$ are private to the agents, and furthermore the coupling constraints on the resources are only known by a central coordinator. Accordingly, the goal of the primal-dual distributed resource allocation (PD-DRA) procedure in Algorithm 1 is to solve (2.1) in a distributed manner, where the agents observe a pricing signal received from the central coordinator and communicate their parameters to the central coordinator [24]. Consequent to this information exchange, the pricing signal and the agents' parameters are updated by the central coordinator and by the individual agents, respectively.

In order to derive the update rules used by Algorithm 1, we first consider the Lagrangian

function of (2.1):

$$\mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N;\boldsymbol{\lambda}) := \frac{1}{N}\sum_{i=1}^N f_i(\boldsymbol{\theta}_i) + \sum_{t=1}^T \lambda_t g_t\Big(\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i\Big), \tag{2.2}$$

where $\lambda_t \geq 0$ is the dual variable associated with constraint $g_t(\cdot)$ and $\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_T]^\intercal \in \mathbb{R}_+^T$ is the vector of the dual variables. Under strong duality (e.g., when the Slater's condition holds), solving problem (2.1) is equivalent to solving its dual problem:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i,\forall i} \mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N;\boldsymbol{\lambda}). \tag{P}$$

As suggested in [24], we consider a regularized version of (P). Let us define

$$\mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i\}_{i=1}^N;\boldsymbol{\lambda}) := \mathcal{L}(\{\boldsymbol{\theta}_i\}_{i=1}^N;\boldsymbol{\lambda}) + \tfrac{\upsilon}{2N}\textstyle\sum_{i=1}^N \|\boldsymbol{\theta}_i\|^2 - \tfrac{\upsilon}{2}\|\boldsymbol{\lambda}\|^2, \tag{2.3}$$

such that $\mathcal{L}_\upsilon(\cdot)$ is υ-strongly convex and υ-strongly concave in $\{\boldsymbol{\theta}_i\}_{i=1}^N$ and $\boldsymbol{\lambda}$, respectively.

Remark 2.2.1 *Adding regularization terms is a typical technique used in optimization, called dual smoothing[50]. We add the regularization terms for the purposes of convergence analysis used in this section, which can be applied to strongly convex/concave functions. Indeed adding the regularization terms might change the solution of the original optimization problem. However, as explained in [51, Proposition 5.2], by an appropriate selection of the regularization parameters, we can recover an optimality gap guarantee for the original problem based on the solution to the regularized problem.*

We define the regularized problem as:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i,\forall i} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i\}_{i=1}^N;\boldsymbol{\lambda}). \tag{P$_\upsilon$}$$

Let $\gamma > 0$ be the step size and $k \in \mathbb{Z}_+$ be the iteration index. The primal-dual recursion performs projected gradient descent/ascent on the primal/dual variables as follows:

$$\boldsymbol{\theta}_i^{(k+1)} = \mathcal{P}_{\mathcal{C}_i}\big(\boldsymbol{\theta}_i^{(k)} - \gamma \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\big),\ \forall i \in [N], \tag{2.4a}$$

$$\boldsymbol{\lambda}^{(k+1)} = \big[\boldsymbol{\lambda}^{(k)} + \gamma \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\big]_+, \tag{2.4b}$$

where $\mathcal{P}_{\mathcal{C}_i}(\cdot)$ is the Euclidean projection operator to set $\mathcal{C}_i$ and $[\cdot]_+$ denotes $\max\{\cdot, 0\}$ operator. According to (2.3), the gradients with respect to (w.r.t.) the primal and the dual variables are given respectively by:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)}) = \frac{1}{N}\Big(&\nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon \boldsymbol{\theta}_i^{(k)} \\ &+ \textstyle\sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} g_t(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}}\Big),\end{aligned} \tag{2.5a}$$

$$\big[\nabla_{\boldsymbol{\lambda}} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N; \boldsymbol{\lambda}^{(k)})\big]_t = g_t\Big(\tfrac{1}{N}\textstyle\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}\Big) - \upsilon \lambda_t^{(k)}, \tag{2.5b}$$

for all i, t. It is worthwhile to highlight that both gradients depend on the average parameter $\overline{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^{(k)}$. From the above equations (2.5a) and (2.5b), we can determine which variables should be communicated between the central coordinator and the agents so that the gradients can be computed locally, see Algorithm 1.

Since the regularized primal-dual problem is strongly convex/concave in primal/dual variables, Algorithm 1 converges linearly to the optimal solution of (P_υ) [24]. To study this, let us concatenate the primal and the dual variables and denote $\boldsymbol{z}^{(k)} := (\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})$ as the primal-dual variable at the kth iteration and define the mapping $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ as:

$$\boldsymbol{\Phi}(\boldsymbol{z}^{(k)}) := \begin{pmatrix} \nabla_{\boldsymbol{\theta}} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)}) \\ -\nabla_{\boldsymbol{\lambda}} \mathcal{L}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)}) \end{pmatrix}. \tag{2.6}$$

Proposition 2.2.1 *[24, Theorem 3.5] Assume that the map $\Phi(\boldsymbol{z}^{(k)})$ is L_Φ Lipschitz continuous. For all $k \geq 1$, we have*

$$\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 \leq (1 - 2\gamma\upsilon + \gamma^2 L_\Phi^2)\,\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2, \tag{2.7}$$

where $\boldsymbol{z}^\star$ is a saddle point to the (P_υ)*. Setting $\gamma = \upsilon/L_\Phi^2$ gives*

$$\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 \leq \left(1 - \upsilon^2/L_\Phi^2\right)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2,$$

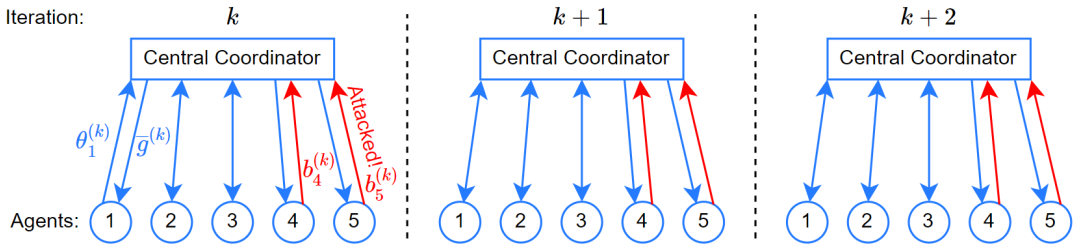
$\forall\ k \geq 1$.

2.2.2 Problem Formulation

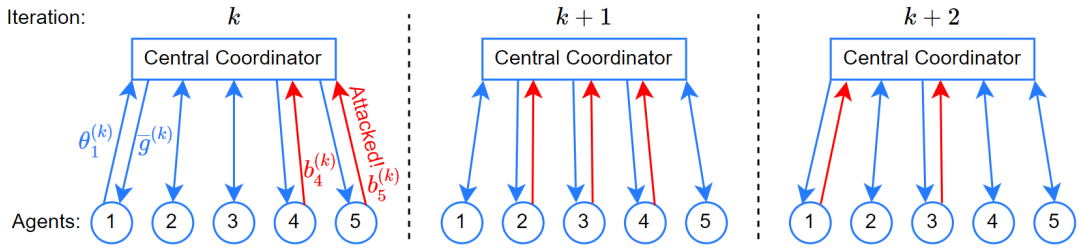
Even though the PD-DRA provides strong theoretical convergence guarantee, it relies on error-free communication between the central coordinator and the agents, and is not robust to attacks on the channels between the agents and the central coordinator, as described below.

We study a situation when the *uplink* communication channels between some of the agents and the central coordinator are compromised.[1] Let $\mathcal{A}^{(k)} \subset [N]$ be the set of agents communicating through *compromised uplink channels* at iteration k, whose identities are unknown to the central coordinator, and let $\mathcal{H}^{(k)} := [N] \setminus \mathcal{A}^{(k)}$ be the set of agents communicating through *trustworthy uplink channels* at iteration k. Instead of receiving $\boldsymbol{\theta}_i^{(k)}$ from each agent $i \in [N]$ at

[1]The work presented in this section studies the case where only uplink channels are compromised. However, the case of downlink corruption can also be addressed. Since the downlink channel is a broadcast channel, a compromised downlink channel results in no agent receiving a trustworthy pricing signal. In that case, there is no optimization method based solution to that problem since there is no communication. If we assume however that all the downlink channels are point-to-point between the central coordinator and each agent, the methods developed in this section can be applied in a similar fashion.



(a) A static impersonation attack scenario, agents 4 and 5 are permanently communicating through compromised channels.



(b) A dynamic impersonation scenario, where the set of agents communicating through compromised channels are changing.

Figure 2.1: Illustration of (a) static impersonation attack, and (b) dynamic impersonation attack. Blue arrows represent trustworthy channels, whereas red arrows represent compromised channels.

iteration k (Algorithm 1 Step 2(a)), the central coordinator receives the following messages:

$$\boldsymbol{r}_i^{(k)} = \begin{cases} \boldsymbol{\theta}_i^{(k)}, & \text{if } i \in \mathcal{H}^{(k)}, \\ \boldsymbol{b}_i^{(k)}, & \text{if } i \in \mathcal{A}^{(k)}. \end{cases} \tag{2.8}$$

We consider a Byzantine attack scenario, under which the messages sent through the compromised channels, $\boldsymbol{b}_i^{(k)}$, can be chosen arbitrarily by an adversary. This also encompasses faulty messages due to erroneous inputs or erroneous channels, since we set no restrictions on $\boldsymbol{b}_i^{(k)}$. The adversary's goal is to harm the system and cause suboptimalities. When the messages are erroneous or chosen adversarily, the central coordinator computes the gradients and therefore the pricing signal using these erroneous messages. The agents then update their parameters based on this erroneous pricing signal, which can lead to an overall suboptimal resource allocation. Moreover, the choice of the compromised channels $\mathcal{A}^{(k)}$ affects the impact of the attack

and the precautions to be taken in order to defend against the attack. As such, we study two Byzantine attack scenarios that differ in the set of the compromised channels as illustrated in Figure 2.1.

Running Example 2.2.2 (Byzantine Attack) *Let agent 1 be communicating through a compromised channel at all iterations, i.e.,* $\mathcal{A}^{(k)} = \{1\}$*,* $\forall k$*. The compromised message sent to the central coordinator is* $\boldsymbol{b}_1^{(k)} = 1$*kW,* $\forall k$*. This means that irrespective of* $\boldsymbol{\theta}_1^{(k)}$*, the central coordinator receives a message indicating agent 1 is willing to charge at rate of* 1*kW.*

2.2.2.1 Attack scenarios

1. A *static impersonation attack*, where an adversary takes over a subset of uplink channels permanently and the set of agents communicating through compromised channels is fixed (i.e., $\mathcal{A}^{(k)} = \mathcal{A},\ \forall k$). Consequently, the central coordinator is never able to communicate reliably with agents $i \in \mathcal{A}$. In this case, it is not feasible to optimize the original problem (P) since the contribution from $f(\boldsymbol{\theta}_i) : i \in \mathcal{A}$ becomes unknown to the central coordinator. Yet, we assume that it is also not possible to deny access to resources to agents who are suspected of potentially being under attack. As a compromise, we formulate the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}_i \in \mathcal{C}_i, i \in \mathcal{H}} \quad & f(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i \in \mathcal{H}} f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j \in \mathcal{A}} g_t \left(\tfrac{1}{N} \textstyle\sum_{i=1}^{N} \boldsymbol{\theta}_i \right) \leq 0, \quad \forall t \in [T]. \end{aligned} \tag{2.9}$$

The objective of (2.9) is to minimize the cost of the agents with trustworthy channels subject to a *robust* set of constraints that consider the *worst case* scenario, in which the parameters of the agents with compromised channels are assumed to be maximizing the constraints (e.g., those agents are assumed to be consuming the maximum amount of resources). It is critical to mention that during a primal-dual algorithm scheme, the mes-

sages received through the compromised channels can be anything. The robust approach is to however ignore those messages, and assume that the parameters of the agents communicating through those channels are maximizing the constraints so that the operation of the system is feasible under any circumstance. Our goal is to develop an attack-resilient PD-DRA to solve the robust optimization problem (2.9).

Running Example 2.2.3 (Robust Optimization Model) *Since agent 1 is sending a compromised message of* $1kW$ *and their true parameter can be anything, the worst-case approach is to assume that they are charging at the maximum rate, which is* $7kW$ *for that agent. Hence, the robust constraint is:*

$$\max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j \in \mathcal{A}} g\left(\frac{1}{5}\sum_{i=1}^{5}\boldsymbol{\theta}_i\right) = \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j \in \mathcal{A}} \frac{1}{5}\sum_{i=1}^{5}\boldsymbol{\theta}_i - 5$$
$$= \frac{1}{5}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i + \max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j \in \mathcal{A}} \frac{1}{5}\sum_{j\in\mathcal{A}}\boldsymbol{\theta}_j - 5 = \frac{4}{5}\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3.6,$$

where we used $|\mathcal{H}| = 4$ *and the notation* $\overline{\boldsymbol{\theta}}_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i$*. The robust constraint states that:*

$$\frac{4}{5}\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3.6 \leq 0 \Rightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq 4.5.$$

The optimal solution in this case is to deliver electricity at a rate of $4.5kW$ *to the trustworthy agents. Since the compromised agent has the same cost function, their true charging rate will also be* $4.5kW$*, even though the message sent is* $1kW$ *and the central coordinator assumes their charging rate is* $7kW$*.*

2. A *dynamic impersonation attack*, where all the agents might be affected by the adversarial attacks but only for a *limited* fraction of time and hence, the set of agents communicating through compromised channels $\mathcal{A}^{(k)}$ has to dynamically change with iteration

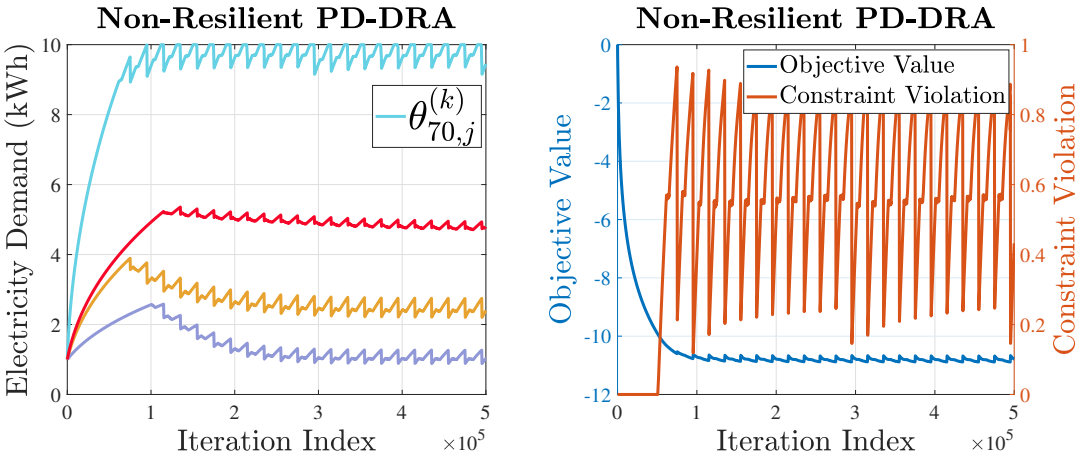


Figure 2.2: Illustration of basic PD-DRA algorithm failure under static impersonation attack. (a) The agents' parameters do not converge, (b) the objective function does not converge and moreover there is constraint violation. We only display one constraint for brevity.

k. As opposed to the static case, this scenario considers the case where the central coordinator is able to communicate reliably with all the agents at some iterations. Due to this distinction, it is necessary to mention that the static attack is not a special case of the dynamic attack and both scenarios are distinguishable from each other. The dynamic scenario could be applicable when agents do not have dedicated communication channels to the central coordinator and instead communicate over random access systems which are more appropriate for distributed deployments. Hence, each user periodically accesses authenticated network devices/subsystems that are controlled by Byzantine adversaries and can alter the user's message. Our goal is to develop an attack-resilient PD-DRA algorithm that can still solve the original regularized problem (P_υ) in this environment.

2.2.2.2 Limitations of the Basic PD-DRA Algorithm

Applying the basic PD-DRA algorithm under a Byzantine attack scenario can lead to undesirable outcomes. Recall that the gradients in (2.5) depend on the average parameter

$\overline{\boldsymbol{\theta}}^{(k)}$. Under a Byzantine attack scenario, if the central coordinator forms the naive average $\widetilde{\boldsymbol{\theta}}^{(k)} = (1/N)\sum_{i=1}^{N} \boldsymbol{r}_i^{(k)}$ and computes the gradients $\nabla g_t(\widetilde{\boldsymbol{\theta}}^{(k)})$ accordingly, this may result in large error since the deviation $\widetilde{\boldsymbol{\theta}}^{(k)} - (1/N)\sum_{i=1}^{N} \boldsymbol{\theta}_i^{(k)}$ can be large (proportional to the maximum diameter of $\mathcal{C}_i$'s). This in turn can obstruct convergence and also overload the system by causing constraint violations.

Running Example 2.2.4 (Basic PD-DRA Failure) *If the central coordinator believes all the agents are sending trustworthy information, then the optimal solution will occur when one agent is demanding* 1*kW and the others are demanding* 6*kW (so that the average is* 5*kW). But since the* 1*kW message is compromised and all the agents have same cost function, the compromised agent's true electricity demand is also at a rate of* 6*kW. Hence, the solution delivers electricity at an average rate of 6kW, which is infeasible.*

We preview our numerical result of applying the basic PD-DRA method under a static impersonation attack scenario for an optimal electric vehicle charging application in Figure 2.2. For constraint $g_t(\cdot)$, we define constraint violation as $\max\{0, g_t(\overline{\boldsymbol{\theta}}^{(k)})\}$. Observe that the PD-DRA method does not provide convergence and the first constraint is being violated. From resource allocation perspective, this means that the agents are asking to consume more resources than the available amount in the system, which is infeasible. For details regarding the experimental setup, please see Subsection 2.2.4.

2.2.3 Resilient PD-DRA Algorithms

Motivated by the failure of the basic PD-DRA procedure under Byzantine attack scenarios, resilient PD-DRA algorithms are necessary to optimize multi-agent systems in a distributed manner when the system is susceptible to attacks. We hold the following assumption to be true throughout the rest of this section and propose two different attack resilient PD-DRA algorithms corresponding to the different attack scenarios outlined in Subsecton 2.2.2.

Assumption 2.2.1 *For all $\boldsymbol{\theta} \in \mathbb{R}^d$ and for all t, the gradient of g_t is bounded with $\|\nabla g_t(\boldsymbol{\theta})\| \leq B$ and is L-Lipschitz continuous. Moreover, since maximum resource that can be consumed by an agent is bounded due to limited amount of resources, we let $\mathbf{0} \in \mathcal{C}_i$ and upper bound the diameters of $\mathcal{C}_i$ by R:*

$$\max_{\boldsymbol{\theta},\boldsymbol{\theta}' \in \mathcal{C}_i} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq R, \; i = 1, ..., N. \tag{2.10}$$

Running Example 2.2.5 (Assumptions) *The constraint in our running example satisfies that $\nabla g(\boldsymbol{\theta}) = 1$, which is bounded by $B = 1$ and is $L = 0$-Lipschitz continuous. Since the maximum charging rate is upper bounded by 7kW for three of the agents and by 10kW for two of the agents, $R = 10$.*

2.2.3.1 Static Impersonation Attack

Under this attack scenario, given the complete lack of any credible information on the resource consumption parameters of the agents that permanently communicate through compromised channels, the central coordinator can only hope to solve the robust optimization model defined in (2.9) instead. This formulation considers a worst-case scenario on how much resources the compromised agents will consume, which ensures constraint satisfaction in all cases. However, the constraints in (2.9) require the knowledge of the set $\mathcal{A}$ and the sets $\mathcal{C}_j, \forall j \in \mathcal{A}$, yet the central coordinator lacks this information.

Hence, in order to develop a robust optimization model that can handle the worst-case scenario without the knowledge of $\mathcal{A}$, we let $\alpha_1 \geq |\mathcal{A}|/N$ as a known upper bound to the fraction of agents communicating through compromised channels and assume $\alpha_1 < 1/2$, where less than half of the agents are communicating through compromised channels.[2] Let $\overline{\boldsymbol{\theta}}_{\mathcal{H}} := \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \boldsymbol{\theta}_i$ be the mean of the agent's parameters that are sent through trustworthy

[2] If more than half of the agents communicate through compromised channels, then the adversary controls the majority and therefore the median, which will be used to estimate the average parameter later in this section. In that case, there is no optimization based solution the central coordinator can implement in order to securely run the system.

channels. We then define the following set of constraints

$$\overline{g}_t(\boldsymbol{\theta}) := g_t(\boldsymbol{\theta}) + \alpha_1\big(RB + \tfrac{1}{2}LR^2\big), \tag{2.11}$$

and formulate a conservative approximation of (2.9):

Lemma 2.2.1 *Under Assumption 2.2.1 the following problem yields a conservative approximation of* (2.9)*, i.e., its feasible set is a subset of the feasible set of* (2.9)*:*

$$\begin{aligned} \min_{\boldsymbol{\theta}_i \in \mathcal{C}_i, i \in \mathcal{H}} \quad & \frac{1}{N}\sum_{i\in\mathcal{H}} f_i(\boldsymbol{\theta}_i) \\ \text{subject to} \quad & \overline{g}_t\left((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\right) \leq 0,\ \forall\, t \in [T], \end{aligned} \tag{2.12}$$

The proof can be found in Appendix A.1.1.

Remark 2.2.2 *The proof of Lemma 2.2.1 is done by upper bounding constraints of* (2.9) *using Assumption 2.2.1 and the fact that $\alpha_1 \geq |\mathcal{A}|/N$. The looser these upper bounds compared to the true values, the more conservative is* (2.12)*. This approach potentially leaves less resources available to the agents communicating through trustworthy channels by assuming more than $|\mathcal{A}|$ number of agents having maximum possible impact on the constraints, irrespective of their set $\mathcal{C}_i$ or the true value/gradient of the constraints.*

Running Example 2.2.6 (Conservative Approximation) *With $B = 1$, $L = 0$, and $R = 10$, the conservative approximation of the running example has the following constraint:*

$$\begin{aligned} \overline{g}((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}) &= g((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}) + \alpha_1(RB + \frac{1}{2}LR^2) \\ &= (1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 5 + 10\alpha_1 \end{aligned}$$

If $\alpha_1 = |\mathcal{A}|/N = 0.2$, then the upper bound is the fraction of compromised channels. In that

case, the constraint is:

$$0.8\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 3 \leq 0 \Leftrightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq 3.75,$$

which is more conservative compared to the constraint of the robust optimization model (which was $\overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq 4.5$*). The optimal solution in this case is to deliver electricity at a rate of* $3.75kW$ *to the agents. The conservatism arises due to the difference between agent-specific maximum charging rate* $7kW$ *and the absolute maximum charging rate* $10kW$*. Since the constraint is linear, the gradient is constant. Hence, the smoothness and Lipschitz bounds hold with equality without causing additional conservatism.*
If however $\alpha_1 = 0.4$*, then the central coordinator assumes two agents communicating through compromised channels. In this case the conservative approximation has the constraint as*

$$0.6\overline{\boldsymbol{\theta}}_{\mathcal{H}} - 1 \leq 0 \Leftrightarrow \overline{\boldsymbol{\theta}}_{\mathcal{H}} \leq \frac{5}{3},$$

which results in charging at an even slower rate since the central coordinator has to be robust against two agents charging at the maximum rate of $10kW$*.*

To develop an attack resilient PD-DRA algorithm, we again define the regularized Lagrangian function of (2.12):

$$\begin{aligned}\overline{\mathcal{L}}_{\upsilon}(\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}};\boldsymbol{\lambda};\mathcal{H}) :=& \tfrac{1}{N}\textstyle\sum_{i\in\mathcal{H}} f_i(\boldsymbol{\theta}_i) + \sum_{t=1}^{T}\lambda_t \overline{g}_t\left((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\right)\\ &+ \tfrac{\upsilon}{2N}\textstyle\sum_{i\in\mathcal{H}}\|\boldsymbol{\theta}_i\|^2 - \tfrac{\upsilon}{2}\|\boldsymbol{\lambda}\|^2.\end{aligned} \tag{2.13}$$

The above function is $(1-\alpha_1)\upsilon$-strongly convex and concave in $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, respectively (since $(1-\alpha_1) \leq \frac{|\mathcal{H}|}{N} \leq 1$). Our main task is to tackle the following modified problem of (P) under

Byzantine attack on (some of) the uplinks:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \min_{\boldsymbol{\theta}_i\in\mathcal{C}_i,\forall i\in\mathcal{H}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}};\boldsymbol{\lambda};\mathcal{H}). \tag{P$'_\upsilon$}$$

Notice that (P$'_\upsilon$) bears a similar form as (P) and thus one may apply the PD-DRA method to the former. The gradients with respect to primal/dual variables are given by:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_i}\overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}};\boldsymbol{\lambda}^{(k)};\mathcal{H}) = \frac{1}{N}\Big(\nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)},\\ + \tfrac{(1-\alpha_1)N}{|\mathcal{H}|}\textstyle\sum_{t=1}^T \lambda_t^{(k)}\nabla_{\boldsymbol{\theta}}\overline{g}_t(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=(1-\alpha_1)\bar{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}}\Big), \forall i\in\mathcal{H},\end{aligned} \tag{2.14a}$$

$$\big[\nabla_{\boldsymbol{\lambda}}\overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}};\boldsymbol{\lambda}^{(k)};\mathcal{H})\big]_t = \overline{g}_t\Big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\Big) - \upsilon\lambda_t^{(k)}. \tag{2.14b}$$

However, such application requires the central coordinator to compute the sample average

$$\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} = \tfrac{1}{|\mathcal{H}|}\textstyle\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i^{(k)}, \tag{2.15}$$

at each iteration. The above might not be computationally feasible under the attack model, since the central coordinator is oblivious to the identity of $\mathcal{H}$. As a solution, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k)}\}_{i\in[N]}$ using a median-based mean estimator described next.

Overview of Median-Based Mean Estimation

Consider a set of N vectors $\{\boldsymbol{x}_i\in\mathbb{R}^d\}_{i=1}^N$, among which at least $(1-\alpha_1)N$ are trustworthy ($\boldsymbol{x}_i\in\mathcal{H}$) and at most $\alpha_1 N$ are compromised ($\boldsymbol{x}_i\in\mathcal{A}$). We consider a simple median-based estimator applied to each coordinate $j=1,\ldots,d$. First, define the coordinate-wise median as:

$$[\boldsymbol{x}_{\mathsf{med}}]_j = \mathsf{med}\left(\{[\boldsymbol{x}_i]_j\}_{i=1}^N\right),$$

where $\mathsf{med}(\cdot)$ computes the median of the operand. Then, our estimator is computed as the mean of the nearest $(1-\alpha_1)N$ neighbors of $[\boldsymbol{x}_{\mathsf{med}}]_j$. Our estimator is:

$$[\widehat{\boldsymbol{x}}_{\mathcal{H}}]_j = \tfrac{1}{(1-\alpha_1)N} \textstyle\sum_{i\in\mathcal{N}_j} [\boldsymbol{x}_i]_j, \tag{2.16}$$

where we have defined the set with $|\mathcal{N}_j| = (1-\alpha_1)N$ as:

$$\mathcal{N}_j = \{i \in [N] : \left|\left[\boldsymbol{x}_i - \boldsymbol{x}_{\mathsf{med}}\right]_j\right| \leq r_j\},$$

such that r_j is chosen to satisfy $|\mathcal{N}_j| = (1-\alpha_1)N$.

The following bounds the performance of (2.16):

Proposition 2.2.2 *Let $\overline{\boldsymbol{x}}_{\mathcal{H}}$ be the mean of the trustworthy vectors. Suppose that*

$$\max_{i\in\mathcal{H}} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}\|_\infty \leq r$$

, then for any $\alpha_1 \in (0, \frac{1}{2})$, it holds that:

$$\|\widehat{\boldsymbol{x}}_{\mathcal{H}} - \overline{\boldsymbol{x}}_{\mathcal{H}}\| \leq \frac{2\alpha_1}{1-\alpha_1}\left(1 + \sqrt{\frac{(1-\alpha_1)^2}{1-2\alpha_1}}\right) r\sqrt{d}. \tag{2.17}$$

The proof can be found in Appendix A.1.2. We note that for sufficiently small α_1, the right hand side on (2.17) can be approximated by $\mathcal{O}(\alpha_1 r\sqrt{d})$. Using this median-based mean estimator, we propose the robust PD-DRA algorithm as follows.

Robust PD-DRA Algorithm

We summarize the static impersonation attack resilient PD-DRA method in Algorithm 2. The algorithm behaves similarly as Algorithm 1 applied to (P'_v), with the exception that the central coordinator is oblivious to $\mathcal{H}$, and it uses a robust mean estimator to find an approximate

Algorithm 2: Robust PD-DRA Algorithm

1: **Input**: Each agent has initial state $\boldsymbol{\theta}_i^{(0)}$.
2: **for** $k = 1, 2, ...$ **do**
3: *(At the Central Coordinator)*:

(a) Receives $\{\boldsymbol{r}_i^{(k)}\}_{i=1}^N$, see (2.8), from agents.

(b) Computes robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ using the estimator (2.16).

(c) Broadcasts the vector $\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} \overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)})$ to agents.

(d) Computes the update for $\boldsymbol{\lambda}^{(k+1)}$ with (2.18b).

4: *(At each agent $i \in \mathcal{H}$)*:

(a) Agent receives $\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)}$.

(b) Agent computes update for $\boldsymbol{\theta}_i^{(k+1)}$ with (2.18a).

5: **end for**

average for the signals sent through the trustworthy links, as illustrated in Figure 2.3. This approximate value is used to compute the new price signals, and sent back to agents. In particular, the primal-dual updates are:

$$\boldsymbol{\theta}_i^{(k+1)} = \mathcal{P}_{\mathcal{C}_i}\big(\boldsymbol{\theta}_i^{(k)} - \tfrac{\gamma}{N}\big(\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)} + \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon \boldsymbol{\theta}_i^{(k)}\big)\big), \tag{2.18a}$$

$$\lambda_t^{(k+1)} = \big[\lambda_t^{(k)} + \gamma\big(\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}) - \upsilon \lambda_t^{(k)}\big)\big]_+. \tag{2.18b}$$

We note that the update rule in (2.18a) is valid for agents in set $\mathcal{H}$, because the gradients of the Lagrangian are defined only for those agents in (2.14a). The agents in set $\mathcal{A}$ may or may not use the same update rule, however, this does not have any impact on the algorithm as they can never communicate their true parameters to the central coordinator.

Lemma 2.2.2 *Algorithm 2 is a primal-dual algorithm [24] for* (P'_υ) *with perturbed gradients:*

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} = \nabla_{\boldsymbol{\theta}} \overline{\mathcal{L}}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}; \mathcal{H}) + \boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}, \tag{2.19a}$$

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} = \nabla_{\boldsymbol{\lambda}} \overline{\mathcal{L}}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}; \mathcal{H}) + \boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}, \tag{2.19b}$$

where we have used concatenated variable as $\boldsymbol{\theta} = (\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}})$. *Under Assumption 2.2.1 and assuming that* $\lambda_t^{(k)} \leq \overline{\lambda}$ *for all* k, *we have:*

$$\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\| \leq (1-\alpha_1)\overline{\lambda} LT \|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\| + \frac{|\mathcal{H}| - (1-\alpha_1)N}{|\mathcal{H}|}\overline{\lambda} BT, \tag{2.20}$$

$$\|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\| \leq (1-\alpha_1) BT \|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\|. \tag{2.21}$$

The proof can be found in Appendix A.1.3. The assumption $\lambda_t^{(k)} \leq \overline{\lambda}$ can be guaranteed since $\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)})$ is bounded, which is proven in Appendix A.1.8. Furthermore, the performance analysis for the median based estimator shows that

$$\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\| = \mathcal{O}(\alpha_1 R\sqrt{d}) \tag{2.22}$$

when α_1 is small. Finally, based on Lemma 2.2.2, we can analyze the convergence of Algorithm 2. Let $\widehat{\boldsymbol{z}}^\star = (\widehat{\boldsymbol{\theta}}^\star, \widehat{\boldsymbol{\lambda}}^\star)$ be a saddle point of (P'_υ) and define

$$\overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(k)}) := \begin{pmatrix} \nabla_{\boldsymbol{\theta}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}, \boldsymbol{\lambda}^{(k)}; \mathcal{H}) \\ -\nabla_{\boldsymbol{\lambda}} \overline{\mathcal{L}}_\upsilon(\{\boldsymbol{\theta}_i^{(k)}\}_{i\in\mathcal{H}}, \boldsymbol{\lambda}^{(k)}; \mathcal{H}) \end{pmatrix}. \tag{2.23}$$

We are ready to present our main result for static attacks.

Theorem 2.2.1 *Assume the map* $\overline{\boldsymbol{\Phi}}(\boldsymbol{z}^{(k)})$ *is* L_Φ*-Lipschitz continuous. For Algorithm 2, for all*

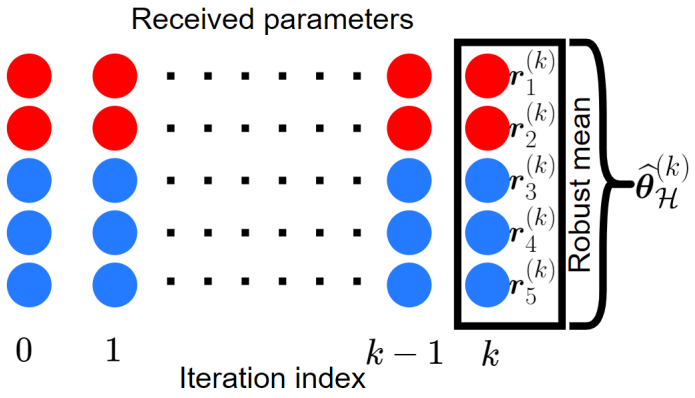


Figure 2.3: Robust mean estimation under static impersonation attack. Red/blue circles correspond to parameters received through compromised/trustworthy channels, respectively. In this example, there are $N = 5$ agents and agents 1 and 2 are always communicating through compromised channels. At iteration k, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k)}\}_{i\in[N]}$.

$k \geq 0$ *it holds:*

$$\|\boldsymbol{z}^{(k+1)} - \widehat{\boldsymbol{z}}^\star\|^2 \leq \big(1 - \gamma\upsilon' + 2\gamma^2 L_\Phi^2\big)\|\boldsymbol{z}^{(k)} - \widehat{\boldsymbol{z}}^\star\|^2 + \left(\frac{4\gamma}{\upsilon'} + 2\gamma^2\right) E_k. \tag{2.24}$$

where $\upsilon' := (1-\alpha_1)\upsilon$ *and* $E_k := \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 + \|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2$ *is the total perturbation at iteration* k*. Moreover, if we choose* $\gamma < \upsilon'/2L_\Phi^2$ *and* E_k *is upper bounded by* $\overline{E}$ *for all* k*, then*

$$\limsup_{k\to\infty} \|\boldsymbol{z}^{(k)} - \widehat{\boldsymbol{z}}^\star\|^2 \leq \frac{\frac{4}{\upsilon'} + 2\gamma}{\upsilon' - 2\gamma L_\Phi^2}\overline{E}. \tag{2.25}$$

The proof can be found in Appendix A.1.4. Combining with (2.22) shows that the resilient PD-DRA method converges to a $\mathcal{O}(\alpha_1^2 R^2 d)$ neighborhood of the saddle point of (P$'_\upsilon$). Moreover, it shows that the convergence rate to the neighborhood is linear, which is similar to the classical analysis in [24].

2.2.3.2 Dynamic Impersonation Attack

Under this attack scenario, the set of agents communicating through compromised channels is dynamically changing with iterations. We make the following assumption on how frequently each agent's communications are compromised:

Assumption 2.2.2 *Let m be a fixed window size and $\alpha_2 < 0.5$ be a known upper bound on how frequent an agent communicates through a compromised channel. Then, for all $k \geq m-1$ and for all agents $i \in [N]$, among the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ at most $\alpha_2 m$ are sent through compromised channels.*

It is important to recall that the dynamic attack scenario does not generalize the static attack scenario and there is a significant distinction between the two. The static attack scenario assumes that a fixed set of agents' communications are *permanently* compromised. It may occur when when the attacker compromises set of communication channels and those channels are assigned to the agents via a static channel allocation scheme.

On the contrary, for the dynamic attack scenario, each user's communications are vulnerable to attacks for *at most* a given α_2 fraction of iterations over a window of size m under Assumption 2.2.2, and hence each agent is able to communicate reliably with the central coordinator at some iterations. This scheme may occur when the attacker compromises a fixed set of communication channels (same as the static scenario), however, the channels are assigned to the agents via a dynamic channel allocation scheme (e.g., do a round-robin channel allocation. If there are m communication channels out of which $\alpha_2 m$ are compromised, assigning channels dynamically in a cyclic way to the agents ensures that over a window of m, every agent has sent $\alpha_2 m$ compromised messages). Although the attacker behaves the same way, we can simulate both scenarios by static/dynamic channel allocation. In cyber-physical systems, such dynamic allocation schemes are commonly used (e.g., dynamic IP assignment to be protected from hackers).

Interestingly, it is possible to develop an algorithm that converges to the optimal solution of Problem (P_v). The intuition behind is that the received parameters over a long period of time contain a fraction of trustworthy information that can extracted by the algorithm to perform faithful computations.

Our algorithm is similar in nature to an *averaging gradient* scheme where the primal-dual updates utilize the averages of time delayed gradients. Furthermore, the scheme is combined with the *robust mean* estimator developed in Sec. 2.2.3.1 to approximate the averages of outdated gradients, as illustrated in Figure 2.4. Specifically, the central coordinator chooses a window size of m. For any iteration $k \geq m-1$, instead of using $\boldsymbol{r}_i^{(k)}$ for computing the average parameter $\overline{\boldsymbol{\theta}}^{(k)}$ and the gradients, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ from the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ using the median-based mean estimator (2.16) for all agents $i \in [N]$, applied on the sequence of historical received parameters. Note that we have replaced α_1 by α_2, N by m in this application. It then uses $\widehat{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^{N} \widehat{\boldsymbol{\theta}}_i^{(k)}$ for computation of the primal-dual updates.

In the previous static impersonation attack scenario, the central coordinator is never able to communicate reliably with the agents with compromised channels, hence we have to compromise and make the problem robust by assuming the worst-case scenario. In this dynamic impersonation attack scenario, Assumption 2.2.2 ensures that out of any m consecutive parameters of an agent received by the central coordinator, at least $(1-\alpha_2)m$ of them will be sent through a trustworthy channel. Consequently, this scenario is easier to tackle since the central coordinator can make use of this history of information. In particular, if the central coordinator can *robustify* the received parameters over a window, the perturbation in the gradients behaves similar to that of incremental aggregate gradients (IAG) methods [52, 53, 54] and therefore guarantee convergence for sufficiently small step sizes. This is the main idea behind the design of our method.

We adopt an averaging scheme that aims to mitigate the effect of the parameters received

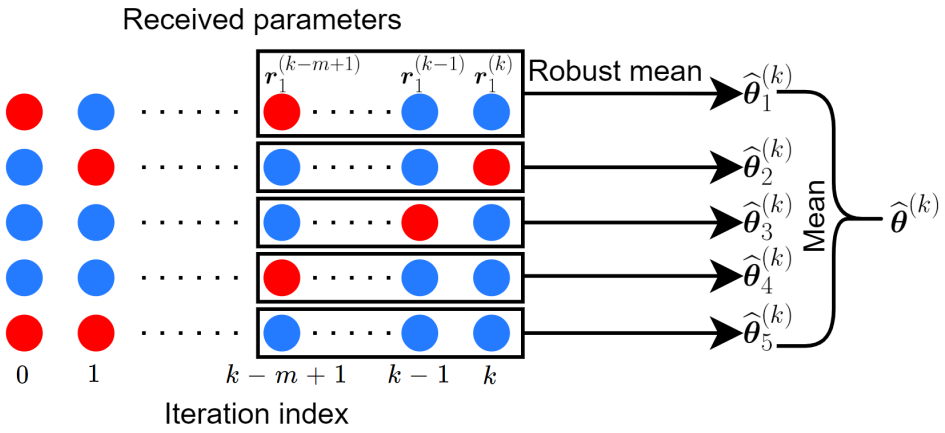


Figure 2.4: Robust mean estimation under dynamic impersonation attacks. Red/blue circles correspond to parameters received through compromised/trustworthy channels, respectively. In this example, there are $N = 5$ agents and the set of agents communicating through compromised channels is changing at every iteration. At iteration k, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ for all agents $i \in [N]$. Then, computes the naive average of $\{\widehat{\boldsymbol{\theta}}_i^{(k)}\}_{i=1}^{N}$ to get the average parameter $\widehat{\boldsymbol{\theta}}^{(k)}$.

through the compromised channels by considering the robust mean of the last m parameters. Specifically, the central coordinator chooses a window size of m. For all $k \geq m-1$, instead of using $\boldsymbol{r}_i^{(k)}$ for computing the average parameter $\overline{\boldsymbol{\theta}}^{(k)}$ and the gradients, the central coordinator computes the robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ of the received parameters $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ using the median-based mean estimator (2.16) for all agents $i \in [N]$ and uses $\widehat{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^{N} \widehat{\boldsymbol{\theta}}_i^{(k)}$ for computations. Figure 2.4 illustrates this scheme.

We summarize our robust averaging PD-DRA method in Algorithm 3. The primal-dual updates are described by:

$$\boldsymbol{\theta}_i^{(k+1)} = \mathcal{P}_{\mathcal{C}_i}\big(\boldsymbol{\theta}_i^{(k)} - \tfrac{\gamma}{N}\big(\widehat{\boldsymbol{g}}^{(k)} + \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}\big)\big), \tag{2.26a}$$

$$\lambda_t^{(k+1)} = \big[\lambda_t^{(k)} + \gamma\big(g_t(\widehat{\boldsymbol{\theta}}^{(k)}) - \upsilon\lambda_t^{(k)}\big)\big]_+. \tag{2.26b}$$

Lemma 2.2.3 *Algorithm 3 is a primal-dual algorithm for* (P_v) *with perturbed gradients:*

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_v(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}) + \boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}, \tag{2.27a}$$

$$\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} = \nabla_{\boldsymbol{\lambda}} \mathcal{L}_v(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)}) + \boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}, \tag{2.27b}$$

where we have used concatenated variable as $\boldsymbol{\theta} = (\{\boldsymbol{\theta}_i\}_{i\in N})$. *Under Assumption 2.2.1 and assuming that* $\lambda_t^{(k)} \leq \overline{\lambda}$ *for all k, we have:*

$$\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\| \leq \frac{\overline{\lambda} LT}{N} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|, \tag{2.28a}$$

$$\|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\| \leq \frac{BT}{N} \sum_{i=1}^{N} \|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|. \tag{2.28b}$$

The proof can be found in Appendix A.1.5. The assumption $\lambda_t^{(k)} \leq \overline{\lambda}$ can be guaranteed since $g_t(\widehat{\boldsymbol{\theta}}^{(k)})$ is bounded, which is proven in Appendix A.1.8. Let $\boldsymbol{z}^{(k)} := (\{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^N, \boldsymbol{\lambda}^{(k)})$ be the primal-dual variable at the kth iteration and define the mapping $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ as in (2.6). We observe that the algorithm's behavior is similar to the incremental aggregated gradient method in [52, 53, 54]. The following Lemma, which is inspired by [52, 53, 54], upper bounds the perturbation in the gradients in (2.28) by the maximum optimality gap in a finite window of size $2m-1$:

Lemma 2.2.4 *Assume the map* $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ *is* L_Φ*-Lipschitz continuous. Let* $E_k := \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 + \|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2$. *Then, for all* $k \geq 2(m-1)$ *we have:*

$$E_k \leq \gamma^2 \overline{C} \max_{0\leq \ell \leq 2(m-1)} \|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2, \tag{2.29}$$

Algorithm 3: Averaging PD-DRA Algorithm

1: **Input**: Each agent has initial state $\boldsymbol{\theta}_i^{(0)}$.
2: **for** $k = 0, 1, \ldots, m-2$ **do**
3: Apply basic PD-DRA (Run Algorithm 1).
4: **end for**
5: **for** $k = m-1, m, \ldots$ **do**
6: *(At the Central Coordinator)*:

(a) Receives $\{\boldsymbol{r}_i^{(k)}\}_{i=1}^N$, see (2.8), from agents.

(b) For all agents $i = 1, \ldots, N$, computes robust mean $\widehat{\boldsymbol{\theta}}_i^{(k)}$ of $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ using the estimator (2.16) with parameters $\alpha_1 \to \alpha_2$, $N \to m$.

(c) Computes $\widehat{\boldsymbol{\theta}}^{(k)} := \frac{1}{N}\sum_{i=1}^N \widehat{\boldsymbol{\theta}}_i^{(k)}$.

(d) Broadcasts the vector $\widehat{\boldsymbol{g}}^{(k)} := \sum_{t=1}^T \lambda_t^{(k)} \nabla_{\boldsymbol{\theta}} g_t(\widehat{\boldsymbol{\theta}}^{(k)})$ to agents.

(e) Computes the update for $\boldsymbol{\lambda}^{(k+1)}$ with (2.26b).

7: *(At each agent i)*:

(a) Agent receives $\widehat{\boldsymbol{g}}^{(k)}$.

(b) Agent computes update for $\boldsymbol{\theta}_i^{(k+1)}$ with (2.26a).

8: **end for**

where

$$\begin{aligned}\overline{C} = &\left(\frac{T^2(\overline{\lambda}^2 L^2 + B^2)}{N}\right) \times \left(\frac{1}{L_\Phi} + (1+\sqrt{d})\overline{\lambda} L T\right)^2 \\ &\times \left(\frac{1+C_\alpha}{1-\alpha_2} + C_\alpha\right)^2 \times (m-1)^2,\end{aligned} \tag{2.30}$$

and

$$C_\alpha = \frac{2\alpha_2}{1-\alpha_2}\left(1 + \sqrt{\frac{(1-\alpha_2)^2}{1-2\alpha_2}}\right)\sqrt{d}.$$

The proof can be found in Appendix A.1.6. Using on Lemmas 2.2.3 and 2.2.4, we can analyze the converge of Algorithm 3:

Theorem 2.2.2 *Assume the map $\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})$ is L_Φ-Lipschitz continuous. For Algorithm 3, for all*

$k \geq 2(m-1)$ *it holds that:*

$$\begin{aligned} \|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\|^2 &\leq (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \\ &+ \left(\frac{4\gamma}{\upsilon} + 2\gamma^2\right)\gamma^2\overline{C}\max_{0\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2. \end{aligned} \tag{2.31}$$

Moreover, if we choose γ sufficiently small such that it satisfies

$$\upsilon - 2\gamma L_\Phi^2 - \frac{4\overline{C}\gamma^2}{\upsilon} - 2\overline{C}\gamma^3 > 0,$$

then:

$$\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \leq \rho^{k-2(m-1)}\|\boldsymbol{z}^{(2(m-1))} - \boldsymbol{z}^\star\|^2, \tag{2.32}$$

and

$$\lim_{k\to\infty}\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 = 0, \tag{2.33}$$

where $\rho = (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2 + \frac{4\overline{C}\gamma^3}{\upsilon} + 2\overline{C}\gamma^4)^{\frac{1}{1+2(m-1)}}$.

The proof can be found in Appendix A.1.7. Theorem 2.2.2 shows that the robust averaging PD-DRA method converges *geometrically* to the optimal solution of (P_υ) under said assumptions.

2.2.3.3 Remarks

A few remarks highlighting design criteria to be explored in practical implementations are in order:

- Theorem 2.2.1 illustrates a trade-off in the choice of the step size γ between convergence speed and accuracy. In particular, (2.24) shows that the convergence rate factor $1-\gamma\upsilon+2\gamma^2 L_\Phi^2$ can be minimized by setting $\gamma = \upsilon/(4L_\Phi^2)$. Meanwhile, the asymptotic upper bound in (2.25) is increasing with γ and it can be minimized by setting $\gamma \to 0$.

- Theorem 2.2.2 illustrates a trade-off between the window size m and the convergence rate.

Observe that increasing the window size m decreases the rate of convergence by increasing ρ (Equations (2.30) and (2.32)). On the other hand, the likelihood that Assumption 2.2.2 holds true in a stochastic setting (e.g., channels being compromised with some probability) increases with a larger window size m.

- Under the dynamic impersonation attack scenario, the choice of α_2 does not affect convergence accuracy to the saddle point of (P_υ), but only changes the convergence rate. As such, choosing the largest α_2 such that $\alpha_2 m = \lfloor \frac{m-1}{2} \rfloor$ (i.e., assuming maximum possible number of iterates received through compromised channels) makes the algorithm robustly applicable to all dynamic impersonation attack scenarios regardless of the frequency of the attack.

- In case the central coordinator can not identify the attack scenario as static or dynamic impersonation (or the attack can be a mixture of both), a mixture of both Algorithms 2 and 3 can be applied. In particular, this can be done by adding Step 6(b) of Algorithm 3 before Step 3(b) of Algorithm 2, and applying the rest of the Algorithm 2 as it is. The central coordinator first computes robust parameters $\widehat{\boldsymbol{\theta}}_i^{(k)}$ by computing the robust mean of $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$ for all agents, and then computes the robust mean of $\{\widehat{\boldsymbol{\theta}}_i^{(k)}\}_{i=1}^{N}$. This effectively makes Algorithm 2 robust to possible dynamic impersonation attacks on uplink channels that are thought to be trustworthy for all iterations.

2.2.4 Numerical Study

In this subsection, we demonstrate the performance of our methods and verify our theoretical claims by applying our algorithms for: 1) an electric vehicle (EV) charging coordinator under static impersonation attack, 2) an electric vehicle charging coordinator under dynamic impersonation attack, and 3) a power distribution network with flexible demand under dynamic impersonation attack. The EV charging coordinator problem resembles classic network utility

maximization problems such as those studied in communication networks whereas the power distribution network problem has more nuisances that we will discuss next. To solve the convex optimization problems in order to get the optimal solutions, we used CVX, a package for specifying and solving convex programs[55].

2.2.4.1 Electric Vehicle (EV) Charging Facility

In this study, the aim is to optimize EV charging demand over time. We consider multiple EVs receiving charge under the same local feeder/transformer. Each agent (or EV owner) has different utility of charging at different times. Hence, at a given time period, it is desired to charge those EVs who have a higher utility (or less cost) for that time period. This problem falls into the broad category of network utility maximization problems, which can be formulated as:

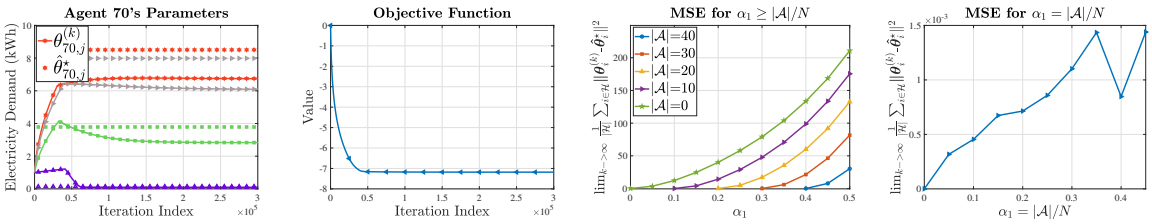
$$\min_{\boldsymbol{\theta}_i \in \mathbb{R}_+^d, \forall i} \quad f(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i) \tag{2.34a}$$

$$\text{subject to} \quad \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\theta}_i \preceq \overline{\boldsymbol{e}}, \tag{2.34b}$$

$$\boldsymbol{\theta}_i^{\min} \preceq \boldsymbol{\theta}_i \preceq \boldsymbol{\theta}_i^{\max}, \quad \forall i, \tag{2.34c}$$

$$\Theta_i^{\min} \leq \mathbf{1}^T \boldsymbol{\theta}_i \leq \Theta_i^{\max}, \quad \forall i, \tag{2.34d}$$

where $N \times \overline{\boldsymbol{e}} \in \mathbb{R}^d$ is the vector of maximum available transformer capacity in all time periods and $\preceq$ denotes component-wise inequality between the vectors. The available capacity changes with time of day as exogenous load on the transformer varies with time as well. The elements $\{\boldsymbol{\theta}_{i,j}\}_{j=1}^d$ of the vector $\boldsymbol{\theta}_i$ correspond to the electricity demand of the EV i at time slots $j = 1 \dots d$. The constraint (2.34c) restricts the amount an EV can charge at each time slot, whereas the constraint (2.34d) bounds the total amount an EV can charge. For this study,



(a) (b) (c) (d)

Figure 2.5: Numerical study results for optimal electric vehicle charging under static impersonation attack. (a) Optimal parameter of agent 70 converges to a neighborhood of the optimal solution of the robust optimization problem for $|\mathcal{A}|/N = 0.2$ and $\alpha_1 = 0.3$, (b) The algorithm provides convergence of the objective function value, (c) Mean squared error for different number of compromised channels and different choices of upper bound α_1, (d) Mean squared error when $\alpha_1 = |\mathcal{A}|/N$.

we set the cost function to be:

$$f_i(\boldsymbol{\theta}) = -\sum_{j=1}^{d} \beta_{i,j} \log \boldsymbol{\theta}_{i,j}, \tag{2.35}$$

where $\beta_{i,j}$ are generated randomly from a uniform distribution in $[0, 1]$. We study this problem under both attack scenarios for $N = 100$ EVs.

Static impersonation attack

We simulated various static impersonation attack scenarios and ran Algorithm 2. The results are displayed in Figure 2.5.

In Figure 2.5a, we plot agent 70's electricity demand for some time periods, with $|\mathcal{A}|/N = 0.2$ and $\alpha_1 = 0.3$. Each different color corresponds to a different dimension of the parameter vector (i.e., electricity demand for different time periods). A colored solid line corresponds to a dimension of the parameter vector iterates generated by the algorithm. A dashed line with the same marker and color as a solid line is the optimal value corresponding to that dimension of the parameter vector, which is the solution of the regularized robust optimization problem (formulated as (P$'_v$)) of (2.34). Observe that Algorithm 2 successfully

provides convergence to a close neighborhood of the optimal solution of the regularized robust optimization problem. Furthermore, in Figure 2.5b we show that the objective function value converges, as opposed to a non-resilient PD-DRA method that is shown to oscillate and violate the constraint in Figure 2.2. Our robust optimization model on the other hand ensures there is no constraint violation.

In Figure 2.5c, we plot the mean squared error (MSE) in primal variables $\boldsymbol{\theta}_i$ for different number of compromised channels $|\mathcal{A}|$ and different choices of α_1, which is the upper bound on fraction of compromised links known by the central coordinator. The MSE is calculated by:

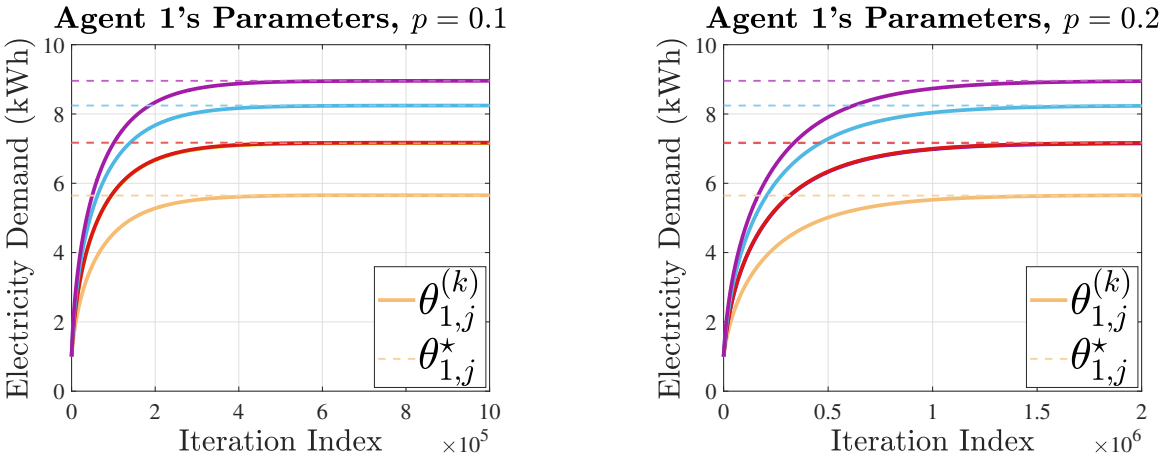
$$\text{MSE} = \lim_{k\to\infty} \frac{1}{|\mathcal{H}|} \sum_{i\in\mathcal{H}} \|\boldsymbol{\theta}_i^{(k)} - \hat{\boldsymbol{\theta}}_i^{\star}\|^2, \tag{2.36}$$

where $\hat{\boldsymbol{\theta}}_i^{\star}$ is the solution to (P$'_v$) with $\alpha_1 = |\mathcal{A}|/N$, i.e., the solution to the regularized and robustified problem with the knowledge of the compromised channels. Naturally, the looser the upper bound, the larger the error, since it increases the amount of conservatism. Hence, having an accurate upper bound on fraction of compromised channels significantly improves the performance.

Finally, in Figure 2.5d we exhibit the efficacy of our approach with median-based mean estimation. We plot the mean squared error in primal variables, when the upper bound on α_1 is tight, i.e., $\alpha_1 = |\mathcal{A}|/N$. The error tends to increase with $|\mathcal{A}|/N$, however, considering the magnitude, the error is negligible and we can conclude that the median-based mean estimator performs well.

Dynamic impersonation attack

We simulated a dynamic impersonation attack scenario and ran Algorithm 3. To simulate a dynamic impersonation attack, we assigned a probability p for an uplink to be compromised



(a) $p = 0.1$, $m = 20$, $\alpha_2 = 0.45$ (b) $p = 0.2$, $m = 100$, $\alpha_2 = 0.49$

Figure 2.6: Numerical study results demonstrating convergence of Algorithm 3 for optimal electric vehicle charging under two dynamic impersonation attack scenarios: (a) $p = 0.1$, (b) $p = 0.2$. Observe that the number of iterations it takes to converge for (b) is much larger than for (a).

at each iteration[3]. For $p = 0.1$, we picked a window size $m = 20$ and $\alpha_2 = 0.45$, whereas for $p = 0.2$, we picked a window size $m = 100$ and $\alpha_2 = 0.49$. The results are displayed in Figure 2.6. Each different color corresponds to a different dimension of the parameter vector. A colored solid line corresponds to a dimension of the parameter vector iterates generated by the algorithm. A dashed line with the same color as a solid line is the optimal value corresponding to that dimension of the parameter vector, which is the solution of the regularized optimization problem (formulated as (P_υ)) of (2.34).

In both scenarios, Algorithm 3 successfully provides convergence to the optimal solution of the regularized problem. Observe that for $p = 0.2$, we chose a larger window size and a larger α_2 in order to meet Assumption 2.2.2. However, this restricts us to choose a smaller step size γ as dictated by Theorem 2.2.2 and in turn slower convergence. This highlights an important trade-off between robustness and convergence rate, where a larger window size m

[3]Although a probabilistic scenario does not guarantee that Assumption 2.2.2 holds, with sufficiently large window size m and α_2, it holds with high probability at each iteration. Even though we do not study this scenario theoretically, our algorithm still performs well.

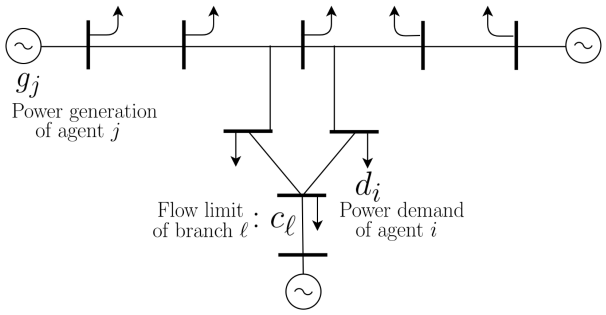


Figure 2.7: IEEE 9 bus system with 3 generators (supplies) represented by sources and 8 loads (demands) represented by arrows.

and larger α_2 makes the algorithm more robust while decreasing the convergence rate.

2.2.4.2 Power Distribution Network

We consider the IEEE $N = 9$ bus system with $N_g = 3$ generators and $N_\ell = 8$ loads as shown in Figure 2.7. The power network cost minimization problem can be stated as:

$$\min_{d_i, g_i \in \mathbb{R}^+} \quad f(\boldsymbol{d}, \boldsymbol{g}) = -\sum_{i=1}^{N_\ell} U_i(d_i) + \sum_{i=1}^{N_g} C_i(g_i) \tag{2.37a}$$

$$\text{subject to} \quad \mathbf{1}^T(\boldsymbol{d} - \boldsymbol{g}) = 0, \tag{2.37b}$$

$$\mathbf{H}(\boldsymbol{d} - \boldsymbol{g}) \leq \boldsymbol{c}, \tag{2.37c}$$

where $\boldsymbol{d} = [d_1 \dots d_N]^T$ and $\boldsymbol{g} = [g_1 \dots g_N]^T$ are the vectors of load and generation at each node, respectively ($d_i = 0$ for nodes without load and $g_j = 0$ for nodes without generators). The first constraint (2.37b) ensures the power supply is equal to the demand, and the second constraint (2.37c) is the power flow constraint limiting the power flow on each branch.

Observe that the formulation in (2.37) does not directly match with our general formulation in (2.1) mainly due to the presence of equality constraint (2.37b), which prevents the application of the robustified formulation in (2.9) and hence the robust PD-DRA algorithm for static impersonation attacks. Nevertheless, our algorithm for dynamic impersonation attacks

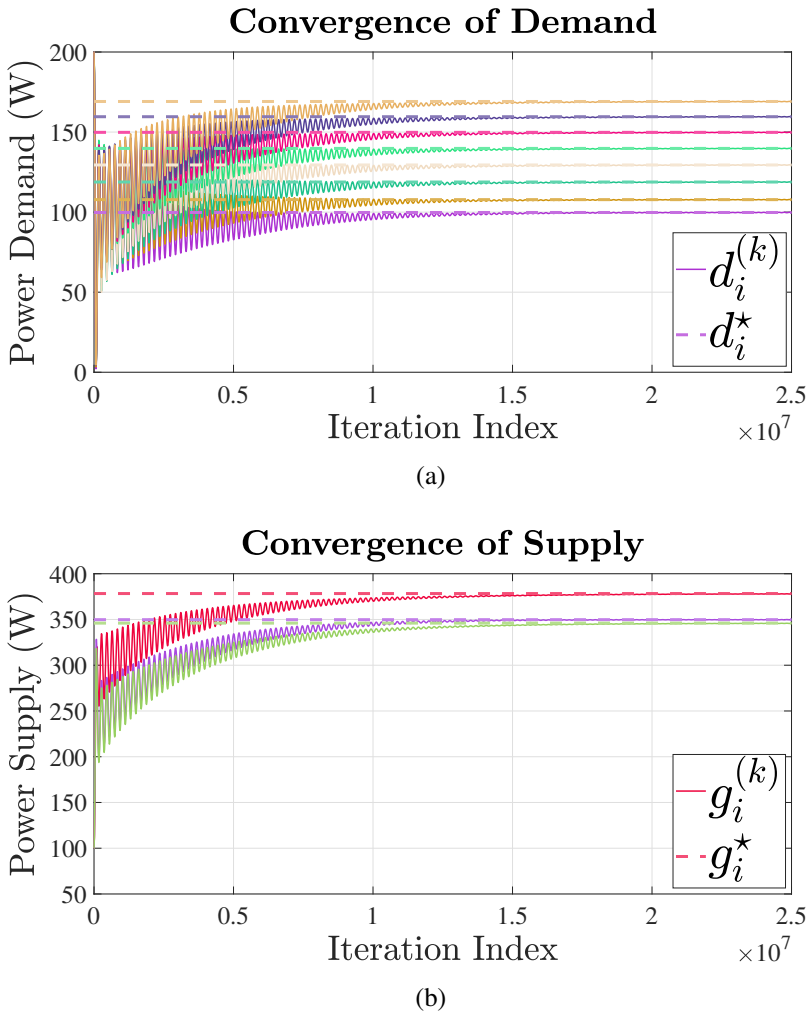


Figure 2.8: Numerical study results for power network under dynamic impersonation attack: (a)/(b) displaying convergence of the demand/supply, respectively.

can still be applied since it does not require any robustified constraints (which cannot be done for equality constraints).

We have chosen the utility function for load i to be $U_i(d_i) = \beta_i \log d_i$ and randomly generated β_i from a uniform distribution in $[500, 1000]$. For generators, we set the cost function $C_i(g_i) = e^{c_i g_i}$, where $c_1 = 0.01,\ c_2 = 0.011,\ c_3 = 0.012$. We obtained the Power Transfer Distribution Factor (PTDF) matrix $\mathbf{H}$ and the vector of flow limits $\boldsymbol{c}$ from MATPOWER[56]. To simulate a dynamic impersonation attack scenario, we assigned a probability p for an uplink to

be compromised at each iteration. We ran Algorithm 3 for $p = 0.15$, $m = 75$ and $\alpha_2 = 0.49$.

The results are shown in Figure 2.8. In both Figures 2.8a and 2.8b, each different color corresponds to a different agent. A colored solid line corresponds to an agent's parameter iterates generated by the algorithm. A dashed line with the same color as a solid line is the to the optimal value of that agent's parameter, which is the solution of the regularized optimization problem (formulated as (P_v)) of (2.37). Our algorithm successfully generates sequences that convergence to the optimal solution of the regularized problem for both power supplying and power demanding agents.

2.2.5 Conclusion

In this section, we studied two strategies for establishing primal-dual algorithms for resource allocation in presence of Byzantine attackers. Specifically, we consider static and dynamic impersonation attack scenarios and propose an attack-resilient primal-dual algorithm for each scenario based on robust mean estimation techniques. We derive bounds for the performance (in terms of distance to optimality) of the proposed algorithms and show that our algorithm for static impersonation attack converges to a neighborhood of the optimal solution of the regularized and robustified resource allocation problem, whereas our algorithm for dynamic impersonation attack converges to the optimal solution of the original regularized problem. We verify our theoretical results via computational simulations for network utility maximization problems involving optimal distributed resource allocation, such as power distribution networks.

2.3 Robust Distributed Optimization With Randomly Corrupted Gradients

In this section, we propose a first-order distributed optimization algorithm that is provably robust to Byzantine failures—arbitrary and potentially adversarial behavior, where all the participating agents are prone to failure. Robust distributed optimization under adversarial manipulation has been studied for various corruption models, see [57, 58] for comprehensive reviews. For example, gradients communicated over a network are usually modeled as corrupted by: non-malicious noise [59], adversarial noise [60], quantization [61], or because the gradients are inexact oracles [62]. Although robust optimization methods with strong theoretical guarantees are well established [63, 60], a drawback of these approaches is that the corrupt gradients are assumed to be within a bounded neighborhood of the trustworthy ones, i.e., corruption can be modeled as a bounded additive noise to the trustworthy gradients.

On the other hand, an adversarial corruption model, which can be unbounded and arbitrary, has been extensively studied in the distributed learning literature under categories of data poisoning [64] and model update poisoning attacks [65, 66]. This line of work models corruption as an arbitrary manipulation on the information sent by the agents or on the data samples stored at the agents. However, the adversary is often assumed to have limited capability, i.e., the adversary is only able to manipulate a certain fraction of agents or data samples. Although successful defense mechanisms based on robust aggregation methods [67, 68, 69, 70, 71, 41, 72] and data sanitation using robust statistics [64] are shown to be robust to these types of manipulation, robust estimation techniques rely on a bounded α fraction of agents/data points being corrupt at all times. Therefore, they are not applicable if there exist iterations with more than α fraction of corrupted agents. For instance, if at any iteration more than half of the agents behave unpredictably and send arbitrarily corrupt information, then the aggregate will be arbitrarily corrupted. In fact, it was recently shown that even more benign-looking manipulations

are able to get through these defense mechanisms with corruption rates as low as $.5 - 1\%$ [73].

In this section, we study another corruption model where existing defense mechanisms are prone to failure. In particular, we adopt a distributed optimization framework where a group of agents communicates local gradient information to a central machine that aggregates and distributes information back to the agents. By modeling the temporal dynamics of the agents' states (either trustworthy or corrupted/Byzantine) via a two-state Markov chain, we allow *all the agents* to be susceptible to *arbitrary corruption*. This type of corruption would occur in practical applications of distributed optimization due to various reasons including but not limited to:

1. Behavioral (intentional or unintentional) changes of the agents: Due to its privacy-preserving nature, the models established for practical applications such as text completion are trained using user text data without observing it. Besides unintentional mistakes that can be made by a user at random times, users can intentionally behave differently at different time periods. For instance, a multilingual person who works in the United States could be typing in English during work hours and in another language after work hours. These periods can also be longer or shorter in duration. If the goal is to train a text completion model for English, then we consider the user as Byzantine when they type in another language and trustworthy when they type in English. In this setting, a Markovian Byzantine agent model would be a suitable corruption model.

2. Cyber attacks in cyber-physical systems: Among the many types of cyber attacks, Byzantine attacks and man-in-the-middle attacks are important to defend against for distributed optimization algorithms. The attacker is free to hack any user at any time, however, the hack is not necessarily successful all the time, for instance, due to the existence of a firewall. In this case, it would be suitable to model the dynamics of a trustworthy agent's state as a Markov chain with a certain probability of turning into Byzantine, which would

capture the aforementioned random characteristics of a Byzantine attack. On the other hand, literature on Byzantine fault detection and man-in-the-middle attack detection establishes that with repeated interactions with the agents, these types of attacks can be detected [74, 75]. However, there are no certain guarantees on how long a successful detection would take as it would depend on how the attacker behaves. To resemble this randomness in the detection time and success, it would be suitable to model the dynamics of a Byzantine agent's state as a Markov chain with a certain probability of turning into trustworthy. Given these features of cyber attacks and defense on cyber-physical systems, it would be suitable to approximate the agent behavior by a Markovian model as opposed to a static model for distributed optimization applications.

A consequence of the Markovian setting is that there could exist iterations at which the majority of the agents send corrupt gradients to the central machine, in which case existing defense mechanisms would fail. For this setting, we develop a robust distributed optimization algorithm with provable convergence guarantees for a number of function classes.

Contributions: Our main contribution is a distributed stochastic optimization algorithm, named Robust Aggregating Normalized Gradient Method (RANGE), that achieves strong convergence guarantees while being robust to a newly proposed Markovian gradient corruption model.

- We propose a novel Markovian Byzantine agent model that models dynamically changing sets of Byzantine agents with no assumptions on the maximum fraction of Byzantine agents at a particular iteration.

- We study two settings for stochastic optimization for RANGE, namely Sample Average Approximation (SAA) and Stochastic Approximation (SA). We prove that for both SAA and SA, when the parameters are tuned appropriately according to the spectral gap of the Markov chain, RANGE converges to a neighborhood of the optimal solution at a linear

rate for strongly convex cost functions.

- We prove that for smooth (possibly non-convex) cost functions, RANGE converges to a neighborhood of a stationary point at a rate of $\mathcal{O}(1/\sqrt{T})$, where T is the number of iterations.

- For the SAA setting, we show that RANGE achieves lower error rates in the Markovian Byzantine agent setup with an expected α fraction of Byzantine agents than state-of-the-art algorithms in the setup with a bounded α fraction of Byzantine agents for all iterations.

- We show that RANGE achieves lower statistical error rates in the SA setting than the SAA setting for sufficiently low corruption rates, i.e., the expected fraction of Byzantine agents. We provide an explicit characterization of such bound.

- We provide numerical evidence demonstrating the efficacy and robustness of RANGE in the proposed setting.

RANGE is designed with three ingredients: (1) temporal robust aggregation, (2) spatial robust aggregation, and (3) gradient normalization. The temporal robust aggregation step estimates the robust mean of each agent's historical gradient data over a finite window to compute a robustified gradient for each agent. Informally, the received gradients over a period of time contain a fraction of trustworthy information that can be extracted by the algorithm to perform faithful computations rather than applying potentially corrupt gradients directly. In case the robustified gradient produced by temporal robust aggregation becomes contaminated by corruption, another layer of defense mechanism is implemented via spatial robust aggregation of all the agents' robustified gradients. Lastly, normalization preserves only the directional information and thus prevents large updates that corrupt gradients might cause in case the temporal

robust aggregation and spatial robust aggregation steps do not sufficiently eliminate corruptions.

Related work: The work presented in this section has connections to the literature on (i) normalized gradient method, (ii) gradient clipping, and (iii) delayed gradient descent.

• *Normalized Gradient Method:* Normalized gradient method is a well-studied algorithm for optimization and is supported by theoretical convergence guarantees for convex [76] and quasi-convex optimization [77]. Using normalized updates is gaining popularity, especially for non-convex optimization [78], since for non-convex objectives, unlike the convex ones, the magnitude of the gradient provides less information about the value of the function, while the direction still indicates the direction of steepest descent. An important benefit of this is the fast evasion of saddle points [79]. Seeing the need for large batch sizes for variance reduction of stochastic gradients as a drawback of normalized updates, a recent work [80] proves that adding momentum removes the need for large batch sizes on non-convex objectives while matching the best-known convergence rates. In a preliminary conference report [11], we investigated the robustness properties of the normalized subgradient method for solving deterministic optimization problems in a centralized fashion. In the current section, we expand [11] into a distributed setup with a stochastic objective function, additionally study non-convex objectives both theoretically and numerically, and employ two additional layers of defense by means of robust mean estimation before applying normalization to improve our algorithm.

• *Gradient Clipping:* As a similar method to normalization, gradient clipping is a common technique in optimization used for privacy [81]. Recent studies demonstrate that gradient clipping can be applied for robustness to model update poisoning attacks [82] and label noise [83]. However, similar to robust distributed optimization literature, due to the limitations on the amount of corruption and adversarial agents, their methods are inapplicable in our setting and can be outperformed, as we will show numerically in Subsection 2.3.6.

• *Delayed Gradient Descent:* Temporal robust aggregation step of our method is in principle similar to a delayed gradient descent method [84], since temporal aggregation is a linear combination of the past gradients. Motivated by applications to distributed optimization over networks, researchers have established convergence guarantees for deterministic [85] and stochastic delayed gradient methods [86]. Given strong theoretical results, we integrate the delayed gradient method to our algorithm via temporal robust aggregation and show that it improves robustness.

Organization: The remainder of this section is organized as follows. In Subsection 2.3.1, we define the problem setting. In Subsection 2.3.2, we describe our algorithm called RANGE and discuss how it can solve the proposed problem. In Subsections 2.3.3 and 2.3.4, we present the convergence properties of RANGE for the SAA and SA settings, respectively. In Subsection 2.3.5, we discuss two special cases of RANGE, one without temporal robust aggregation and one with independent random corruption. In Subsection 2.3.6, we provide numerical results for RANGE.

Notations and conventions: Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm. For any $N \in \mathbb{N}$, $[N]$ denotes the finite set $\{1, ..., N\}$. Given a vector v, if $\|v\| = 0$, then $v/\|v\| = 0$. The $\mathcal{O}(\cdot)$ notation hides constants, logarithmic terms, and only includes the dominant terms. Given a function $f(x, z)$, $\partial_k f(x, z)$ denotes the partial derivative of $f(x, z)$ with respect to k'th coordinate of x.

2.3.1 Problem Setup

In this subsection, we formally set up our problem and introduce key concepts and definitions that will be used in this section. We are interested in the stochastic optimization problem

$$x^\star = \arg\min_{x \in \mathcal{X}} F(x) = \arg\min_{x \in \mathcal{X}} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[f(x, z)], \tag{2.38}$$

where $f(x, z)$ is a cost function of a parameter vector $x \in \mathcal{X} \subseteq \mathbb{R}^d$ associated with a data point $z \in \mathcal{Z}$ and the data points are sampled from some unknown distribution $\mathcal{D}$. To solve (2.38), we study two settings for stochastic optimization, namely Sample Average Approximation (SAA) [87] and Stochastic Approximation (SA) [88], in a distributed setup with one central machine and N agents that compute stochastic gradients at a point x via $\nabla f(x, z)$ based on independent samples $z \sim \mathcal{D}$.

In iterative distributed first-order methods, given the parameter vector x_t at iteration t, the central machine receives the feedback $\nabla F_{i,t}(x_t)$ from all the agents, aggregates by computing the average, and applies a descent step to get the updated parameter x_{t+1}. Here,

$$F_{i,t}(x_t) = \frac{1}{b} \sum_{j=1}^{b} f(x_t, z_{i,t}^j) \tag{2.39}$$

is the empirical risk function and $\{z_{i,t}^j\}_{j \in [b]}$ are the b data points used for gradient computation at agent i and iteration t. In SAA, each agent uses a fixed set of data samples to estimate the gradient at all iterations, i.e., $\{z_{i,\tau}^j\}_{j \in [b]} = \{z_{i,\tau'}^j\}_{j \in [b]}$ and $F_{i,\tau}(x) = F_{i,\tau'}(x)$ $\forall i \in [N], x \in \mathcal{X}, \tau, \tau' \in \mathbb{N}_0$ [89]. In SA, the agents sample b new data points from $\mathcal{D}$ at each iteration and therefore $F_{i,\tau}(x)$ and $F_{i,\tau'}(x)$ are independent random variables $\forall i \in [N]$, $x \in \mathcal{X}$, $\tau, \tau' \in \mathbb{N}_0$ such that $\tau \neq \tau'$ [89].

Such methods, however, rely on the feedback received from each agent being trustworthy gradient information and might fail to converge when the feedback becomes corrupt, as one single corrupt feedback can have an arbitrarily large effect. Denote the set of agents communicating corrupt gradient information, i.e., Byzantine agents, at iteration t by $\mathcal{B}^t$, and the set of agents communicating trustworthy gradient information, i.e., trustworthy agents, at iteration t

by $\mathcal{T}^t$. At each iteration t, the feedback is determined as:

$$g_{i,t} = \begin{cases} \nabla F_{i,t}(x_t) & \text{if } i \in \mathcal{T}^t, \\ \star & \text{if } i \in \mathcal{B}^t, \end{cases} \tag{2.40}$$

where the corrupt feedback $\star$ is arbitrary and is possibly chosen by an adversary, who may have full knowledge of the problem. We note that this model encompasses a large class of scenarios where the feedback can become corrupt (e.g., errors in communication or computation, corrupt data, adversarial manipulation) since we set no restrictions on $\star$.

Contrary to existing literature, we study dynamically changing sets of Byzantine agents $\mathcal{B}^t$ and trustworthy agents $\mathcal{T}^t$, where the transition of each agent from Byzantine/trustworthy state to trustworthy/Byzantine state happens probabilistically at each iteration. In particular, we define

$$p_b = \mathbb{P}(i \in \mathcal{B}^{t+1} | i \in \mathcal{T}^t), \quad \forall i \in [N], \forall t, \tag{2.41}$$

$$p_t = \mathbb{P}(i \in \mathcal{T}^{t+1} | i \in \mathcal{B}^t), \quad \forall i \in [N], \forall t, \tag{2.42}$$

where $0 < p_b < p_t < 1/2$. Accordingly, each agent's state transition over time is governed by a two-state Markov chain with transition matrix

$$M = \begin{bmatrix} 1 - p_b & p_b \\ p_t & 1 - p_t \end{bmatrix}, \tag{2.43}$$

and stationary distribution

$$\pi^\star = \begin{bmatrix} \dfrac{p_t}{p_t + p_b} & \dfrac{p_b}{p_t + p_b} \end{bmatrix}, \tag{2.44}$$

where state 0 corresponds to the trustworthy state and state 1 corresponds to the Byzantine state. We note that the exact knowledge of the transition probabilities is not necessary. We can

take p_b as an upper bound on the trustworthy to Byzantine transition probability, and p_t as a lower bound on the Byzantine to trustworthy transition probability.

In the next subsection, we explain the first-order method we propose to obtain a near-optimal solution to (2.38) in setting defined by (2.40)-(2.42). For completeness, we end this subsection with a couple of standard definitions from convex analysis regarding a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$.

Definition 2.3.1 *A differentiable function f is said to be **L-smooth** if there exists $L > 0$ such that*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L\|x_1 - x_2\|, \tag{2.45}$$

for all $x_1, x_2 \in \mathcal{X}$.

Definition 2.3.2 *A differentiable function f is said to be **μ-strongly convex** if there exists $\mu > 0$ such that*

$$\langle \nabla f(x_1) - \nabla f(x_2), x_1 - x_2 \rangle \geq \mu\|x_1 - x_2\|^2, \tag{2.46}$$

for all $x_1, x_2 \in \mathcal{X}$.

2.3.2 Robust Aggregating Normalized Gradient Method (RANGE)

To solve Problem (2.38) in the Byzantine setting defined by (2.40)-(2.42), we propose an algorithm called Robust Aggregating Normalized Gradient MEthod (RANGE), which is summarized in Algorithm 4. There are three main interacting ideas behind Algorithm 4 to guarantee convergence and robustness: 1) temporal robust aggregation, 2) spatial robust aggregation, and 3) gradient normalization. Temporal robust aggregation in Step 8 of Algorithm 4 aims to compute a robustified gradient for all agents by estimating a robust mean of a window of past gradients. The intuition behind this is that despite corruptions, the feedback received over a long period from every single agent contains a fraction of trustworthy information that

Algorithm 4: Robust Aggregating Normalized Gradient Method (RANGE)

1: **Input:** Initialize $x_1 \in \mathcal{X}$, step size γ, window size m, $m_0 \in \mathbb{N}_0$, T, and $\alpha_1, \alpha_2 < 0.5$ s.t. $\alpha_1 m, \alpha_2 N \in \mathbb{N}_0$.
2: **for** $t = 1$ **to** $T + m - 1 + m_0$ **do**
3: Broadcast x_t to the agents.
4: Receive $g_{i,t}$, defined in (2.40), for $i \in [N]$.
5: **if** $t \leq m - 1$ **then**
6: Set $\hat{g}_{i,t} = g_{i,t}$.
7: **else**
8: Compute the robust mean $\hat{g}_{i,t}$ of $\{g_{i,t-\tau}\}_{\tau=0}^{m-1}$ using (2.47) with parameters α_1 and m, for $i \in [N]$.
9: **end if**
10: Compute the robust mean $\hat{\hat{g}}_t$ of $\{\hat{g}_{i,t}\}_{i\in[N]}$ using (2.47) with parameters α_2 and N.
11: Compute $x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \gamma \hat{\hat{g}}_t / \|\hat{\hat{g}}_t\|\right)$.
12: **end for**

the algorithm can extract to perform accurate computations. To defend against the scenarios where the robustified gradient produced by the temporal robust aggregation step is corrupted for some of the agents (e.g., if the window only contains corrupted gradients), in Step 10 of Algorithm 4 we implement a second layer of robust mean estimation when aggregating all the agents' robustified gradients in order to eliminate those corrupted gradients. Lastly, by gradient normalization in Step 11 of Algortihm 4, we restrict the aggregate gradient to only contain directional information. This prevents arbitrarily large updates in case the temporal robust aggregation and spatial robust aggregation steps do not sufficiently eliminate the corruptions.

Let us provide a summary of Algorithm 4. At each iteration t, the central node receives the feedback $g_{i,t}$ according to (2.40) from all the agents. If $t \geq m$, it estimates a robustified gradient $\hat{g}_{i,t}$ for each agent $i \in [N]$ by performing a temporal robust aggregation over a window of gradients $\{g_{i,t-\tau}\}_{\tau=0}^{m-1}$ using the median-based mean estimator that will be described later. If $t < m$, it simply sets $\hat{g}_{i,t} = g_{i,t}$. Then, it aggregates the robustified gradients $\{\hat{g}_{i,t}\}_{i\in[N]}$ using the same median-based mean estimator to get the robust aggregate $\hat{\hat{g}}_t$ and moves the iterate along $\hat{\hat{g}}_t / \|\hat{\hat{g}}_t\|$ with step size γ. Finally the algorithm projects the point back to the decision set

$\mathcal{X}$.

To get a good grasp of why RANGE works, let us discuss how the mechanics of each step assist the convergence of the algorithm, starting with the robust mean estimator.

Robust mean estimator: Suppose that we have a set of k vectors $\{v_i \in \mathbb{R}^d\}_{i=0}^{k-1}$ that may contain corrupted values, whose identities are not known. We wish to estimate the mean of the trustworthy vectors robustly by minimizing the impact of the corrupt gradients on the mean estimate, potentially by filtering the corrupt gradients out. We consider a simple median-based estimator applied to each coordinate $j = 1, \ldots, d$. First, define the coordinate-wise median as $[v_{\mathsf{med}}]_j = \mathsf{med}\left(\{[v_i]_j\}_{i=0}^{k-1}\right)$, where $\mathsf{med}(\cdot)$ computes the coordinate-wise medians. Then, our estimator is computed as the mean of the nearest $(1-\alpha)k$ neighbors of $[v_{\mathsf{med}}]_j$, where α is a chosen threshold parameter such that $\alpha k \in \mathbb{N}_0$. We propose the estimator

$$[\widehat{v}]_j = \frac{1}{(1-\alpha)k} \sum_{i \in \mathcal{N}_j} [v_i]_j, \tag{2.47}$$

where $\mathcal{N}_j = \{i \in \{0, 1, \ldots, k-1\} : \left|\left[v_i - v_{\mathsf{med}}\right]_j\right| \leq r_j\}$, such that r_j is chosen to satisfy $|\mathcal{N}_j| = (1-\alpha)k$.

The outcome of this estimator depends on the threshold parameter α. If α is chosen such that the number of trustworthy vectors is less than $(1-\alpha)k$, then $\mathcal{N}_j$ will contain arbitrarily corrupted gradients and the estimate will be arbitrarily corrupted. However, if α is chosen such that the number of trustworthy vectors is at least $(1-\alpha)k$, we have the following theoretical guarantees for the performance of this estimator:

Proposition 2.3.1 *[5, Proposition 2] Let $\mathcal{H}$ be the set of trustworthy vectors and $|\mathcal{H}| \geq (1-\alpha)k$. Let $\overline{v}_{\mathcal{H}}$ be the mean of the trustworthy vectors. Suppose that $\max_{i \in \mathcal{H}} \|v_i - \overline{v}_{\mathcal{H}}\|_\infty \leq r$, then for any $\alpha \in [0, 1/2)$, it holds that:*

$$\|\widehat{v} - \overline{v}_{\mathcal{H}}\| \leq C_\alpha r, \tag{2.48}$$

where

$$C_\alpha = \frac{2\alpha}{1-\alpha}\left(1+\sqrt{\frac{(1-\alpha)^2}{1-2\alpha}}\right)\sqrt{d}. \tag{2.49}$$

We note that the right hand side of (2.48) can be approximated as $\mathcal{O}(\alpha r\sqrt{d})$ for small α.

Temporal robust aggregation: Following the mechanics of the robust mean estimator, two scenarios can happen every time temporal robust aggregation is applied in Step 8 of Algorithm 4 to a window of m latest gradients from each agent: (i) there are less than $(1-\alpha_1)m$ trustworthy gradients in the window of size m; (ii) there are at least $(1-\alpha_1)m$ trustworthy gradients in the window of size m. Under scenario (i), $\mathcal{N}_j$ contains arbitrarily corrupted gradients, and therefore we have to assume that the estimated mean is arbitrarily corrupted. We say that the temporal robust aggregation *fails* in this scenario. Under scenario (ii), the estimated mean is close to the true mean of the trustworthy gradients, and the error is bounded by (2.48). Therefore if scenario (ii) happens at any iteration, instead of using a probably corrupt gradient that can be adversarial, the temporal robust aggregation step computes a robustified gradient close to the mean of past trustworthy gradients. We say that the temporal robust aggregation *succeeds* in this scenario. Accordingly, we can view scenario (ii) as a perturbed version of the delayed gradient method, whose convergence properties have been well-studied [85]. Note that both scenarios (i) and (ii) happen with some probability determined by p_t, p_b, α_1 and m.

Spatial robust aggregation: Similar to the temporal robust aggregation step, two scenarios can happen every time spatial robust aggregation is applied in Step 10 of Algorithm 4 to N robustified gradients: (I) there are less than $(1-\alpha_2)N$ agents for which the temporal robust aggregation step succeeds, (II) there are at least $(1-\alpha_2)N$ agents for which the temporal robust aggregation step succeeds. By similar arguments as above, scenario (I) results in an arbitrarily corrupted estimate, whereas scenario (II) results in an estimate that is close to the true mean of the successfully robustified gradients, and the error is bounded by (2.48). We note that both scenarios (I) and (II) happen with some probability determined by α_2, N, and probabilities of

scenarios (i) and (ii).

Gradient normalization: The main idea behind the normalization step is to prevent large updates that corrupt gradients might cause. Since in the case of scenario (I), the temporal robust aggregation and the spatial robust aggregation steps fail to produce an aggregate gradient estimate for which the theoretical error bounds hold, we have to assume that the aggregate gradient estimate becomes arbitrarily corrupted. When this happens and corruptions get past through the two layers of defense, we limit the amount of damage caused by the corrupted aggregate gradient estimate by normalization.

In the next subsection, we state the convergence guarantees of RANGE for strongly convex and smooth (possibly non-convex) cost functions for the SAA setting.

2.3.3 Convergence Properties of RANGE for the SAA Setting

Before presenting the convergence results, we need to state some technical assumptions.

Assumption 2.3.1 *For all $k \in [d]$ and $x \in \mathcal{X}$, define the random variable*

$$f_k(x, z) := \partial_k f(x, z) - \partial_k F(x)$$

. We assume that for all $k \in [d]$ and $x \in \mathcal{X}$, $f_k(x, z)$ is a sub-gamma random variable with variance factor σ and scale parameter a for some $a \geq 0$, i.e.:

$$\log \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[e^{\lambda f_k(x,z)}] \leq \frac{\lambda^2\sigma^2}{2(1-a|\lambda|)},\ \forall x, k, |\lambda| < \frac{1}{a}. \tag{2.50}$$

Assumption 2.3.1 shows bounded moments of the loss function with respect to the data distribution. Note that sub-Gaussian/sub-exponential random variables satisfy Assumption 2.3.1 with $a = 0/a = \sigma$, respectively. Therefore, Assumption 2.3.1 is less restrictive than sub-Gaussian/sub-exponential assumptions in the literature [41, 71].

Assumption 2.3.2 *The function $f(\cdot, z)$ is L-smooth, $\forall z \in \mathcal{Z}$.*

In addition, when $F(\cdot)$ is strongly convex, we have the following assumption on $\mathcal{X}$ and the minimizer of $F(\cdot)$:

Assumption 2.3.3 *The parameter set $\mathcal{X}$ is assumed to be convex and compact with diameter R. Furthermore, $F(x)$ has a unique minimizer $x^\star \in \mathcal{X}$ satisfying $\nabla F(x^\star) = 0$.*

Together with the convexity of F, the above assumption implies that the minimizer of $F(\cdot)$ in $\mathcal{X}$ is also the minimizer of $F(\cdot)$ in $\mathbb{R}^d$. We note that by selecting $\mathcal{X}$ as the euclidean norm ball of a large radius R, the assumption can be satisfied.

Recall from Proposition 2.3.1 that in order for the error bound (2.48) to hold in Algorithm 4 Step 8, the robust mean estimator (2.47) requires that at least $(1-\alpha_1)m$ vectors in $\{g_{i,t-\tau}\}_{\tau=0}^{m-1}$ are trustworthy gradients (scenario (ii) in Sec. 2.3.2). Similarly, in order for the error bound (2.48) to hold in Algorithm 4 Step 10, the robust mean estimator (2.47) requires that at least $(1-\alpha_2)N$ vectors in $\{\hat{g}_{i,t}\}_{i\in[N]}$ are successfully robustified (scenario (II) in Sec. 2.3.2). In order to mathematically formalize these scenarios, we define the following random variables:

- $W_{i,t} = 1$ if $i \in \mathcal{B}^t$, 0 otherwise,
- $Y_{i,t} = 1$ if for agent i, $\sum_{\tau\in[t-m+1,t]} W_{i,\tau} > \alpha_1 m$ (scenario (i)), $Y_{i,t} = 0$ otherwise (scenario (ii)),
- $Z_t = 1$ if $\sum_{i\in[N]} Y_{i,t} > \alpha_2 N$ (scenario (I)), $Z_t = 0$ otherwise (scenario (II)).

Using the above definitions of random variables, when $Y_{i,t} = 1$, the temporal robust aggregation step fails to produce a robustified gradient for agent i. Therefore when $Z_t = 1$, the algorithm fails to produce a robustified update direction as the spatial robust aggregation step becomes contaminated. The challenge of the convergence analysis of Algorithm 4 arises from studying both scenarios $Z_t = 0$ and $Z_t = 1$ along with their probabilities of happening.

However, given the Markovian property of $\{W_{i,t}\}_{\forall t}$, Z_t is not independent of the past, which presents an obstacle in the convergence analysis. To overcome this, we state the next lemma, which establishes a uniform bound on the probability that $Z_t = 1$ given the network state at an earlier time instant:

Lemma 2.3.1 *Let $\mathcal{S}_t = \{x_t, \{\pi_t^i\}_{i\in[N]}\}$ denote the system state at iteration t, where π_t^i is the distribution of the state of agent i at iteration t. Define*

$$\Pi_{m_0}^1 = \frac{p_b + p_t(1 - p_b - p_t)^{m_0}}{p_b + p_t}, \tag{2.51}$$

for all $m_0 \in \mathbb{N}_0$. Given the algorithm parameters $(m, N, \alpha_1, \alpha_2)$ and the transition matrix M, for all $m_0 \in \mathbb{N}_0$ such that $1/2 > \alpha_1 > \Pi_{m_0}^1$ and for all $t \geq m + m_0$, there exists a uniform bound on $\mathbb{P}(Z_t = 1|\mathcal{S}_{t-m+1-m_0}) = \mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]$ independent of $\mathcal{S}_{t-m+1-m_0}$ such that:

$$\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}] \leq P_Z(m_0, m, N, \alpha_1, \alpha_2, M). \tag{2.52}$$

Let $P_Z^m(m_0) := P_Z(m_0, m, N, \alpha_1, \alpha_2, M)$. Then, the above bound holds for:

$$P_Z^m(m_0) = \sum_{k=\alpha_2 N+1}^{N} \binom{N}{k} (P_Y^m(m_0))^k (1 - P_Y^m(m_0))^{(N-k)}, \tag{2.53}$$

where

$$P_Y^m(m_0) = \exp\left(-m(\alpha_1 - \Pi_{m_0}^1)^2(p_b + p_t)\right). \tag{2.54}$$

Proof of Lemma 2.3.1 can be found in Appendix A.2.3. Given a non-negative integer $m_0 \in \mathbb{N}_0$, Lemma 2.3.1 sets an upper bound on $\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]$ independent of the system state at time $t - m + 1 - m_0$, but only as functions of the algorithm parameters $(m, N, \alpha_1, \alpha_2)$, the transition matrix M, and m_0. Although Lemma 2.3.1 provides a practical closed form bound, it is derived by using a Chernoff-type bound for Markov chains [90]. It facilitates

exposition of the method but it is not tight. In Appendix A.2.8, we provide a tighter bound on $P_Z^m(m_0)$. Note that by Hoeffding's inequality, we have $P_Z^m(m_0) \leq e^{-2(\alpha_2 - P_Y^m(m_0))^2 N}$.

Remark 2.3.1 *It is worthwhile to discuss how the upper bound on* (2.53) *depends on p_b and p_t. By chain rule, we have that:*

$$\frac{dP_Z^m(m_0)}{dp_i} = \frac{dP_Z^m(m_0)}{dP_Y^m(m_0)} \times \frac{dP_Y^m(m_0)}{dp_i},\ \mathrm{i} \in \{b, t\}. \tag{2.55}$$

Note that $P_Z^m(m_0)$ is $1 - F_B(\alpha_2 N, N, P_Y^m(m_0))$, where F_B is the cumulative distribution function of the binomial distribution with parameters $(N, P_Y^m(m_0))$ evaluated at $\alpha_2 N$. $F_B(\alpha_2 N, N, P_Y^m(m_0))$ is given by [91]:

$$(N - \alpha_2 N)\binom{N}{\alpha_2 N} \int_0^{1-P_Y^m(m_0)} t^{N-\alpha_2 N-1}(1-t)^{\alpha_2 N} dt, \tag{2.56}$$

which is decreasing with $P_Y^m(m_0)$. Accordingly, we have that $dP_Z^m(m_0)/dP_Y^m(m_0) > 0$. We also show in Appendix A.2.4 that for $m_0 = \mathcal{O}(1/(p_b + p_t))$, i.e., when m_0 is chosen at the order of the mixing time

$$\frac{dP_Y^m(m_0)}{dp_b} > 0, \quad \frac{dP_Y^m(m_0)}{dp_t} < 0. \tag{2.57}$$

Therefore, $dP_Z^m(m_0)/dp_b > 0$ and $dP_Z^m(m_0)/dp_t < 0$. This is in accordance with the intuition that the corruption rate increases with p_b and decreases with p_t. In Figure 2.9, we plot $P_Z^m(m_0)$ for varying p_b and p_t while keeping the rest of the variables constant.

Remark 2.3.2 *We discuss the upper bound in Lemma 2.3.1 for the edge cases of the parameters p_b and p_t. Lemma 2.3.1 requires $0 < p_b < p_t < 1/2$, i.e., the Markov chain to be ergodic. Subsection 2.3.5.3 discusses the case $p_b = p_t = 0$. Moreover, for a fixed p_t, we can evaluate the bounds as $p_b \to 0$. Set $m_0 = k_0/p_t$ and $m = k/(\alpha_1^2 p_t)$ for some $k_0, k \gg 1$. Then, we have*

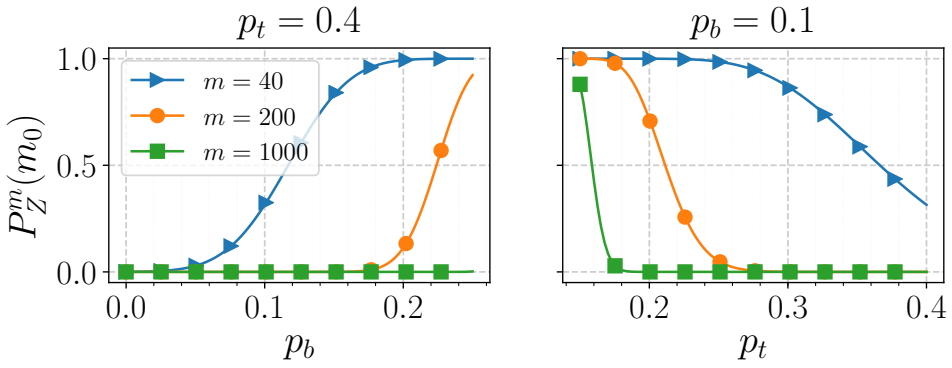


Figure 2.9: Plot of $P_Z^m(m_0)$ for varying p_b with $p_t = 0.4$ (left), and varying p_t with $p_b = 0.1$ (right) for window sizes $m = \{40, 200, 1000\}$. The rest of the parameters are kept constant at $\alpha_1 = 0.45$, $\alpha_2 = 0.3$, $N = 10$, and $m_0 = 100$.

that

$$\lim_{p_b \to 0} P_Y^m(m_0) = \exp\left(-k\frac{(\alpha_1 - (1-p_t)^{\frac{k_0}{p_t}})^2}{\alpha_1^2}\right) \tag{2.58}$$

$$\leq \exp\left(-k(1 - e^{-k_0}/\alpha_1)^2\right) \approx 0 \tag{2.59}$$

Therefore, $P_Z^m(m_0) \approx 0$ *for* $k, k_0 \gg 1$ *as* $p_b \to 0$.

In the other extreme as both $p_b, p_t \to 1/2$, *we have* $\Pi_{m_0}^1 = 1/2$ *and therefore* $P_Y^m(m_0) = 1$, *which results in* $P_Z^m(m_0) = 1$. *This is because when the corruption rate* $p_b/(p_b + p_t) \to 1/2$, *it becomes impossible to distinguish trustworthy information from the corrupted.*

2.3.3.1 Strongly Convex and Smooth Functions

We are now ready to present the main technical result on convergence guarantees of RANGE for a strongly convex cost function $F(\cdot)$. For the following result, we do not require strong convexity of individual cost functions $f(\cdot, z)$.

Theorem 2.3.1 *Let* $F(\cdot)$ *be* μ*-strongly convex and Assumptions 2.3.1, 2.3.2, and 2.3.3 hold. Define the condition number as* $\kappa := L/\mu$. *Let* $m_0 \in \mathbb{N}_0$ *be a non-negative integer such that*

$\Pi^1_{m_0} < 1/2$. *If the algorithm parameters* $(m, N, \alpha_1, \alpha_2)$ *and the transition matrix* M *satisfy*

$$P_Z^m(m_0) < \frac{1}{1+\kappa}, \tag{2.60}$$

then for any $T \geq 1$*, the iterates produced by Algorithm 4 in the SAA setting, with*

$$\gamma \leq \min\left\{\frac{4\sigma}{\overline{C}(m_0)\mu\sqrt{(1-\alpha_2)Nb}}, \frac{\kappa R}{2}\right\}, \tag{2.61}$$

where

$$\begin{aligned}\overline{C}(m_0) =& 1 + 4P_Z^m(m_0)(1+1/\kappa)(m-1+m_0) \\ &+ 4\kappa(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1)),\end{aligned} \tag{2.62}$$

have the following property:

$$\begin{aligned}\mathbb{E}[\|x_{T+m+m_0} - x^\star\|^2] \leq& \left(\|x_1 - x^\star\| + \gamma(m+m_0-1)\right)^2 \left(1 - c_0(m_0)\gamma\right)^T \\ &+ \mathcal{O}\left(\frac{1}{\sqrt{Nb}} + \frac{C_{\alpha_2}}{\sqrt{b}}\right),\end{aligned} \tag{2.63}$$

where

$$c_0(m_0) = \frac{2}{\kappa R}(1 - P_Z^m(m_0)(1+\kappa)), \tag{2.64}$$

and C_{α_i} *for* $i = 1, 2$*, are given by* (2.49)*.*

Proof outline: The proof follows by bounding the distance of x_{t+1} to the optimal solution $x^\star$ in terms of x_t via perturbed gradient analysis. We define the perturbation as $\nabla F(x_t)/\|\nabla F(x_t)\| - \hat{\hat{g}}_t/\|\hat{\hat{g}}_t\|$ and bound the norm of the perturbation in the events $Z_t = 1$ and $Z_t = 0$ separately.

The complete proof of Thoerem 2.3.1 can be found in Appendix A.2.1. When $Z_t = 1$,

we assume the worst-case scenario such that $\hat{\hat{g}}_t$ is moving x_t in the opposite direction of $x^\star$. When $Z_t = 0$, we split the perturbation into 3 terms (Lemma A.2.1 in Appendix A.2.2): 1) the error due to stochastic and delayed gradients, which depends on the variance σ^2, step-size γ, window size m, 2) the error of the temporal robust aggregator given by the median-based mean estimator's bound (2.48), which is proportional to the maximum distance between the stochastic gradients in a window of m, and 3) the error of the spatial robust aggregator given by (2.48) again, which depends on the maximum distance between the robustified gradients of the N agents. To upper bound the expected value of the aforementioned maximum distances, we use Theorem 2.5 in [92], which offers a convenient bound for the expected value of the maximum of finitely many exponentially integrable random variables. □

The complete proof of Theorem 2.3.1 and the explicit constants of (2.63) can be found in Appendix A.2.1. According to Theorem 2.3.1, RANGE provides convergence to a neighborhood of the optimal solution at a linear rate as long as (2.60) is satisfied. The neighborhood of convergence is

$$\mathcal{O}\left(\frac{1}{\sqrt{Nb}} + \frac{C_{\alpha_2}}{\sqrt{b}}\right), \tag{2.65}$$

where N is the number of agents and b is the number of data samples used for gradient computation (i.e., mini-batch size).

Remark 2.3.3 *The convergence rate in* (2.63) *is governed by the term* $(1 - c_0(m_0)\gamma)$. *Accordingly, the smaller* $c_0(m_0)$*, the slower the convergence. Note that* $c_0(m_0)$ *given by* (2.64) *is decreasing with* κ*. For ill-conditioned problems with big* κ*,* $c_0(m_0)$ *is small and therefore convergence is slower. Additionally, observe that* $c_0(m_0)$ *is decreasing with* $P_Z^m(m_0)$*. In accordance with Remark 2.3.1, the bigger* p_b *or the smaller* p_t*, the slower the convergence since* $P_Z^m(m_0)$ *gets bigger.*

Remark 2.3.4 *In Appendix A.2.1, we show that the only dependence of the neighborhood of convergence in* (2.65) *on* m_0 *is through* $P_Z^m(m_0)$ *given by* (2.53)*. By taking* $m_0 \gg 1$*, we*

minimize (2.51) *to get* $\Pi^1_{m_0} \approx p_b/(p_b + p_t)$*, which then minimizes* $P^m_Y(m_0)$ *and* $P^m_Z(m_0)$*. Hence we get a tight asymptotic bound with respect to* m_0 *for* $m_0 \to \infty$*.*

Impact of Temporal Robust Aggregation: The temporal robust aggregation step helps reducing the neighborhood of convergence by reducing the effective fraction of Byzantine agents at each iteration. The convergence neighborhood given by (2.65) consists of two terms: first term due to variance of the stochastic gradients and the second term due to Byzantine agents. In [41], it is shown that in the setting with a bounded α fraction of Byzantine agents, no algorithm can achieve an error lower than

$$\tilde{\Omega}\left(\frac{1}{\sqrt{Nb}} + \frac{\alpha}{\sqrt{b}}\right). \tag{2.66}$$

In the stationary distribution of the Markov chain, the probability that an agent is Byzantine is equal to $p_b/(p_b+p_t)$, and the expected fraction of Byzantine agents in an iteration is $p_b/(p_b+p_t)$. Therefore, it is reasonable to argue that α in (2.66) is similar to $p_b/(p_b+p_t)$. For Lemma 2.3.1 to hold, we need $\alpha_1 > \Pi^1_{m_0} \approx p_b/(p_b + p_t)$ for $m_0 \gg 1$ and thus α_1 is of the order of α in (2.66). On the other hand, our error bound in (2.65) is a function of C_{α_2}, where $C_{\alpha_2} = \mathcal{O}(\alpha_2)$ for small α_2. Because RANGE aims to eliminate α_2 fraction of agents' robustified gradients via spatial robust aggregation, it can be viewed as the effective fraction of Byzantine agents, and hence our bound is consistent with (2.66). Interestingly, we can set α_2 as arbitrarily small as possible. Note that we need to satisfy (2.60) for convergence. However, in (2.53), we can select α_2 sufficiently small for small $P^m_Y(m_0)$. But $P^m_Y(m_0)$ is given by (2.54), which is a function of α_1, m, and m_0. As such, we can always set m arbitrarily large such that $P^m_Y(m_0)$ is arbitrarily small for all $m_0 \in \mathbb{N}_0$, and hence we can select α_2 arbitrarily small to satisfy (2.60). As a result, by employing the temporal robust aggregation step before spatial robust aggregation, which is beneficial only when agents' states change over time, we reduce the effective fraction of Byzantine agents from $p_b/(p_b+p_t)$ to α_2. This setup shows that the lower bound (2.66) does not hold for the proposed Markovian setting as a result of temporal robust aggregation.

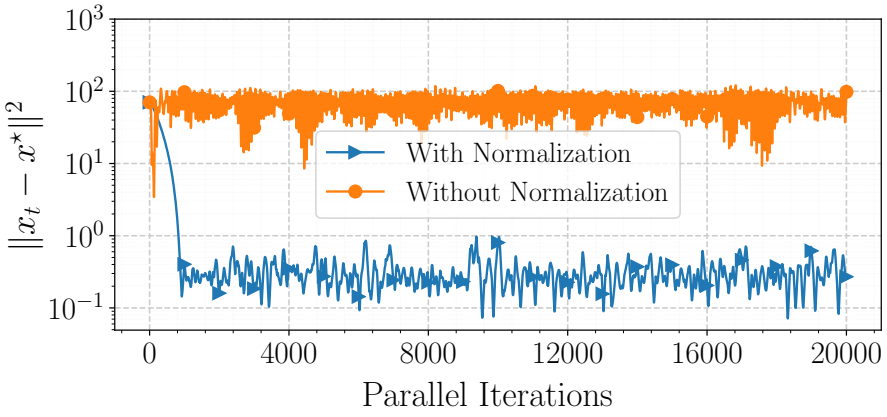


Figure 2.10: RANGE with and without the normalization step for the linear regression problem in Subection 2.3.6.1.

Remark 2.3.5 *Without any corruption, RANGE suffers an error of* $\mathcal{O}(1/\sqrt{Nb})$*. Even if we use unbiased stochastic gradients with diminishing step-sizes, this error is unavoidable due to the normalization step. This is because as observed by [80], if* $\nabla F_i(x_t) = \nabla F(x_t) + \eta$ *for very small* η*, it might be that* $\nabla F_i(x_t)/\|\nabla F_i(x_t)\|$ *is very far from* $\nabla F(x_t)/\|\nabla F(x_t)\|$*, especially when* $\|\nabla F(x_t)\|$ *is close to 0. On the other hand, the normalization step is the key feature of our algorithm to achieve robustness. Due to the stochasticity of the Markovian model, there is a probability (* $\leq P_Z^m(m_0)$*) that the first two layers of robust aggregation fail. In this case, the overall aggregate can be arbitrarily corrupted and we require normalization to defend against such cases. Figure 2.10 numerically demonstrates the importance of normalization by simulating RANGE with and without normalization for the linear regression problem in Subsection 2.3.6.1.*

Observe that choice of m, α_1, and α_2 plays an important role, as they determine whether (2.60) holds or not. Next, we discuss how to pick m, α_1, and α_2 in practice.

Choices of m**,** α_1**, and** α_2**:** For a given transition matrix M, there always exists a set of parameters m, α_1, and α_2 such that (2.60) is satisfied. Finding a principled way to select the set of optimal parameters is currently an open question. Instead, next we describe an implementable closed form expression for the minimum window size as a function of α_1, α_2,

p_t, and p_b. Using Hoeffding's inequality on (2.53), we rewrite (2.60) as:

$$P_Z^m(m_0) \leq \exp(-2(\alpha_2 - P_Y^m(m_0))^2 N) < \frac{1}{1+\kappa}. \tag{2.67}$$

This gives us the condition on α_2[4]:

$$\alpha_2 > P_Y^m(m_0) + \sqrt{\frac{\log(1+\kappa)}{2N}}. \tag{2.68}$$

Considering (2.54), given $m_0 \in \mathbb{N}_0$, if

$$\exp\left(-m(\alpha_1 - \Pi_{m_0}^1)^2(p_b + p_t)\right) < \alpha_2 - \sqrt{\frac{\log(1+\kappa)}{2N}} \tag{2.69}$$

holds, then (2.60) is satisfied. Additionally, we require that $\alpha_1 < 0.5$ for the algorithm's input so that the robust mean estimator succeeds, which requires $\Pi_{m_0}^1 < 0.5$. This also gives us a lower bound on m_0:

$$m_0 > \frac{\log(p_t - p_b) - \log(2p_t)}{\log(1 - p_b - p_t)} \tag{2.70}$$

We can rearrange (2.68) to get the minimum window size that is sufficient for convergence as a function of α_1, α_2, p_t, p_b, and any choice of m_0 that satisfies (2.70):

Corollary 2.3.1 *For all* $m_0 \in \mathbb{N}_0$ *satisfying* (2.70)*, if* $\alpha_2 > \sqrt{\log(1+\kappa)/(2N)}$*,* $\alpha_1 > \Pi_{m_0}^1$*, and*

$$m > -\frac{\log(\alpha_2 - \sqrt{(1+\kappa)/(2N)})}{(\alpha_1 - \Pi_{m_0}^1)(p_b + p_t)} \tag{2.71}$$

then (2.60) *holds.*

Corollary 2.3.1 is convenient in practice for selecting α_1, α_2, and m. Given p_b,

[4]This is the condition on α_2 for the Hoeffding bound (2.67) to hold, rather than the exact inequality in (2.53). Consequently, it results in an additive $\sqrt{\log(1+\kappa)/(2N)}$ term that is independent of $P_Y^m(m_0)$. However, this does not contradict our statement that we can set α_2 arbitrarily small by reducing $P_Y^m(m_0)$ with a large window m, since that statement is based on the exact form in (2.53) rather than the Hoeffding bound (2.67).

p_t, and κ, one picks m_0 and α_1 such that $\Pi^1_{m_0} < \alpha_1 < 0.5$ and α_2 such that $\sqrt{\log(1+\kappa)/(2N)} < \alpha_2 < 0.5^4$ and $\alpha_2 N \in \mathbb{N}_0$. Then, the window size is m is picked such that $\alpha_1 m \in \mathbb{N}_0$ and (2.71) holds. With these parameter choices, (2.60) is satisfied. Note that the $(p_t + p_b)$ term in the denominator of (2.71) corresponds to the spectral gap of the Markov chain. Therefore, smaller spectral gap, which also implies larger relaxation and mixing time [93], results in a larger minimum window size. Intuitively, we select the window size at the order of the mixing time so that the Markov chain gets close to its stationary distribution and the Byzantine agents transition into trustworthy state. This allows the temporal robust aggregation step to successfully produce a robustified gradient by extracting the trustworthy information.

2.3.3.2 Smooth (Possibly Non-Convex) Functions

Next, we study the convergence of RANGE for smooth (possibly non-convex) cost functions. For this problem class, we need the following assumption on $\mathcal{X}$:

Assumption 2.3.4 *Problem* (2.38) *is unconstrained, i.e.,* $\mathcal{X} = \mathbb{R}^d$.

The next theorem states the convergence guarantees of RANGE for smooth $F(\cdot)$:

Theorem 2.3.2 *Let Assumptions 2.3.1, 2.3.2, and 2.3.4 hold. Choose step-size* $\gamma = \gamma_0/\sqrt{T}$ *with* $\gamma_0 > 0$. *Let* $m_0 \in \mathbb{N}_0$ *be a non-negative integer such that* $\Pi^1_{m_0} < 1/2$. *If the algorithm parameters* $(m, N, \alpha_1, \alpha_2)$ *and the transition matrix* M *satisfy*

$$P_Z^m(m_0) < 1/2, \tag{2.72}$$

then for any $T \geq 1$, *the iterates* $\{x_t\}_{t=m+m_0}^{T+m-1+m_0}$ *produced by Algorithm 4 in the SAA setting*

satisfy

$$\frac{1}{T}\sum_{t=m+m_0}^{T+m-1+m_0}\mathbb{E}[\|\nabla F(x_t)\|] \leq \frac{F(x_1)-F(x^\star)}{\sqrt{T}\gamma_0(1-2P_Z^m(m_0))}+\frac{\overline{C}(m_0)\gamma_0}{\sqrt{T}}+\mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{\sqrt{Nb}}+\frac{C_{\alpha_2}}{\sqrt{b}}\right), \tag{2.73}$$

where

$$\overline{C}(m_0) = L\Big(1/2+4(m-1+m_0)P_Z^m(m_0) + 2(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1))\Big), \tag{2.74}$$

and C_{α_i} for $i=1,2$, are given by (2.49).

The proof of Theorem 2.3.2 is similar to that of Theorem 2.3.1, and the complete proof and the explicit constants of (2.73) can be found in Appendix A.2.5. According to Theorem 2.3.2, RANGE produces a point $\tilde{x} \in \{x_{m+m_0}, \ldots, x_{T+m-1+m_0}\}$ such that

$$\mathbb{E}\left[\|\nabla F(\tilde{x})\|\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{Nb}}+\frac{C_{\alpha_2}}{\sqrt{b}}+\frac{1}{\sqrt{T}}\right). \tag{2.75}$$

Note that when $T \to \infty$, the right-hand side of the above inequality is the same as (2.65).

Remark 2.3.6 *We mention that Theorem 2.3.2 is valid for smooth convex cost functions and strongly convex cost functions with an unbounded parameter set as well. For convex functions with a bounded set, we can add regularization terms to make them strongly convex, in which case the guarantees of Theorem 2.3.1 hold. Adding regularization terms is a typical technique used in optimization, called dual smoothing [50].*

In the next subsection, we state the convergence guarantees of RANGE for strongly convex and smooth (possibly non-convex) cost functions for the SA setting.

2.3.4 Convergence Properties of RANGE for the SA Setting

Theorems 2.3.1 and 2.3.2 state convergence guarantees of RANGE for the SAA setting. The next theorem states the convergence result for the SA setting for strongly convex cost functions.

Theorem 2.3.3 *Let $F(\cdot)$ be μ-strongly convex and Assumptions 2.3.1, 2.3.2, and 2.3.3 hold. Define the condition number as $\kappa := L/\mu$. Let $m_0 \in \mathbb{N}_0$ be a non-negative integer such that $\Pi_{m_0}^1 < 1/2$. If the algorithm parameters $(m, N, \alpha_1, \alpha_2)$ and the transition matrix M satisfy*

$$P_Z^m(m_0) < \frac{1}{1+\kappa}, \tag{2.76}$$

then for any $T \geq 1$, the iterates produced by Algorithm 4 in the SA setting, with

$$\gamma \leq \min\left\{\frac{4\sigma}{\overline{C}(m_0)\mu\sqrt{(1-\alpha_2)N(1-\alpha_1)mb}}, \frac{\kappa R}{2}\right\}, \tag{2.77}$$

have the following property:

$$\begin{aligned}\mathbb{E}[\|x_{T+m+m_0} - x^\star\|^2] \leq &\left(\|x_1 - x^\star\| + \gamma(m+m_0-1)\right)^2 (1-c_0(m_0)\gamma)^T \\ &+ \mathcal{O}\left(\frac{1}{\sqrt{(1-\alpha_1)mNb}} + \frac{C_{\alpha_2} + C_{\alpha_1}(1+C_{\alpha_2})}{\sqrt{b}}\right),\end{aligned} \tag{2.78}$$

where $c_0(m_0)$, $\overline{C}(m_0)$ and C_{α_i} for $i = 1, 2$, are given by (2.64)*,* (2.62)*, and* (2.49)*.*

Proof of Theorem 2.3.3 and the explicit constants of (2.78) can be found in Appendix A.2.6. According to Theorem 2.3.3, RANGE provides convergence to a neighborhood of the optimal solution at a linear rate, where the neighborhood of convergence is

$$\mathcal{O}\left(\frac{1}{\sqrt{(1-\alpha_1)mNb}} + \frac{C_{\alpha_2} + C_{\alpha_1}(1+C_{\alpha_2})}{\sqrt{b}}\right). \tag{2.79}$$

Comparing the above result to (2.65), there are two impacts of using the SA setting instead of the SAA setting:

1. The error due to variance of the stochastic gradients reduces by a factor of $\mathcal{O}(\sqrt{(1-\alpha_1)m})$,

2. The error due to Byzantine agents increases by a factor of $\mathcal{O}(1 + C_{\alpha_1} + C_{\alpha_1}/C_{\alpha_2})$.

When agents use new samples at each iteration in the SA setting, temporal robust aggregation results in a variance reduction, since it estimates the mean of $(1-\alpha_1)m$ independent minibatch gradients. However, this comes at the cost of the higher error caused by the Byzantine agents. Given these two counteracting impacts of the SA setting on the error, the order of error in (2.79) is less than (2.65) if

$$C_{\alpha_1} < \frac{1}{\sqrt{N}(1 + C_{\alpha_2})}. \tag{2.80}$$

Therefore, RANGE performs better in the SA setting compared to the SAA setting if $\alpha_1 \ll 1$, which is possible if $p_b \ll p_t$. Otherwise, the benefit of variance reduction provided by temporal robust aggregation is dominated by the damage caused by the Byzantine agents.

The next theorem states the convergence result of RANGE for the SA setting for non-convex cost functions.

Theorem 2.3.4 *Let Assumptions 2.3.1, 2.3.2, and 2.3.4 hold. Choose step-size $\gamma = \gamma_0/\sqrt{T}$ with $\gamma_0 > 0$. Let $m_0 \in \mathbb{N}_0$ be a non-negative integer such that $\Pi^1_{m_0} < 1/2$. If the algorithm parameters $(m, N, \alpha_1, \alpha_2)$ and the transition matrix M satisfy*

$$P^m_Z(m_0) < 1/2, \tag{2.81}$$

then for any $T \geq 1$, the iterates $\{x_t\}_{t=m+m_0}^{T+m-1+m_0}$ produced by Algorithm 4 in the SA setting

satisfy

$$\frac{1}{T}\sum_{t=m+m_0}^{T+m-1+m_0}\mathbb{E}[\|\nabla F(x_t)\|] \leq \frac{F(x_1)-F(x^\star)}{\sqrt{T}\gamma_0(1-2P_Z^m(m_0))}+\frac{\overline{C}(m_0)\gamma_0}{\sqrt{T}}+\mathcal{O}\left(\frac{1}{T}\right) + \mathcal{O}\left(\frac{1}{\sqrt{(1-\alpha_1)mNb}}+\frac{C_{\alpha_2}+C_{\alpha_1}(1+C_{\alpha_2})}{\sqrt{b}}\right), \tag{2.82}$$

where $\overline{C}(m_o)$ *and* C_{α_i} *for* $i=1,2$*, are given by* (2.74) *and* (2.49)*.*

Proof of Theorem 2.3.4 and the explicit constants of (2.82) can be found in Appendix A.2.7. Note that when $T\to\infty$, the right hand side of (2.82) is the same as (2.79).

2.3.5 Special Cases

2.3.5.1 Window Size $m=1$

When we select the window size m to be 1, we have to set $\alpha_1=0$ since $\alpha_1 m\in\mathbb{N}_0$ and $\alpha_1<0.5$. This means that we skip the temporal robust aggregation step and set the robustified gradient $\hat{g}_{i,t}=g_{i,t}$. In this case, the counterpart of Lemma 2.3.1 with $m=1$ gives:

$$P_Z^1(m_0)\leq\sum_{k=\alpha_2 N+1}^{N}\binom{N}{k}(P_Y^1(m_0))^k(1-P_Y^1(m_0))^{(N-k)}, \tag{2.83}$$

where

$$\begin{aligned}P_Y^1(m_0)&=\max_{i\in[N],t}\mathbb{P}(Y_{i,t}=1|\mathcal{S}_{t-m_0})=\max_{i\in[N],t}\mathbb{P}(W_{i,t}=1|\mathcal{S}_{t-m_0})\\&=\frac{p_b+p_t(1-p_b-p_t)^{m_0}}{p_b+p_t}.\end{aligned} \tag{2.84}$$

Accordingly, Theorems 2.3.1, 2.3.2, 2.3.3, and 2.3.4 hold with $m=1$ and $P_Z^1(m_0)$ given by (2.83) and (2.84). Note that $P_Y^1(m_0)\to p_b/(p_b+p_t)$ as $m_0\to\infty$.

2.3.5.2 Independent Random Corruption

A special case of the agents' state transition occurs when $p_b + p_t = 1$. We get $M = [\pi^{\star T}\ \pi^{\star T}]^T$, and hence the state of an agent at $t+1$ is independent of the state at t. In this case, an agent becomes Byzantine and sends corrupted gradient information randomly with probability p_b at all iterations. Hence, we can state the counterpart of Lemma 2.3.1 without the need to condition on a previous time instant, i.e.,

$$P_Z^m := \mathbb{E}[Z_t] = \sum_{k=\alpha_2 N+1}^{N} \binom{N}{k} (P_Y^m)^k (1-P_Y^m)^{(N-k)}, \tag{2.85}$$

where

$$P_Y^m = \sum_{j=\alpha_1 m+1}^{m} \binom{m}{j} p_b^j p_t^{m-j}. \tag{2.86}$$

Accordingly, Theorems 2.3.1, 2.3.2, 2.3.3, and 2.3.4 hold with $m_0 = 0$ and $P_Z^m(m_0)$ replaced by P_Z^m given by (2.85) and (2.86).

2.3.5.3 Recovering the Static Corruption Model

The static setting where agents do not change states is recovered by letting $p_b = p_t = 0$. Recall from Remark 2.3.2 that Lemma 2.3.1 can not be used in this case as it requires the Markov chain to be ergodic. However, if we assume that $p_b/(p_b+p_t)$ fraction of the agents are initially corrupted, our results recover the static setting by choosing $\alpha_2 = p_b/(p_b+p_t)$ and $m_0 = 0$ to ensure $P_Z^m(0) = 0$ for any m. Furthermore, we can remove the temporal robust aggregation step by letting $m = 1$ and $\alpha_1 = 1$, since it will not eliminate any corruption and does not provide any variance reduction for the SAA setting.

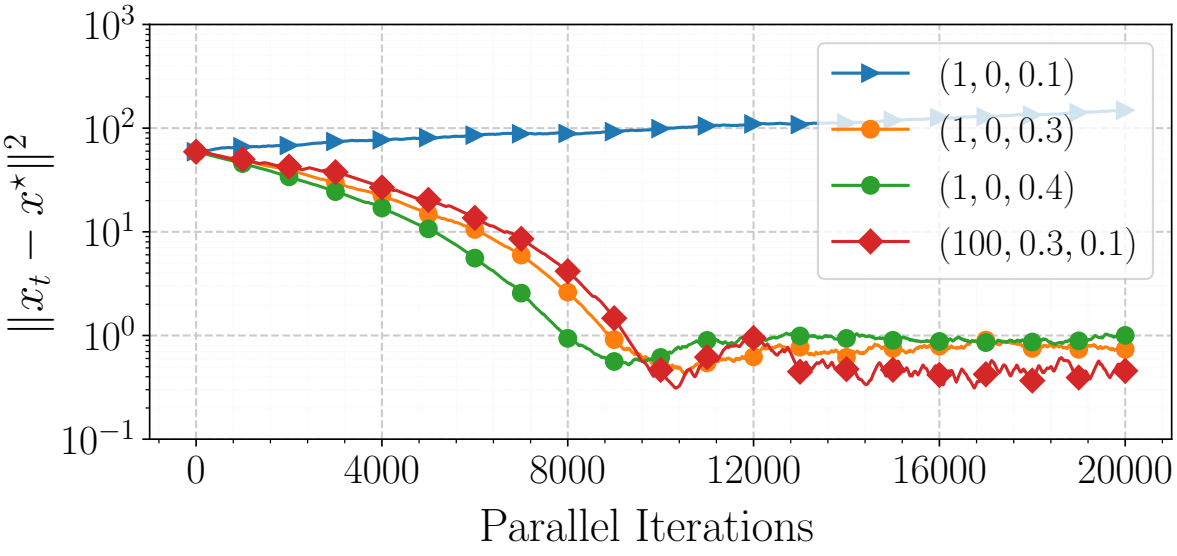


Figure 2.11: Convergence performance of RANGE in linear regression with four configurations of (m, α_1, α_2) for $p_t = 0.1$, $p_b = 0.025$.

2.3.6 Numerical Experiments

In this subsection, we present numerical evidence supporting our theoretical results and demonstrating the efficacy of RANGE. The first experiment is a simple linear regression with synthetic data to illustrate the benefits of the temporal robust aggregation step of RANGE and to compare the SAA and the SA settings. The second experiment is an image classification task on the EMNIST dataset [94] using a neural network to compare the performance of RANGE to existing distributed optimization algorithms in practical non-convex tasks for the SAA setting. Both experiments were performed on a laptop computer with Intel® Core™ i7-8750H CPU (6×2.20 GHz) and 16 GB DDR4 2666MHz RAM.

2.3.6.1 Linear Regression with Synthetic Data

We consider the following stochastic optimization problem:

$$x^\star = \arg\min_{x\in\mathcal{X}} \mathop{\mathbb{E}}_{V,Y}[|Y - V^T x|], \tag{2.87}$$

where $V \in \mathbb{R}^d$ is the random vector corresponding to the data points and $Y \in \mathbb{R}$ is the random variable corresponding to the associated label values or outputs. We let $d = 100$ and constructed the solution vector $x^\star$ by sampling a random point from the interior of the d-ball with radius $R = 10$. In the SAA setting, the goal is to solve the following deterministic optimization problem

$$\min_{x\in\mathcal{X}} \|y - vx\|^2, \tag{2.88}$$

where $v \in \mathbb{R}^{B\times d}$ is a matrix containing the B data vectors in its rows and $y \in \mathbb{R}^B$ is the vector containing the B associated label values or outputs. We let $B = 1000$ and randomly generated the entries of v from $\mathcal{N}(0, 1)$. We distributed the data points equally among $N = 10$ agents. For all $i \in [B]$, we generated the outputs y according to $y_i = v_i x^\star + \xi_i$, where $\xi_i \sim \mathcal{N}(0, R^2)$ is the noise and v_i is the i'th row of v.

The main goal of the experiment is to demonstrate: 1) how α_2 affects the robustness and the performance of the algorithm, and 2) how temporal robust aggregation can help us pick a better α_2 and improve the performance. We note that there are two effects of α_2 on the performance: 1) In the spatial robust aggregation step, the algorithm eliminates α_2 fraction of the robustified gradients $\{\hat{g}_{i,t}\}_{i\in[N]}$ and therefore provides robustness in case there are less than α_2 fraction of corrupted $\{\hat{g}_{i,t}\}_{i\in[N]}$, and 2) by aggregating $(1-\alpha_2)N$ number of agents' robustified gradients, it reduces the variance of the stochastic gradients. In order to guarantee robustness, it is desirable to increase α_2 so that we do not add the corrupted $\{\hat{g}_{i,t}\}_{i\in[N]}$ to the aggregate. On the other hand, increasing α_2 results in aggregating less number of agents'

gradients, which in turn results in a higher variance of the stochastic gradients. Given these counteracting impacts of α_2, it is not desirable to choose it too small or too big.

We ran RANGE for 20k iterations $p_t = 0.1$ and $p_b = 0.025$ using four configurations of (m, α_1, α_2). At each iteration, we picked the corrupt gradient as $2\|\nabla F(x_t)\|(x^\star - x_t)/\|x^\star - x_t\|$. In Figure 2.11, we plot the convergence behaviour of RANGE for all four configurations. When $m = 1$ and $\alpha_2 = 0.1$, we observe that the iterates diverge. Since $0.1 < p_b/(p_b + p_t) = 0.2$, the expected value of the fraction of Byzantine agents at each iteration is larger than α_2, and hence the aggregate gradient estimate becomes corrupted most of the time. On the contrary, setting $\alpha_2 = 0.3$ or $\alpha_2 = 0.4$ provides robustness and RANGE converges. However, we observe that the configuration with $\alpha_2 = 0.3$ performs slightly better than the one with $\alpha_2 = 0.4$. This is because smaller α_2 aggregates more agents' gradients, which results in a larger variance reduction. All in all, while selecting a smaller α_2 provides variance reduction, it reduces the robustness of RANGE by including more agents at the spatial robust aggregation step.

On the other hand, the configuration with $m = 100$, $\alpha_1 = 0.3$ and $\alpha_2 = 0.1$ outperforms the rest. When we utilize the temporal robust aggregation step, it effectively reduces the expected value of the fraction of Byzantine agents at each iteration. Consequently, we can select a smaller α_2 in order to benefit from larger variance reduction while still being robust thanks to temporal robust aggregation.

Next, we study the SA setting by re-sampling B new data points at each iteration. Keeping $p_t = 0.1$ constant, we simulate a high corruption rate with $p_b = 0.025$ and a low corruption rate with $p_b = 0.01$. We set $\alpha_1 = 0.3$ when $p_b = 0.025$ and $\alpha_1 = 0.1$ when $p_b = 0.01$. We fixed the window size $m = 100$ and $\alpha_2 = 0.1$ for both cases. In Figure 2.12 we plot the convergence behavior of RANGE in the SA and the SAA settings for both corruption rates. We observe that when the corruption rate is low, RANGE performs better in the SA setting due to variance reduction provided by temporal robust aggregation. However, when the corruption

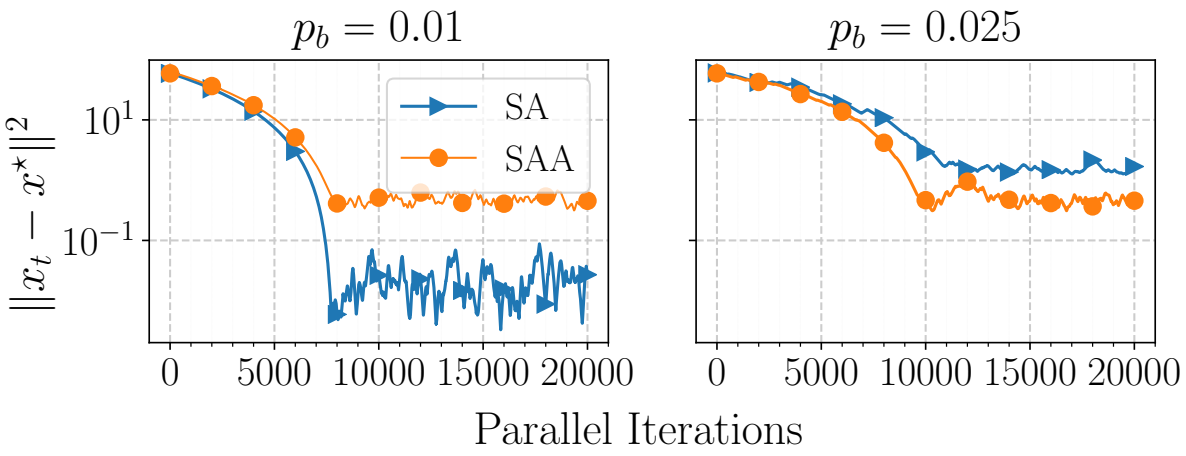


Figure 2.12: Comparison of the SA and the SAA settings in linear regression.

rate is high, RANGE performs worse in the SA setting as the damage caused by the Byzantine agents dominates.

2.3.6.2 Image Classification with EMNIST Dataset

In this study, we experiment with the image classification task on the EMNIST dataset [94] in the SAA setting. We train a feed-forward neural network with two hidden layers and 64 neurons at each hidden layer. We partition the data into $N = 200$ equal sizes, representing the data at N agents. The batch size for gradient computation is set to $b = 300$.

We train the neural network using (a) RANGE, (b) vanilla SGD, (c) median aggregation [41], and (d) norm clipping [82]. We note that although the median aggregation method in [41] and the norm clipping method in [82] are developed for the setting with a bounded fraction of Byzantine agents, we implement them in the Markovian Byzantine agent setting because no existing work studies the same setup as ours. For transition probabilities $p_t = 0.2$ and $p_b \in \{0,\ 0.05,\ 0.15\}$, we train three networks with learning rates 0.1, 0.01, and 0.001, and pick the best-performing one. We let $m = 50$, $\alpha_1 = 0.25$, $\alpha_2 = 0.2$ for RANGE when $p_b = 0$ and $p_b = 0.05$; $m = 50$, $\alpha_1 = 0.45$, $\alpha_2 = 0.3$ when $p_b = 0.15$. We set the threshold for

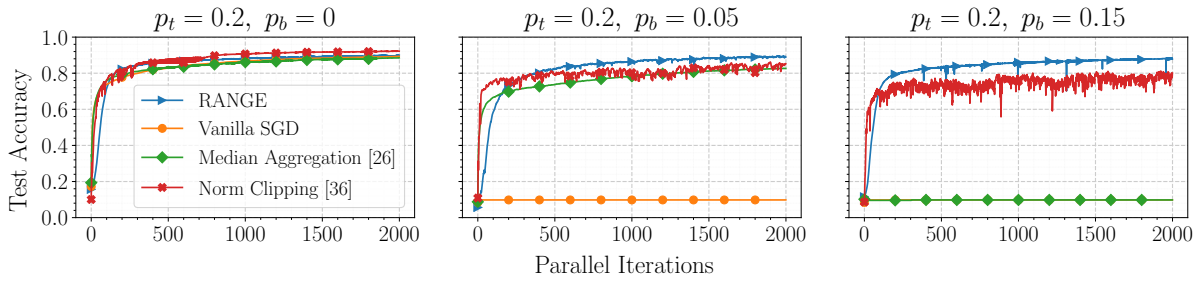


Figure 2.13: Training neural networks for image classification on the EMNIST dataset. Four distributed optimization algorithms are compared under $p_t = 0.2$ and $p_b = 0$ (left), $p_b = 0.015$ (middle), and $p_b = 0.15$ (right) using test accuracy of the trained network as metric. The legend is shared among all plots.

norm clipping to be 10 as it is shown to perform well in [82]. We simulate corruption by simply inverting and boosting the magnitude of the gradient, i.e., by setting $\star = -c\nabla F_{i,t}(x_t)$ in (2.40), where c is sampled uniformly from $[5, 15]$ at each iteration.

In Figure 2.13 we plot the test accuracy of the models during training. For $p_b = 0$, all algorithms successfully train the neural network as expected. For $p_b = 0.05$, vanilla SGD has negligible performance as it is not robust to corruption. On the other hand, norm clipping and median aggregation methods have satisfactory performance. Nonetheless, RANGE outperforms norm clipping and median aggregation algorithms by margins of 3.7% and 6.3%, respectively. For $p_b = 0.15$, we observe that the median aggregation method also fails. The median is no longer robust since it is corrupted if more than half of the agents become Byzantine at any iteration, which frequently happens when $p_b/(p_b + p_t)$ is high. Norm clipping is still robust, however, RANGE beats it by a margin of 8.3%.

2.3.7 Conclusions

We introduced a distributed optimization algorithm, named RANGE, that is provably robust to Byzantine failures. By modeling each agent's state transition trajectory over time, namely from trustworthy to Byzantine and vice versa, as a two-state Markov chain, we allow all the

agents to be prone to failure. RANGE is based on three ideas: 1) temporal robust aggregation, which computes a robustified gradient for each agent by estimating a robust mean of a window of past gradients, 2) spatial robust aggregation, which computes a robust mean of all the agents' robustified gradients to estimate the aggregate gradient, and 3) gradient normalization, which restricts the aggregate gradient to only contain directional information and therefore prevents arbitrarily large updates that corrupt gradients might cause. We prove that for strongly convex and smooth non-convex cost functions RANGE achieves convergence to a neighborhood of a stationary point, where the neighborhood depends on the variance of the stochastic gradients and the corruption rate. Numerical experiments on linear regression and image classification on EMNIST dataset demonstrate the robustness and efficacy of RANGE.

Chapter 3

Safe Pricing for Resource Allocation in Safety-Critical Networks

3.1 Introduction

Many applications falling within the scope of resource allocation over networks, e.g., power distribution systems [15], congestion control in data networks [18], wireless cellular networks [17], and congestion control in urban traffic networks [22], deal with a multi-user optimization problem that falls under the general umbrella of *network utility maximization* (NUM) problems. The shared goal in these problems is to *safely* and *efficiently* allocate the shared resources to the users, where safety refers to satisfying the constraints of the system that depend on the resource allocation of all the users, and efficiency refers to the total utility of the users for a given resource allocation.

In NUM problems, the user-specific utility functions are assumed to be private to the users and therefore a centralized solution is not possible. Accordingly, distributed optimization methods have become suitable tools thanks to the separable structure of NUM problems [23, 95]. The idea is to decompose the main problem into sub-problems that can be solved by the in-

dividual users and use the solutions of the sub-problems to solve the main problem [96, 97], and this has been advocated for use in different applications, e.g., [16, 18]. Among the two main types of decomposition methods, primal decomposition methods correspond to a direct allocation of the resources by a central coordinator and solve the primal problem, whereas dual decomposition methods based on the Lagrangian dual problem [98] correspond to resource allocation via pricing and solve the dual problem [23]. Due to the structure of NUM problems, the latter approach has been widely adopted in the literature [23, 99, 100]. Additionally, it gives users the freedom of determining their own demand based on pricing-type signals.

Although there is extensive literature on pricing algorithms based on dual decomposition, the majority of studies focus on linear constraints [99, 100, 101, 102, 9], or on non-linear constraints with the assumption of separability and full user knowledge of these constraints [103, 104, 105]. Furthermore, none of the aforementioned studies propose an iterative pricing algorithm that induces resource demand satisfying the hard constraints of the problem *during* the iterative optimization process. Instead, these studies only provide bounds on the infeasibility amount of the resource demand (e.g., [100, 102]). Our preliminary work in [9] is an exception, which is limited to problems with linear inequality constraints characterized by binary matrices. Thus, pricing-based solutions can only be realized after convergence to a near-feasible point for resource allocation systems with safety-critical constraints. Therefore, implementation of such solutions requires a negotiation process through a two-way communication network if the system has hard safety-critical constraints, which can be considered impractical in many applications.

The research presented in this chapter is motivated by the context of network resource allocation applications, which involve a number of key considerations. First, users themselves determine their own resource demand in response to the prices, with the actual demand only becoming observable ex-post. Second, it is essential that the systems in question have safety-critical hard constraints that must not be violated by users' resource demand at any time. Fi-

nally, it is important to recognize that the constraints associated with such resource allocation systems can form arbitrary convex and compact feasible sets. One particularly relevant example of this type of application can be seen in the context of price-based demand response, in which users determine their own electricity consumption in response to prices that must be set such that the realized demand does not violate the capacity constraints of the electric grid [106]. This is necessary to ensure the safe and reliable operation of the grid, as violating the capacity constraints could have serious physical implications that could compromise the overall integrity of the system. In light of these considerations, it is evident that the resource demand of users must always satisfy the constraints of the system, even as they respond dynamically to pricing information.

To this end, in this chapter, we develop an iterative pricing algorithm to solve NUM problems with arbitrary convex and compact feasible sets, called safe pricing for NUM (SPNUM). We design our algorithm based solely on the realized demand in response to prices and communicate to the users only the prices for the resources at each iteration. Our contributions can be summarized as follows:

- We introduce a novel algorithm, the SPNUM, for solving NUM problems with arbitrary convex and compact feasible sets through pricing. Our algorithm iteratively designs prices and allows users the freedom of determining their own decision variable based on prices according to their own profit maximization problem (without imposing any iterative variable update rule on the users).

- We characterize a principled way to choose algorithm parameters to guarantee feasible primal iterates at all iterations. Furthermore, we prove that the static regret incurred by the feasible primal iterates produced by the SPNUM, i.e., the cumulative gap between the optimal objective value and the objective function evaluated at the primal iterates, up to time T is bounded by $\mathcal{O}(\log(T))$.

- We numerically evaluate our algorithm to support our theoretical findings and compare its performance to existing first-order distributed methods for NUM problems.

To the best of the authors' knowledge, no previous work has studied pricing algorithms for NUM problems on arbitrary convex feasible sets that are unknown to the users, even without consideration of safety. While primal-dual algorithms [107, 24, 108, 5] can handle non-separable arbitrary convex feasible sets, they rely on a primal update rule users need to follow in order to converge as opposed to maximizing their own profit based on observed prices. To this end, our contributions extend beyond safety, since SPNUM solves NUM problems on arbitrary convex feasible sets by iteratively designing prices and allowing the users to determine their own resource demand according to their own profit maximization problem.

The primal feasibility and the regret guarantees of the SPNUM result from a combination of two ingredients: 1) given prices and demand at a given instant, we apply a projected gradient method on a shrunk feasible set to get the next desired demand, and 2) we estimate the price response function of the users around the current prices and determine the next prices so that the induced demand is close to the desired demand. To ensure the algorithm behaves as a projected gradient method, the induced demand must be in the strict interior of the feasible set. The algorithm operates on a shrunk feasible set to account for the error between induced and desired demand, and gradually reduces shrinkage and step size to converge to the optimal solution.

Related work: Besides dual (sub)gradient methods, a few other branches of literature study a similar problem to ours. We highlight how those lines of work do not meet our particular design criteria and what differentiates the work presented in this chapter from them. Additional details on distributed optimization algorithms and their classifications can be found in the surveys [26, 109].

1. *Primal-dual methods*: Primal-dual methods tackle multi-user optimization problems

with arbitrary convex global constraints by applying a projected gradient descent/ascent on the primal/dual variables of the Lagrangian [107, 24, 108, 5]. The dual variables are updated using the aggregate resource demand information of the users and can be used for pricing of the resources. Therefore the update rule for the dual variables meets our design goals. However, the primal variables, i.e., the resource demand of the users, are updated by applying one step of gradient descent instead of solving for the profit-maximizing optimal demand in response to prices. Accordingly, these algorithms do not resemble the selfish profit-maximizing behavior of the users we adopt in this chapter.

2. *Projected gradient methods*: The main goal of the projected gradient methods is to maintain feasibility by projecting the primal variables on the feasible convex set after each update step. Scholars have extensively studied the convergence properties of the projected gradient methods under different assumptions [96, 110, 111]. On the other hand, the main challenge brought by our setup is that the primal variables are controlled solely by the users and cannot be manipulated (e.g., projected). Even though we can determine a feasible desired resource allocation by means of a projected gradient method, the prices that induce such resource demand are unknown due to the privacy of the utility functions, which brings unique challenges not addressed by the previous literature.

3. *Interior point methods*: Interior point methods are commonly used to solve inequality-constrained problems by using barrier functions to convert them into a sequence of equality-constrained problems, which are then solved using Newton's method [112]. While producing feasible iterates, the use of Newton's method requires the Hessian, which is often not available in practical applications, such as demand response without two-way communications. To address this limitation, previous works such as [113] and [114] have proposed feasible interior point methods that approximate the Hessian using first or second-order information exchange. However, these methods do not match the profit maximization rule we would like to preserve in

this chapter, which allows users to freely determine their resource consumption in response to posted prices. Closest to our setup and design goals in this chapter would be [115, 116], where separable optimization problems with linear constraints are considered. While [115] proposes a Newton-like dual update that approximates the Hessian using first-order information, only the asymptotic convergence of the algorithm is proven and the feasibility of primal iterates is not guaranteed. [116] proposes an interior point method using Lagrangian dual decomposition with theoretical guarantees, but requires the exact Hessian for dual updates.

Organization: The remainder of this chapter is organized as follows. In Section 3.2, we formalize the problem setup. In Section 3.3, we describe the SPNUM (Algorithm 5) and in Section 3.4, we prove its feasibility and regret guarantees. In Section 3.5, we provide a numerical study demonstrating the efficacy of the SDGM.

Notation and Basic Definitions: We denote the set of real numbers by $\mathbb{R}$ and the set of non-negative real numbers by $\mathbb{R}_+$. For vectors, $\|\cdot\|$ denotes the standard Euclidean norm and $\|\cdot\|_p$ denotes the p-norm. For matrices, $\|\cdot\|$ denotes the matrix norm. Given a positive integer $n > 0$, $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$. For two vectors $x, y \in \mathbb{R}^d$, $\langle x, y\rangle$ denotes the inner product of x and y. Given a vector $x = [x_1^\top,\ x_2^\top,\ \ldots,\ x_n^\top]^\top \in \mathbb{R}^d$, $x_i \in \mathbb{R}^{d_i}$ denotes the i'th block of x. For a matrix $A \in \mathbb{R}^{m\times n}$, A_j denotes the j'th row of A, $A_{:,j}$ denotes the j'th column of A. Given a matrix $A \in \mathbb{R}^{m\times m}$, $\mathrm{diag}(A) \in \mathbb{R}^m$ is the vector of the diagonals of A, $\kappa(A)$ is the condition number of A, and $\sigma_{\min}(A)$/$\sigma_{\max}(A)$ are the minimum/maximum singular values of A. Given a function $f : \mathcal{X} \subseteq \mathbb{R}^d \to \mathbb{R}$, ∇f denotes the gradient of f, $\nabla^k f$ denotes the k'th order gradient of f, and $\mathrm{dom} f$ denotes the domain $\mathcal{X}$ of f. Given two vectors $x, y \in \mathbb{R}^m$, $x \leq y$ implies element-wise inequality. Given a set $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{X}^{\text{int}}$ denotes the interior of $\mathcal{X}$. Given a convex and compact set $\mathcal{X} \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, $\Pi_{\mathcal{X}}(x)$ denotes the Euclidian projection of x onto $\mathcal{X}$. We denote the closed and the open Euclidean ball with radius r centered at origin as $\bar{\mathcal{B}}(r)$ and $\mathcal{B}(r)$, respectively. I_d denotes the identity matrix of size

d, $\mathbf{1}_d$ denotes the vector of all 1's with dimension d, and e_i denotes the unit vector with 1 in i'th dimension and 0 everywhere else.

Definition 3.1.1 *A differentiable function $f(\cdot)$ is said to be **μ-strongly concave** over the domain $\mathcal{X}$ if there exists $\mu > 0$ such that*

$$\langle \nabla f(x_2) - \nabla f(x_1), x_1 - x_2 \rangle \geq \mu \|x_1 - x_2\|^2 \tag{3.1}$$

holds for all $x_1, x_2 \in \mathcal{X}$.

Definition 3.1.2 *A differentiable function $f(\cdot)$ is said to be **L-smooth** over the domain $\mathcal{X}$ if there exists $L > 0$ such that*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L \|x_1 - x_2\| \tag{3.2}$$

holds for all $x_1, x_2 \in \mathcal{X}$.

Definition 3.1.3 *A function $f(\cdot)$ is said to be **M-Lipschitz continuous** over the domain $\mathcal{X}$ if there exists $M > 0$ such that*

$$\|f(x_1) - f(x_2)\| \leq M \|x_1 - x_2\| \tag{3.3}$$

holds for all $x_1, x_2 \in \mathcal{X}$.

3.2 Problem Setup

We study the standard NUM problem [18], where the goal is to allocate resources to n users subject to a set of coupling constraints such that the total utility of the users is maximized. It

can be formulated as the following optimization problem:

$$\max_{x \in \text{dom} f \subseteq \mathbb{R}^d} f(x) = \sum_{i=1}^{n} f_i(x_i) \tag{3.4a}$$

$$\text{s.t. } x \in \mathcal{X}, \tag{3.4b}$$

where $f_i(\cdot)$ is the concave utility function of user i that depends on the d_i-dimensional vector of resource consumption, denoted by $x_i \in \text{dom} f_i \subseteq \mathbb{R}^{d_i}$, and $\mathcal{X} \subset \mathbb{R}^d$ is the convex and compact set of feasible resource allocations. We also have $\sum_{i\in[n]} d_i = d$, $\text{dom} f = \prod_{i\in[n]} \text{dom} f_i$, and define $\bar{d} = \max_{i\in[n]} d_i$.

For all users $i \in [n]$, we define the set $\mathcal{X}_i = \{x_i \in \mathbb{R}^{d_i} : \exists x \in \mathcal{X} \text{ s.t. } x_i \text{ is the } i\text{'th block of } x\}$ as the set of values that user i's resource demand vector can take in the aggregate feasible set $\mathcal{X}$. Note that since $\mathcal{X}$ is convex and compact, $\mathcal{X}_i$ is convex and compact, $\forall i \in [n]$. Furthermore, if $x \in \mathcal{X}$, then $x_i \in \mathcal{X}_i$ and if $x \in \mathcal{X}^{\text{int}}$, then $x_i \in \mathcal{X}_i^{\text{int}}$ hold by definition. We make the following assumptions on the feasible set $\mathcal{X}$, and on the utility functions over $\mathcal{X}_i$, $\forall i \in [n]$.

Assumption 3.2.1 *The feasible set $\mathcal{X}$ is a subset of* $\text{dom} f$*, i.e., $\mathcal{X} \subseteq \text{dom} f$. The diameter of the feasible set $\mathcal{X}$ is bounded by R, i.e., $\|x - y\| \leq R$, $\forall x, y \in \mathcal{X}$. There exists a vector $\tilde{x}$ in the interior of $\mathcal{X}$ such that $\tilde{x} \in \mathcal{X}^{\text{int}}$.*

Assumption 3.2.2 *For all $i \in [n]$, the utility function $f_i(\cdot)$ is μ-strongly concave, L-smooth, M-Lipschitz continuous, and has β-smooth gradient over $\mathcal{X}_i$.*

Example 1 (Utility function) *For instance, take $f_i(x_i) = f_\alpha(x_i)$ to be an α-fair utility function (see [117]) and let $\mathcal{X}_i = [\underline{x}_i, \bar{x}_i]$ with $\underline{x}_i > 0$. We have that $\nabla f_i(x_i) \leq 1/\underline{x}_i^\alpha$, $-\alpha/\underline{x}_i^{\alpha+1} \leq \nabla^2 f_i^(x_i) \leq -\alpha/\bar{x}_i^{\alpha+1}$, and $\alpha(\alpha+1)/\bar{x}_i^{\alpha+2} \leq \nabla^3 f_i(x_i) \leq \alpha(\alpha+1)/\underline{x}_i^{\alpha+2}$, $\forall x \in \mathcal{X}_i$. Therefore, $f_i(x_i)$ is $\alpha/\bar{x}_i^{\alpha+1}$-strongly concave, $\alpha/\underline{x}_i^{\alpha+1}$-smooth, and $1/\underline{x}_i^\alpha$-Lipschitz continuous, and has $\alpha(\alpha+1)/\underline{x}_i^{\alpha+2}$-smooth gradient over $\mathcal{X}_i$.*

Under Assumption 3.2.2, the objective function (3.4a) is strongly concave with coefficient μ. Accordingly, the convex optimization problem (3.4) has a unique solution denoted by $x^\star$ and an optimal objective value denoted by $f^\star$.

Since $f_i(\cdot)$ are private to the users, (3.4) cannot be solved centrally. Therefore, distributed optimization methods based on the dual decomposition framework have been proposed in the literature (e.g., [23] for the case when $\mathcal{X}$ is a polytope) in order to incentivize selfish users with private utility functions to follow the optimal global solution. The common high-level idea is to divide the main problem into subproblems that can be solved by the individual users upon observing a pricing signal, and iteratively design prices $\{p^0, p^1, \dots\}$ to converge to the optimal resource allocation vector $x^\star$. In this framework, upon observing a price $p_i \in \mathbb{R}^{d_i}$, each user $i \in [n]$ determines their own decision variable according to their own profit maximization problem:

$$g_i(p_i) = \underset{x_i \in \mathrm{dom} f_i}{\arg\max} f_i(x_i) - \langle p_i, x_i \rangle. \tag{3.5}$$

We call $g_i(\cdot)$ the price response function of user i and let $g(p) = [g_1(p_1)^\top,\ g_2(p_2)^\top, \dots,\ g_n(p_n)^\top]$ be the concatenated vector of price responses given a price vector $p \in \mathbb{R}^d$.

In the next section, we propose an algorithm to iteratively design $p^t,\ \forall t \geq 1$, that produce feasible primal solutions, i.e., $x^t \in \mathcal{X},\ \forall t \geq 1$, where $x_i^t = g_i(p_i^t)$ is determined by user i through (3.5). In addition, the algorithm should produce primal iterates that result in a sublinear static regret per user, which is measured by

$$R(T) = \frac{1}{n} \sum_{t=1}^{T} f^\star - f(x^t). \tag{3.6}$$

It is worthwhile to highlight that even without the safety criterion, the literature on distributed optimization methods does not provide a distributed solution based on pricing to

(3.4) with any type of convergence guarantees. Existing works in the literature 1) utilize a pricing algorithm based on the dual decomposition framework but consider linear constraints [99, 100, 101, 102, 9] or non-linear and separable constraints known by the users [103, 104, 105], or 2) solve the Lagrangian dual problem by primal-dual methods [107, 24, 108, 5], which restrict the users to follow a primal update method that cannot be enforced in the setting where users only care about maximizing their own profit dictated by (3.5). Therefore, a pricing algorithm that induces a sequence of primal iterates converging to the optimal solution of (3.4) with general convex and compact feasible sets $\mathcal{X}$ is novel in the distributed optimization literature.

Additionally, we note that the definition of regret in (3.6) quantifies the difference between the efficiencies of the optimal resource allocation and the proposed algorithm up to time T. When the primal iterates $\{x^t\}_{t\in[T]}$ are in the feasible set $\mathcal{X}$, users' resource demand can actually be realized through the posted prices without waiting for the convergence of the algorithm, and therefore regret is a meaningful measure. On the other hand, although the above sum is computable for many of the existing works mentioned earlier (e.g., [100, 101] with linear constraints), they do not guarantee feasible primal iterates but only establish bounds on the amount of constraint violation at a given iteration t. Therefore, solutions are only realizable after convergence to a near-feasible point for resource allocation systems with safety-critical constraints. As such, they can be viewed as complex negotiations with users over what their potential demand would be in response to different prices in order to converge to the optimal price, which renders regret a less meaningful measure. By incorporating primal feasibility into our design goals, we aim to continually allocate resources to the users through posted prices *during the iterative optimization process* and measure the overall efficiency of this process through regret.

3.3 Safe Pricing Algorithm for NUM

In this section, we describe the price update algorithm we propose, called Safe Pricing for NUM (SPNUM), that produces feasible primal iterates satisfying a sublinear regret. To do so, we will use some definitions and results from [6] regarding the geometric properties of convex and compact sets. While the primary focus of [6] centers on a linear stochastic bandit setup that bears little resemblance to the NUM setup under study, the definitions of the shrunk set outlined in the former are applicable to the present context as well.

3.3.1 Geometric Properties of the Feasible Set

The main ingredient that ensures the safety of SPNUM is that it operates on a shrunk feasible set, which is formally defined as follows:

Definition 3.3.1 *For a compact set $\mathcal{X} \subset \mathbb{R}^d$ and a positive scalar $\Delta \in \mathbb{R}_+$, we define the **shrunk version** of $\mathcal{X}$ as $\mathcal{X}_\Delta := \{x \in \mathcal{X} : x + v \in \mathcal{X}, \forall v \in \bar{\mathcal{B}}(\Delta)\}$.*

Example 2 *(Shrunk polytope) Let $A \in \mathbb{R}^{m\times d}$ and $\mathcal{X} = \{x \in \mathbb{R}^d : Ax \leq c\}$ be a polytope. The shrunk version of $\mathcal{X}$ is defined as $\mathcal{X}_\Delta = \{x \in \mathbb{R}^d : A_j^\top x \leq c_j - \Delta\|A_j\|,\ j \in [m]\}$.*

Remark 3.3.1 *If $\mathcal{X}$ is convex and compact, then $\mathcal{X}_\Delta$ is also convex and compact.*[1]

Given the above definition of the shrunk version of a set, one can consider the maximum shrinkage that a set can withstand while still being nonempty. We introduce the *maximum shrinkage of a set* in the following definition.

Definition 3.3.2 *For a compact set $\mathcal{X} \subset \mathbb{R}^d$, we define the **maximum shrinkage** of $\mathcal{X}$, as $H_\mathcal{X} := \sup\{\Delta : \mathcal{X}_\Delta \neq \emptyset\}$.*

[1] We can equivalently define $\mathcal{X}_\Delta$ using Minkowski subtraction. The Minkowski subtraction of sets $A, B \subseteq \mathbb{R}^d$ is defined as $A \ominus B := \{a - b : a \in A, b \in B\}$, or equivalently, $A \ominus B = \bigcap_{b\in B}(A - b)$. Therefore, $\mathcal{X}_\Delta = \mathcal{X} \ominus \mathcal{B}(\Delta)$ is an intersection of convex and closed sets and hence is convex and closed [118, Section 3.1]. By Definition 3.3.1, $\mathcal{X}_\Delta$ is a subset of $\mathcal{X}$, and therefore bounded. A closed and bounded convex set is convex and compact.

3.3.2 Description of the Algorithm

Algorithm 5: Safe Pricing for NUM

1: **Input:** p^0, Δ^t, γ^t, η^t.
2: *(Initialization stage)*:
3: Each user $i \in [n]$ receives p_i^0 and $p_i^{-t} = p_i^0 + \eta^0 e_{1+\mathrm{mod}(t,d_i)}$, $\forall t \in [d_i]$ and solves

$$x_i^t = g_i(p_i^t),\ t = -d_i, -d_i+1, \ldots, 0. \tag{3.7}$$

4: For all $i \in [n]$, estimate the Jacobian of g_i as:

$$\hat{\nabla} g_i^0 = \left[\frac{x_i^{-d_i} - x_i^0}{\eta^0}, \ldots, \frac{x_i^{-1} - x_i^0}{\eta^0} \right] \tag{3.8}$$

5: **for** $t = 0, 1, \ldots$ **do**
6: *(Update stage)*
7: Compute $\hat{x}^{t+1} = \Pi_{\mathcal{X}_{\Delta^t}}(x^t + \gamma^t p^t)$.
8: Set $p_i^{t+1} = p_i^t + [\hat{\nabla} g_i^t]^{-1}(\hat{x}_i^{t+1} - x_i^t)$, for all $i \in [n]$.
9: Each user $i \in [n]$ receives p_i^{t+1} and solves $x_i^{t+1} = g_i(p_i^{t+1})$
10: *(Sampling stage)*
11: Each user $i \in [n]$ receives $p_i^{t+1,s} = p_i^{t+1} + \eta^{t+1} e_{1+\mathrm{mod}(t,d_i)}$ and solves $x_i^{t+1,s} = g_i(p_i^{t+1,s})$
12: For each user $i \in [n]$

$$[\hat{\nabla} g_i^t]_{:,1+\mathrm{mod}(t,d_i)} \leftarrow (x_i^{t+1,s} - x_i^{t+1})/\eta^{t+1} \tag{3.9}$$

$$\hat{\nabla} g_i^{t+1} = \hat{\nabla} g_i^t \tag{3.10}$$

13: **end for**

The proposed method, called safe pricing for NUM (SPNUM) and outlined in Algorithm 5, consists of two stages at each iteration: 1) update stage (Step 6) and 2) sampling stage (Step 10). The update stage proceeds similarly to a projected gradient method on the primal iterates while designing prices that induce realized iterates close to a *desired iterate*. The sampling stage estimates the Jacobians of the price response functions of the users, which are used during the update stage.

In the update stage, the algorithm first determines a desired next iterate $\hat{x}^{t+1}$ in Step 7.

However, because the primal variables are not directly controllable, prices that induce x^{t+1} that is close to $\hat{x}^{t+1}$ have to be determined at Step 8. Accordingly, at the heart of the update stage lie two key steps:

1. At iteration t, the central coordinator observes x^t and determines the next desired iterate $\hat{x}^{t+1}$ by means of a projected gradient ascent step in Step 7. This is because if $x^t \in \mathcal{X}^{\text{int}}$, then $x_i \in \mathcal{X}_i^{\text{int}}$, which implies that $p_i^t = \nabla f_i(x^t)$ by Assumption 3.2.2 and the first order optimality condition for (3.5). Therefore, $p^t = \nabla f(x^t)$. In addition, projection is performed onto a shrunk set $\mathcal{X}_{\Delta^t}$, where Δ^t controls the amount of shrinkage at time t. This is the key ingredient to ensure the safety of the algorithm because the uncertainty in the price response functions will cause the actual induced iterate x^{t+1} in response to the price vector p^{t+1} to deviate from the desired iterate $\hat{x}^{t+1}$. By adding this safety margin to the constraint, we can ensure safety if $\|x^{t+1} - \hat{x}^{t+1}\| \in \bar{\mathcal{B}}(\Delta^t)$. Finally, by utilizing a diminishing safety margin sequence $\{\Delta^t\}_{t\geq 0}$, we can ensure convergence to the optimal solution of (3.4).

2. Once the desired next iterate $\hat{x}^{t+1}$ is determined, the central coordinator has to determine p_i^{t+1} that would ideally induce $\hat{x}_i^{t+1}$, $\forall i \in [n]$. However, the price response function is unknown to the central coordinator, and therefore an exact solution is not possible. Instead, the central coordinator makes a linear approximation of the price response function using the Jacobian estimate of g_i, $\forall i \in [n]$. In particular, the central coordinator keeps an estimate of the Jacobian denoted by $\hat{\nabla} g_i^t$ initialized in Steps 3 and 4 of the algorithm, which is constructed by varying the price vector along each dimension and estimating the gradient using the difference equation. This results in the following linear approximation of the price response function around p_i^t:

$$\hat{g}_i(p) = x_i^t + \hat{\nabla} g_i^t (p - p_i^t). \tag{3.11}$$

By setting $p = p_i^{t+1}$, $\hat{g}_i(p_i^{t+1}) = \hat{x}^{t+1}$, and rearranging, we get the price update rule in Step 8.

This requires that the $\hat{\nabla} g_i^t$ is an invertible matrix, which will be proven in Section 3.4.

After determining p^{t+1} and x^{t+1}, the algorithm proceeds to the sampling stage to update the Jacobian estimates. To achieve this, the central coordinator varies the price vector p_i^{t+1} along the dimension $1 + \mathrm{mod}(t, d_i)$ in Step 11 for user $i \in [n]$, resulting in a sampling price of $p_i^{t+1,s}$. The response is observed and denoted as $x_i^{t+1,s}$. The difference between $x_i^{t+1,s}$ and x_i^{t+1} divided by the amount of price variation serves as an estimate of the gradient of the price response function along the $1+\mathrm{mod}(t, d_i)$'th principal axis, which becomes the $1+\mathrm{mod}(t, d_i)$'th column of the Jacobian estimate $\hat{\nabla} g_i^{t+1}$ in Step 12. It is worthwhile to highlight that for a user i, the error between $\hat{x}_i^{t+1}$ and x_i^{t+1} has two sources: 1) the difference between the estimated Jacobian and the actual Jacobian, i.e., $\hat{\nabla} g_i^t - \nabla g_i(p_i^t)$, and 2) the high order terms not captured by the linear approximation, i.e., $R_1 = g_i(p^t) - \nabla g_i(p^t)(p - p^t)$.

It is necessary that there exists an initial price vector p^0 such that the demand vectors in response to the initial sampling prices in (3.7) are in $\mathcal{X}^{\text{int}}$ so that the algorithm can proceed as described above. Since this has to hold before getting any feedback from the users, we make the following assumption:

Assumption 3.3.1 *There exists a known price vector p^0 such that $g(p^0) \in \mathcal{X}^{\text{int}}$ and for all $i \in [n]$, $x_i^{-d_i} \in \mathcal{X}_i^{\text{int}}$.*

The above assumption guarantees that the initial demand vectors in (3.7) are in $\mathcal{X}_i^{\text{int}}$, $\forall i \in [n]$ and therefore the initial Jacobian estimation is meaningful.

Remark 3.3.2 *One way to satisfy Assumption 3.3.1 is to choose η^0 such that $\mathcal{X}_{\frac{\sqrt{n}\eta^0}{\mu}}$ is non-empty and p^0 such that $g(p^0) \in \mathcal{X}_{\frac{\sqrt{n}\eta^0}{\mu}}$, which is proven in Appendix B.8.*

In the next section, we characterize a principled way to choose parameters Δ^t, γ^t, and η^t in order to produce feasible primal iterates. Additionally, we prove that the regret incurred by the iterates produced by Algorithm 5 is $\mathcal{O}(\log(T))$ after T iterations, and the last iterate converges to the optimal solution at the rate $\mathcal{O}(\log(T)/T)$.

3.4 Feasibility and Regret Analysis

In order to prove the safety and the regret guarantees of our algorithm, we will need to bound the distance between a point in $x \in \mathcal{X}$ and its projection onto the shrunk set $\Pi_{\mathcal{X}_\Delta}(x)$. The following definition from [6] formalizes this notion called the *sharpness of a set*, which is defined as the maximum distance from any point in a set to the projection of it onto the shrunk version of that set.

Definition 3.4.1 *For a convex and compact set $\mathcal{X} \subset \mathbb{R}^d$, we define the **sharpness** of $\mathcal{X}$ as*

$$\mathrm{Sharp}_{\mathcal{X}}(\Delta) := \sup_{x \in \mathcal{X}} \|\Pi_{\mathcal{X}_\Delta}(x) - x\|, \tag{3.12}$$

for all non-negative Δ such that $\mathcal{X}_\Delta$ is nonempty.

The following proposition establishes a bound on the sharpness of convex and compact sets as a linear function of Δ:

Proposition 3.4.1 *[6, Corollary 11] For a convex, compact set $\mathcal{X} \subset \mathbb{R}^d$ with non-empty interior, we have that $\mathrm{Sharp}_{\mathcal{X}}(\Delta) \leq \Gamma_{\mathcal{X}} \Delta$ where $\Gamma_{\mathcal{X}} \geq 1$ is a constant that depends only on the geometry and the dimension of $\mathcal{X}$.*

Example 3 (Sharpness of a polytope [6]) *Let $\mathcal{X} = \{x \in \mathbb{R}^d : Ax \leq c\}$ be a polytope with a nonempty interior. Define $\mathcal{I}_A$ to refer to the collection of all sets of d indices such that for each $\{i_1, i_2, ..., i_d\} \in \mathcal{I}_A$ the vectors $A_{i_1}, A_{i_2}, ..., A_{i_d}$ are linearly independent. For each $\ell \in \mathcal{I}_A$ where $\ell = \{i_1, i_2, ..., i_d\}$, we define $A^\ell = [A_{i_1}^\top \; A_{i_2}^\top \; ... \; A_{i_d}^\top]^\top$. We have that $\mathrm{Sharp}_{\mathcal{X}}(\Delta) \leq \sqrt{d} K_{\mathcal{X}} \Delta$, where $K_{\mathcal{X}} := \max_{\ell \in \mathcal{I}_A} \kappa(A^\ell)$.*

Example 4 (Sharpness of a ball in $\mathbb{R}^d$) *Let $\mathcal{X} = \{x \in \mathbb{R}^d : (x - x_0)^\top (x - x_0) \leq r^2\}$ be a ball in $\mathbb{R}^d$ with radius r centered at x_0. We have that $\mathrm{Sharp}_{\mathcal{X}}(\Delta) = \Delta$.*

Although we do not specify a closed-form expression of $\Gamma_{\mathcal{X}}$ for a general convex and compact set $\mathcal{X}$, it relates to the sharpness of polytopes that are contained in $\mathcal{X}$, which have closed-form bounds as given by Example 3. We refer the reader to [6] (Proposition 10) for a detailed discussion.

The next lemma characterizes the regularity properties of $g_i(p_i)$ over the set of prices that induce a resource demand in $\mathcal{X}_i^{\text{int}}$ for a user $i \in [n]$. This property is crucial for our analysis and for the feasibility of the algorithm, as we need to show that the inverse of the matrix $\hat{\nabla} g_i^t$ for the price update rule in Step 8 is a valid operation.

Lemma 3.4.1 *Let $\mathcal{P}_i = \{p_i \in \mathbb{R}^{d_i} : g_i(p_i) \in \mathcal{X}_i^{\text{int}}\}$ be the set of prices that induce a resource demand in $\mathcal{X}_i^{\text{int}}$ for a user $i \in [n]$. Over $\mathcal{P}_i$, $g_i(p_i)$ is bijective, $1/\mu$-Lipschitz continuous, and β/μ^3-smooth. Accordingly, $g_i(p_i)$ is invertible and $\nabla g_i(p_i) = [\nabla^2 f_i(g_i(p_i))]^{-1}$.*

The proof of Lemma 3.4.1 can be found in Appendix B.1. Lemma 3.4.1 establishes that the true Jacobian of the price response function for user i is invertible because it corresponds to the inverse of the Hessian of the strongly concave utility function of user i. However, this does not imply that the estimated Jacobian $\hat{\nabla} g_i^t$ is invertible since it is constructed by finite difference gradient approximation. The next lemma states that the estimated Jacobian $\hat{\nabla} g_i^t$ is close enough to $\nabla g_i(p^t)$, which allows us to bound the minimum singular value of it and therefore guarantees invertibility with the appropriate choice of algorithm parameters.

Lemma 3.4.2 *Let $\gamma^t = 1/(\mu(t+\tau))$, $\Delta^t = \Delta/(t+\tau)^2$, and $\eta^t = \mu\Delta^{t-1}/(4\sqrt{n})$ for some $\Delta > 0$ and*

$$\tau = \max\left\{2, 2\bar{d} - 1, 1 + \frac{2\mu\Delta\Gamma_{\mathcal{X}}}{M\sqrt{n}}, \sqrt{\frac{\Delta}{H_{\mathcal{X}}}}, \frac{L\beta M\sqrt{\bar{d}}\left(\mu + 32L\Gamma_{\mathcal{X}}\sqrt{n}(\bar{d}-1)\right)}{2\mu^4\Gamma_{\mathcal{X}}}\right\}. \tag{3.13}$$

Suppose that at iteration t, $x^k \in \mathcal{X}^{\text{int}}_{\frac{\eta^k\sqrt{n}}{\mu}}$, $\forall k \in [\max\{t-\bar{d}+1, 0\}, t]$. Then, the following holds

for all $i \in [n]$:

$$\|\hat{\nabla} g_i^t - \nabla g_i(p_i^t)\| \leq e_i^t, \tag{3.14}$$

where

$$e_i^t = \frac{2\beta\sqrt{d_i}}{\mu^3}\left(\eta^t + 2L(d_i - 1)(M\sqrt{n}\gamma^t + 2\Delta^t\Gamma_{\mathcal{X}})\right) \leq \frac{1}{2L}. \tag{3.15}$$

Accordingly, $\sigma_{\min}(\hat{\nabla} g_i^t) \geq \frac{1}{2L}$ and therefore $\hat{\nabla} g_i^t$ is invertible.

The proof of Lemma 3.4.2 can be found in Appendix B.2. Lemma 3.4.2 characterizes a principled way to choose the algorithm parameters with respect to a free parameter Δ in order to bound the difference between $\hat{\nabla} g_i^t$ and $\nabla g_i(p^t)$. In the following subsections, we will first characterize the choice of Δ that guarantees primal feasibility at all iterations and then prove the regret and convergence guarantees of Algorithm 5 under this choice of parameters.

3.4.1 Feasibility Analysis

The following proposition characterizes the choice of the parameters Δ^t, γ^t, and η^t to ensure feasible primal iterates:

Proposition 3.4.2 *Let $\gamma^t = 1/(\mu(t+\tau))$ and $\Delta^t = \Delta/(t+\tau)^2$, $\eta^t = \mu\Delta^{t-1}/(4\sqrt{n})$, where τ is given by* (3.13) *and*

$$\Delta = \beta L M n^{3/2}(6L + \sqrt{d}(\mu/\sqrt{n} + 32L(\bar{d} - 1)))/\mu^5. \tag{3.16}$$

Then for all $t \geq 0$, $\|\hat{x}^{t+1} - x^{t+1}\| < 3\Delta^t/4$ and $\|x^{t+1} - x^{t+1,s}\| \leq \Delta^t/4$. Accordingly, for all $t \geq 0$, the iterates x^t and $x^{t,s}$ produced by Algorithm 5 are feasible and in the strict interior of the feasible set, i.e., $x^t \in \mathcal{X}^{\text{int}}_{\frac{\eta^t\sqrt{n}}{\mu}}$ and $x^{t,s} \in \mathcal{X}^{\text{int}}$, $\forall t \geq 1$.

The proof of Proposition 3.4.2 can be found in Appendix B.3. Given that under Proposition 3.4.2, x^t for all $t \geq 1$ are feasible and therefore implementable, the static regret (3.6) is a valid choice of performance metric. Next, we prove that the regret of Algorithm 5 is $\mathcal{O}(\log(T))$ and the primal variables converge to the optimal solution at the rate $\mathcal{O}(\log(T)/T)$.

3.4.2 Regret and Convergence Analysis

As our algorithm alternates between executing one update and one sampling stage, after T iterations it will have executed $T/2$ update stages and $T/2$ sampling stages. In this case, the regret per user is fairly calculated as:

$$R(T) = \frac{1}{n}\sum_{t=1}^{T/2}(f(x^\star) - f(x^t) + f(x^\star) - f(x^{t,s})). \tag{3.17}$$

The following theorem establishes an upper bound on the regret incurred by the primal iterates produced by Algorithm 5, and the squared distance between last iterate $x^{T/2}$ and the optimum solution $x^\star$:

Theorem 3.4.1 *Let p^0, γ^t, Δ^t, and η^t be chosen as in Proposition 3.4.2. Then for all $t \geq 0$, the iterates produced by Algorithm 5 are feasible. Furthermore, the regret $R(T)$ for $T \geq 2$ satisfies*

$$R(T) \leq \mathcal{O}(\log(T)(1 + \Delta\Gamma_{\mathcal{X}}/n)), \tag{3.18}$$

where $\mathcal{O}(\cdot)$ hides other constants. In addition, the last primal iterate $x^{T/2}$ satisfies

$$\|x^{T/2} - x^\star\|^2 \leq \mathcal{O}(\log(T)/T). \tag{3.19}$$

Proof outline: Since the algorithm proceeds similarly to a projected gradient method, the proof is similar to that of a projected gradient ascent for strongly concave functions. We have an

additional error term due to $\|x^{t+1} - \hat{x}^{t+1}\|$, which is $\mathcal{O}(\Delta^t)$. The error term impacts the result as $\mathcal{O}(\sum_{t=1}^{T/2} \Delta^t/\gamma^t)$, which results in an additive $\mathcal{O}(\log(T)\Delta\Gamma_{\mathcal{X}}/n)$ term.

The complete proof of Theorem 3.4.1 and the explicit constants of (3.18) can be found in Appendix B.6. According to Theorem 3.4.1, Algorithm 5 produces feasible solutions that achieve a sublinear regret of $\mathcal{O}(\log(T))$. Furthermore, the primal variables induced by the prices converge to the optimal solution at the rate $\mathcal{O}(\log(T)/T)$.

Remark 3.4.1 *When* $d_i = 1,\ \forall i \in [n]$, $\Delta = \mathcal{O}(\beta n^{3/2})$ *and* $R(T) = \mathcal{O}(\log(T)(1 + \sqrt{n}\beta\Gamma_{\mathcal{X}}))$.

In the next section, we numerically demonstrate our theoretical results about the primal variables induced by Algorithm 5 and compare its performance to existing pricing algorithms.

3.5 Numerical Studies

In this section, we demonstrate the efficacy of SPNUM via three numerical studies: 1) a benchmarking study to compare SPNUM's convergence and feasibility performance to existing pricing methods that solve the NUM problem, 2) a toy NUM problem with a non-linear feasible set to demonstrate the success of SPNUM on non-linear feasible sets, and 3) a parameter study to demonstrate how the regret depends on the second order smoothness parameter β, sharpness parameter $\Gamma_{\mathcal{X}}$, and the number of users n.

3.5.1 Benchmarking Study

In this study, our aim is to compare the safety and convergence performance of SPNUM to the existing algorithms on feasible sets characterized by linear inequalities, i.e., $\mathcal{X} = \{x \in \mathbb{R}^d : Ax \leq c\}$. We compare SPNUM to DG [102], which can achieve a linear convergence rate, and SDGM [9], which can provide safety when A is a binary matrix.

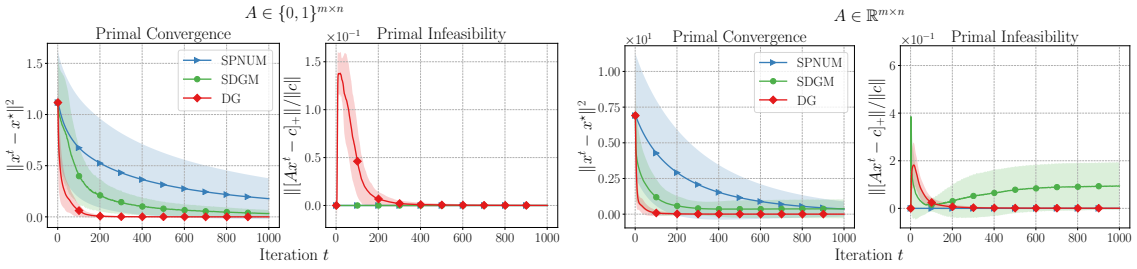


Figure 3.1: Results for the benchmarking study. In all plots, SPNUM is shown in blue, SDGM in green, and DG in red. The shaded areas correspond to one standard deviation. In (a), we plot the convergence of the primal variables measured by $\|x^t - x^\star\|^2$ (left) and the infeasibility amount measured by $\|[Ax^t - c]_+\|/\|c\|$ (right) for all three algorithms when $A \in \{0,1\}^{m\times n}$ is a binary matrix. In (b), we plot the convergence of the primal variables measured by $\|x^t - x^\star\|^2$ (left) and the infeasibility amount measured by $\|[Ax^t - c]_+\|/\|c\|$ (right) for all three algorithms when $A \in \mathbb{R}^{m\times n}$ is a real matrix.

We have implemented all algorithms on two types of A matrices: 1) A is a binary matrix and 2) A is a real matrix. For both cases, we randomly generated a collection of 50 networks with a random number of users n taking (integer) values in range $[5, 20]$, and a random number of constraints m taking values in the interval $[5, 10]$ (generated independently). Inspired by [102], for all users $i \in [n]$, we let the utility function be $f_i(x_i) = -0.5(x_i-3)^2 - x_i - \theta_i \log(1+e^{x_i})$, where θ_i is sampled uniformly from $[0, 1]$ for each network configuration (we shifted the quadratic term by 3 to ensure that the optimal solution is on the boundary of the feasible set). We set $\mathrm{dom} f_i = [0, 1]$ for all $i \in [n]$. For each network configuration, we first randomly generated a matrix $\hat{A}$ by sampling $m \times n$ Bernoulli random variables for the binary matrix case, and by sampling $m \times n$ random variables from the continuous uniform distribution in $[-1, 1]$ for the real matrix case. We then let $A = [\hat{A}^\top \; I_n]^\top$. For the binary case, we let $c = \mathbf{1}_{m+n}$, and for the real case, we let $c = [0.1\mathbf{1}_m^\top \; \mathbf{1}_n^\top]^\top$.[2]

We note that $\mathcal{X}_i \subseteq [0, 1], \; \forall i \in [n]$. Within $\mathcal{X}_i$, using bounds on θ_i and computing the derivatives of f_i, we get $M = 2$, $L = 5/4$, $\mu = 1$, $\beta = \sinh(1)/(2(1 + \cosh(1))^2) \approx 0.0909$.

[2]For SPNUM, we additionally include the constraints $x \geq 0$ in $\mathcal{X}$ to satisfy Assumption 3.2.1. For the other algorithms, this is not needed.

Finally, from Example 3 we have $\Gamma_{\mathcal{X}} \leq \sqrt{n}\kappa(A)$.

For each configuration, we initialized the dual variables and prices to induce $x_i^0 = \eta^0/\mu,\ \forall i \in [n]$, and ran all three algorithms for a horizon of $T = 1000$. We demonstrate the results for the binary matrix case and the real matrix case in Figure 3.1a and Figure 3.1b, respectively. In Figure 3.1a we observe that **1)** while DG converges the fastest, it is not safe, **2)** SDGM and SPNUM converge slower but are safe, and **3)** SDGM converges faster than SPNUM because it is designed specifically for this setting. On the other hand, in Figure 3.1b we observe that **1)** SDGM does not provide safety and convergence when A is a real matrix, as its assumptions do not hold anymore (note that the plot for $\|x^t - x^\star\|^2$ flattens for SDGM), **2)** SPNUM successfully provides safety and convergence.

3.5.2 SPNUM on Non-linear Feasible Set

This study aims to demonstrate numerically the regret and safety guarantees of SPNUM on a problem with a feasible set characterized by non-linear inequalities. We select the feasible set $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$ as the unit ball in $\mathbb{R}^d$ centered at the origin. At the beginning of each run, we sample the number of users n as an integer from the range $[5, 20]$ uniformly at random. For all $i \in [n]$, we let the utility function be $f_i(x_i) = -0.5(x_i - y_i)^2 - x_i - \theta_i \log(1 + e^{x_i})$, where θ_i is sampled uniformly from $[0, 1]$ and y_i is sampled uniformly from $[-2, 2]$ at random at the beginning of each run.

Noting that $\mathcal{X}_i = [-1, 1]$, using bounds on θ_i and y_i and computing the derivatives of f_i, we get $M = 4 + e/(1+e)$, $L = 5/4$, $\mu = 1$, $\beta = \sinh(1)/(2(1+\cosh(1))^2) \approx 0.0909$. Finally, from Example 4 we have $\Gamma_{\mathcal{X}} = 1$.

We initialize the prices to induce $x_i^0 = \eta^0/\mu,\ \forall i \in [n]$, and ran SPNUM 100 times for a horizon of $T = 50$. The results are illustrated in Figure 3.2. The figure shows that **1)** the regret of SPNUM grows as $\mathcal{O}(\log(t))$, **2)** SPNUM guarantees feasible iterates at all iterations, and **3)**

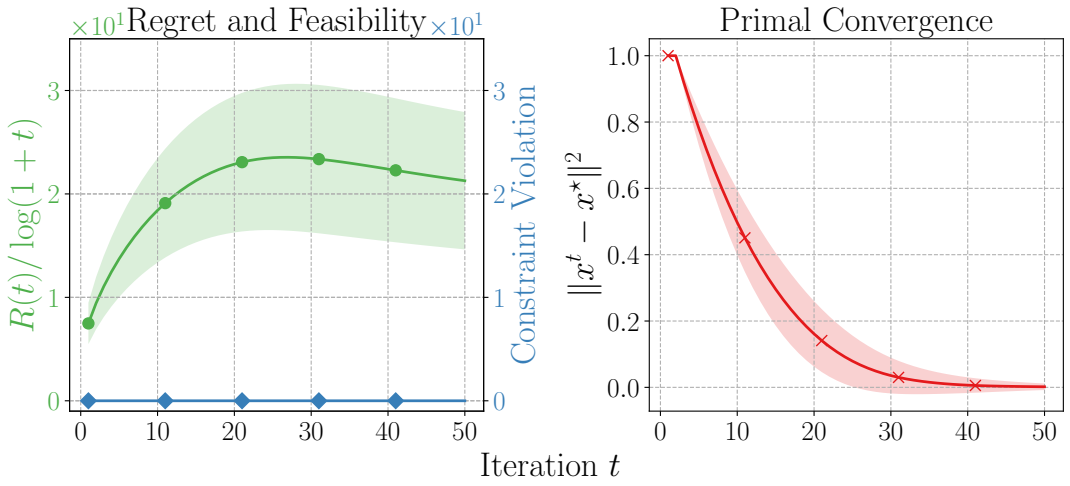


Figure 3.2: Results for the numerical study on SPNUM on non-linear feasible set. In the left figure, the regret divided by $\log(1+t)$ is plotted in green, and constraint violation is plotted in blue, where constraint violation is 0 if $x^t \in \mathcal{X}$ and 1 otherwise. In the right figure, we plot the primal convergence measured as $\|x^t - x^\star\|^2$. Shaded areas correspond to one standard deviation.

the primal iterates produced by SPNUM converge to the optimal solution.

3.5.3 Impact of Sharpness on Regret

In this study, our aim is to support our theoretical results about SPNUM with numerical examples. In particular, we study the impact of sharpness parameter $\Gamma_{\mathcal{X}}$ and the number of users n on regret through β. We set $d_i = 1$, in which case $R(T) = \mathcal{O}(\log(T)(1+\beta\sqrt{n}\Gamma_{\mathcal{X}}))$ as stated in Remark 3.4.1. For each user i, we set $f_i(x_i) = \theta_i(\cos(\omega(x-1))/\omega^2 - 10(x-2)^2 - x\sin(\omega)/\omega)$, where θ_i is sampled uniformly from $[1,2]$. This particular choice of f_i allows us to control β while keeping the other parameters constant by simply choosing ω. Using the bounds on θ_i and computing the derivatives of f_i, we get $M = 40$, $L = 42$, $\mu = 19$, and $\beta = 2\omega$.

In order to have control over the sharpness parameter $\Gamma_{\mathcal{X}}$, we study linear constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n : x \geq 0, Ax \leq c\}$, where $A_{ij} = (1-K)/(1+K(n-1))$ if $i \neq j$, and $A_{ii} = 1$. This choice of A allows us to parameterize the feasible set as a function of the condition number K, since $\kappa(A) = K$ and $\Gamma_{\mathcal{X}} = \sqrt{n}\kappa(A)$. Finally, since f_i is increasing over

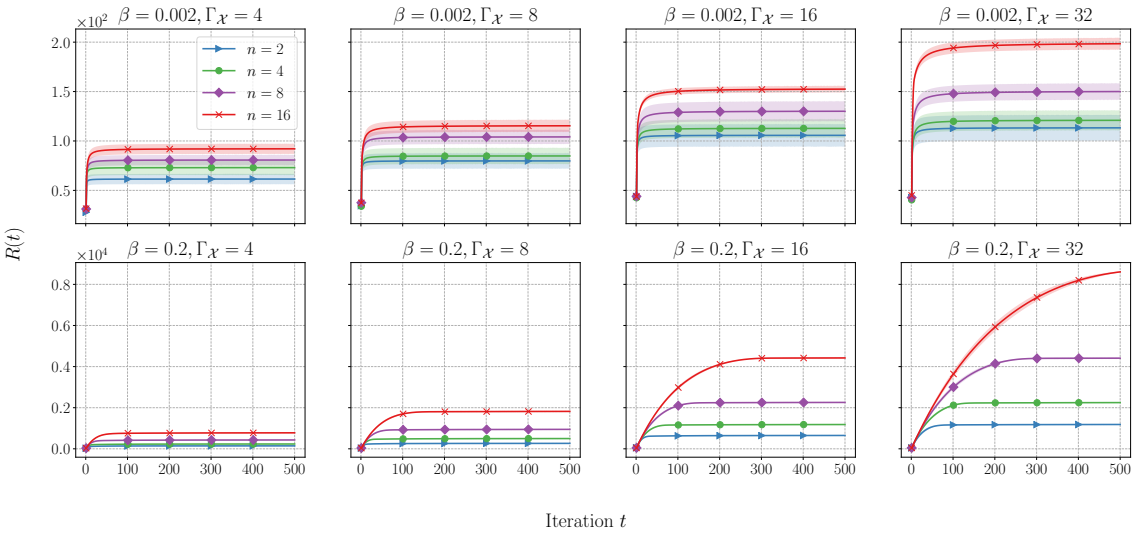


Figure 3.3: Results for the numerical study on the impact of sharpness on regret. The figures on each row share the same y-axis. The shaded areas correspond to one standard deviation. The title of each subfigure denotes the $(\beta, \Gamma_{\mathcal{X}})$ configuration, and the regret incurred by different values of n are plotted for each configuration. We observe that in the top row of figures, i.e., when β is small, both $\Gamma_{\mathcal{X}}$ and n have little effect on the regret (e.g., increasing $\Gamma_{\mathcal{X}}$ by 8 times only doubles the regret for all n). On the other hand, the bottom row of figures demonstrates that when β is larger, then $\Gamma_{\mathcal{X}}$ and n have a significant impact.

$\mathcal{X}_i$, the optimal solution is given by $x^\star = \mathbf{1}_n$.

For $n = \{2, 4, 8, 16\}$, $\omega = \{0.001, 0.1\}$, and $K = \{4/\sqrt{n}, 8/\sqrt{n}, 16/\sqrt{n}, 32/\sqrt{n}\}$, we randomly sampled 10 sets of $\{\theta_i\}_{i\in[n]}$, initialized p_i^0 so that $x_i^0 = \eta^0/\mu$, $\forall i \in [n]$, and ran SPNUM for a horizon of $T = 500$. Note that this corresponds to configurations of $\beta = \{0.002, 0.2\}$ and $\Gamma_{\mathcal{X}} = \{4, 8, 16, 32\}$. We plot the regret for each configuration in Figure 3.3. The results indicate that **1)** when β is small, $\Gamma_{\mathcal{X}}$ and n have little impact on the regret, and **2)** when β is large, regret grows with $\Gamma_{\mathcal{X}}$ and n as the term proportional to $\sqrt{n}\beta\Gamma_{\mathcal{X}}$ dominates.

3.6 Conclusion

In this chapter, we introduced a novel algorithm, called the safe pricing for NUM (SP-NUM), for solving resource allocation problems over networks with arbitrary convex and compact feasible sets in a distributed fashion. Our algorithm iteratively designs prices for resources and allows the users the determine their own resource demand in response to prices according to their own profit maximization problem. The prices produced by SPNUM ensure that the induced demand satisfies the constraints of the system during the optimization process, which promotes safety. This is done by: 1) shrinking the constraint set and applying a projected gradient method to the primal variables to determine the updated desired demand, and 2) determining the prices that would induce the desired demand by estimating the price response function of the users using the historical data. By carefully controlling the amount of shrinkage to account for the error in the estimated price response, we ensure the safety of the algorithm. In addition, we have proven that the regret incurred by the SPNUM is $\mathcal{O}(\log(T))$, and the primal variables converge to the optimal solution at the rate of $\mathcal{O}(\log(T)/T)$.

Chapter 4

Ride Pricing and Control Policies for Autonomous Urban Mobility Fleets

4.1 Introduction

The rapid evolution of enabling technologies for autonomous driving coupled with advancements in eco-friendly electric vehicles (EVs) has facilitated state-of-the-art transportation options for urban mobility. Owing to these developments in automation, it is possible for an autonomous-mobility-on-demand (AMoD) fleet of autonomous EVs to serve the society's transportation needs, with multiple companies now heavily investing in AMoD technology [119].

The introduction of autonomous vehicles for mobility-on-demand services provides an opportunity for better fleet management. Specifically, idle vehicles can be *rebalanced* throughout the network in order to prevent accumulating at certain locations and to serve induced demand at every location. Autonomous vehicles allow rebalancing to be performed centrally by a platform operator who observes the state of all the vehicles and the demand, rather than locally by individual drivers. Furthermore, EVs provide opportunities for cheap and environment-

friendly energy resources (e.g., solar energy). However, electricity supplies and prices differ among the network both geographically and temporally. As such, this diversity can be exploited for cheaper energy options when the fleet is operated by a platform operator that is aware of the electricity prices throughout the whole network. Moreover, a dynamic pricing scheme for rides is essential to maximize profits earned by serving the customers. Coupling an optimal fleet management policy with a dynamic pricing scheme allows the revenues to be maximized while reducing the rebalancing cost and the waiting time of the customers by adjusting the induced demand.

Building upon the aforementioned opportunities presented by electric AMoD systems, our study delves into static and dynamic control policies for these systems. In particular, we develop and underscore the advantages of real-time control strategies, supported by empirical performance assessments on real network and demand data in Section 4.2.

Additionally in Section 4.3, we extend our investigation to examine the impact of competition on critical factors within electric AMoD systems operated by profit-maximizing platform operators. This analysis encompasses ride prices, aggregate demand served, profits of the firms, and consumer surplus, providing a comprehensive assessment of the competitive landscape in this domain.

4.2 Dynamic Pricing and Fleet Management for Electric Autonomous Mobility on Demand Systems

In this section, we study joint pricing and fleet management control policies for electric AMoD systems. We consider a model that captures the opportunities and challenges of an AMoD fleet of EVs, and consists of complex state and action spaces. In particular, the platform operator has to consider the number of customers waiting to be served at each location (ride request queue lengths), the electricity prices, traffic conditions, and the states of the EVs (locations, battery energy levels) in order to make decisions. These decisions consist of pricing for rides for every origin-destination (OD) pair and routing/charging decision for every vehicle in the network. Upon taking an action, the state of the network undergoes through a stochastic transition due to the randomness in customer behaviour, electricity prices, and travel times.

We first adopt the common approach of network flow modeling to develop an optimal static pricing, routing, and charging policy that we use as a baseline in this section. However, flow-based solutions generate fractional flows which can not directly be implemented. Moreover, a static policy executes same actions independent of the network state and is oblivious to the stochastic events that occur in the real setting. Hence, it is not optimal to utilize the static policy in a real dynamic environment. Therefore, a real-time policy that generates integer solutions and acknowledges the network state is required, and can be determined by solving the underlying dynamic program. Due to the continuous and high dimensional state-action spaces however, it is infeasible to develop an optimal real-time policy using exact dynamic programming algorithms. As such, we utilize deep reinforcement learning (RL) to develop a near-optimal policy. Specifically, we show that it is possible to learn a policy via Proximal Policy Optimization (PPO) [120] that increases the total profits generated by jointly managing the fleet of EVs (by making routing and charging decisions) and pricing for the rides. We demonstrate the performance of our policy by using the total profits generated and the queue

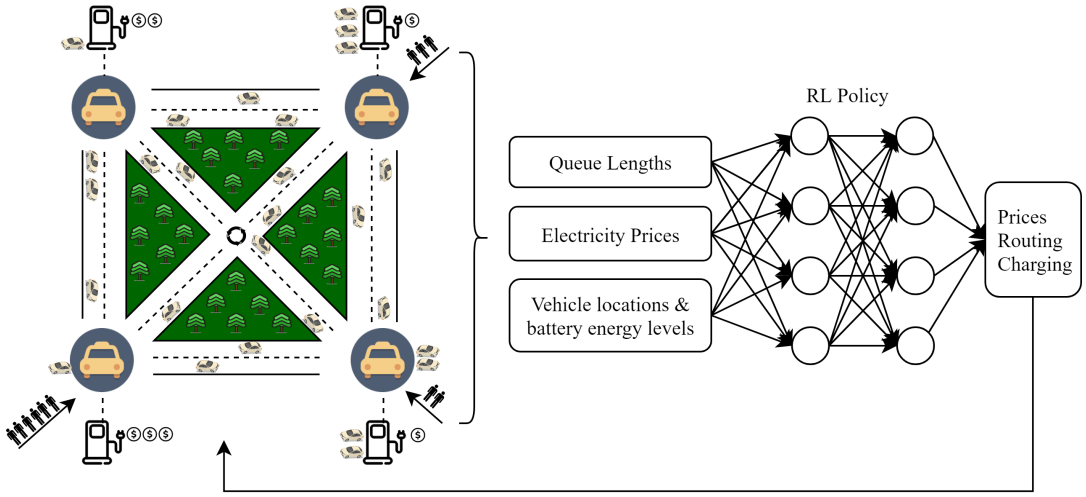


Figure 4.1: The schematic diagram of our framework. Our deep RL agent processes the state of the vehicles, queues and electricity prices and outputs a control policy for pricing as well as autonomous EVs' routing and charging.

lengths as metrics.

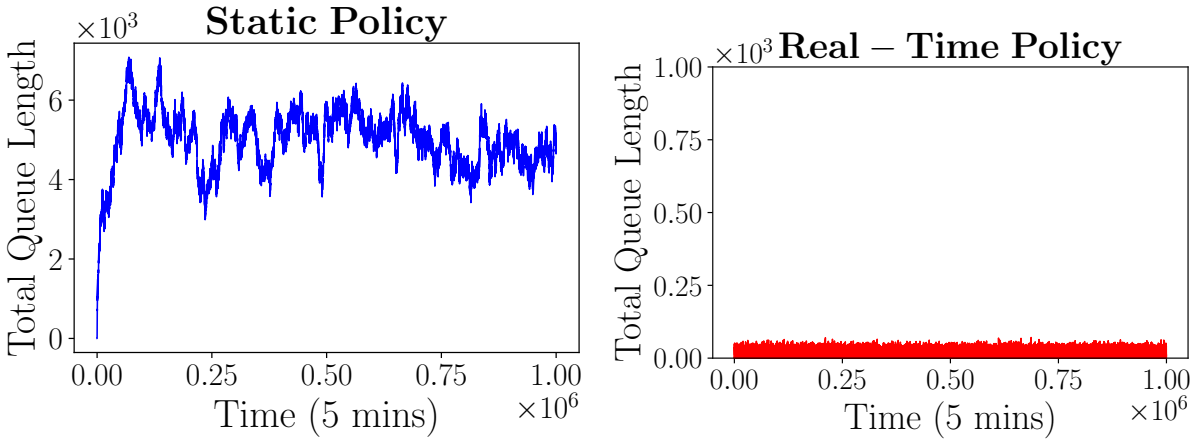
Our contributions can be summarized as follows:

1. We formalize a vehicle and network model that captures the aforementioned characteristics of an AMoD fleet of EVs as well as the stochasticity in demand and electricity prices.

2. We analyze the static problem, where we consider a time-invariant environment (time-invariant arrivals, electricity prices, etc.) to characterize the family of policies that guarantee stability of the dynamic system, to gain insight towards the actual dynamic problem, and to further provide a baseline for comparison.

3. We employ deep RL methods to learn a joint pricing, routing and charging policy that effectively stabilizes the queues and increases the profits.

We visualize our real-time framework as a schematic diagram in Figure 4.1 and preview our results in Figure 4.2, showing that a real-time pricing and routing policy can successfully keep the queue lengths 400 times lower than the static policy. This policy is also able to decrease



(a) (b)

Figure 4.2: (a) The optimal static policy manages to stabilize the queues over a very long time period but is unable to clear them whereas (b) RL control policy stabilizes the queues and manages to keep them significantly low (note the scales).

the charging costs by 25% by utilizing smart charging strategies (which will be demonstrated in Subsection 4.2.4).

Related work: Comprehensive research perceiving various aspects of AMoD systems is being conducted in the literature. Studies surrounding fleet management focus on optimal EV charging in order to reduce electricity costs as well as optimal vehicle routing in order to serve the customers and to rebalance the empty vehicles throughout the network so as to reduce the operational costs and the customers' waiting times. Time-invariant control policies adopting queueing theoretical [121], fluidic [122], network flow [123], and Markovian [124] models have been developed by using the steady state of the system. The authors of [125] consider ride-sharing systems with mixed autonomy. However, the proposed control policies in these papers are not adaptive to the time-varying nature of the future demand. As such, there is work on developing time-varying model predictive control (MPC) algorithms [126, 127, 128, 129, 130]. The authors of [128, 129] propose data-driven algortihms and the authors of [130] propose a stochastic MPC algorithm focusing on vehicle rebalancing. In [126], the authors also consider a fleet of EVs and hence propose an MPC approach that optimizes vehicle routing and

scheduling subject to energy constraints. Using a fluid-based optimization framework, the authors of [131] investigate tradeoffs between fleet size, rebalancing cost, and queueing effects in terms of passenger and vehicle flows under time-varying demand. The authors in [132] develop a parametric controller that approximately solves the intractable dynamic program for rebalancing over an infinite-horizon. Similar to AMoD, carsharing systems also require rebalancing in order to operate efficiently. By adopting a Markovian model, the authors of [133] introduce a dynamic proactive rebalancing algorithm for carsharing systems by taking into account an estimate of the future demand using historical data. In [134], the authors develop an integrated multi-objective mixed integer linear programming optimization and discrete event simulation framework to optimize vehicle and personnel rebalancing in an electric carsharing system. Using a network-flow based model, the authors of [135] propose a two-stage approximation scheme to establish a real-time rebalancing algorithm for shared mobility systems that accounts for stochasticity in customer demand and journey valuations.

Aside from these, there are studies on applications of RL methods in transportation such as adaptive routing [136], traffic management [137, 138], traffic signal control [139, 140], and dynamic routing of autonomous vehicles with the goal of reducing congestion in mixed autonomy traffic networks [141]. Relevant studies to the work presented in this section aim to develop dynamic policies for rebalancing as well as ride request assignment via decentralized reinforcement learning approaches [142, 143, 144, 145]. In these works however, the policies are developed and applied locally by each autonomous vehicle and this decentralized approach may sacrifice system level optimality. A centralized deep RL approach tackling the rebalancing problem is proposed in [146], which is closest to the approach we adopt in this section. Although their study adopts a centralized deep RL approach similar to the work presented in this section, they have a different system model and solely focus on the rebalancing problem and do not consider pricing for rides as a control variable for the queues nor the charging problem of EVs as reviewed next.

Regarding charging strategies for large populations of EVs, [147, 148, 149] provide in-depth reviews and studies of smart charging technologies. An agent-based model to simulate the operations of an AMoD fleet of EVs under various vehicle and infrastructure scenarios has been examined in [150]. By augmenting optimal battery management of autonomous electric vehicles to the classic dial-a-ride problem (DARP), the authors of [151] introduce the electric autonomous DARP that aims to minimize the total travel time of all the vehicles and riders. The authors of [12] propose an online charge scheduling algorithm for EVs providing AMoD services. By adopting a static network flow model in [13], the benefits of smart charging have been investigated and approximate closed form expressions that highlight the trade-off between operational costs and charging costs have been derived. Furthermore, [152] studies interactions between AMoD systems and the power grid. In addition, [153] studies the implications of pricing schemes on an AMoD fleet of EVs. In [154], the authors propose a dynamic joint pricing and routing strategy for non-electric shared mobility on demand services. [155] studies a quadratic programming problem in order to jointly optimize vehicle dispatching, charge scheduling, and charging infrastructure, while the demand is defined exogenously.

To the best of our knowledge, there is no existing work on centralized real-time management for electric AMoD systems addressing the joint optimization scheme of vehicle routing and charging as well as pricing for the rides. In this section, we aim to highlight the benefits of a real-time controller that jointly: (i) routes the vehicles throughout the network in order to serve the demand for rides as well as to relocate the empty vehicles for further use, (ii) executes smart charging strategies by exploiting the diversity in the electricity prices (both geographically and temporally) in order to minimize charging costs, and (iii) adjusts the demand for rides by setting prices in order to stabilize the system (i.e., the queues of customers waiting for rides) while maximizing profits.

Organization: The remainder of the section is organized as follows. In Subsection 4.2.1,

we present the system model and define the platform operator's optimization problem. In Subsection 4.2.2, we discuss the static planning problem associated with the system model and characterize the optimal static policy. In Subsection 4.2.3, we propose a method for developing a near-optimal real-time policy using deep reinforcement learning. In Subsection 4.2.4, we present the numerical results of the case studies we have conducted in Manhattan and San Francisco to demonstrate the performance of our real-time control policy.

4.2.1 System Model and Problem Definition

Network and Demand Models: We consider a fleet of AMoD EVs operating within a transportation network characterized by a fully connected graph consisting of $\mathcal{M} = \{1, \dots, m\}$ nodes that can each serve as a trip origin or destination. We study a discrete-time system with time periods normalized to integral units $t \in \{0, 1, 2, \dots\}$. In this discrete-time system, we model the arrival of the potential riders with OD pair (i, j) as a Poisson process with an arrival rate of $\lambda_{ij}(t)$ in period t, where $\lambda_{ii}(t) = 0$. We adopted a price-responsive rider model studied in [156]. We assume that the riders are heterogeneous in terms of their willingness to pay. In particular, if the price for receiving a ride from node i to node j in period t is set to $\ell_{ij}(t)$, the induced arrival rate for rides from i to j is given by $\Lambda_{ij}(t) = \lambda_{ij}(t)(1 - F(\ell_{ij}(t)))$, where $F(\cdot)$ is the cumulative distribution of riders' willingness to pay with a support of $[0, \ell_{\max}]$[1]. Thus, the number of new ride requests in time period t is $A_{ij}(t) \sim \text{Pois}(\Lambda_{ij}(t))$ for OD pair (i, j).

Vehicle Model: To capture the effect of trip demand and the associated charging and routing (routing also implies rebalancing of the empty vehicles) decisions on the costs associated with operating the fleet (maintenance, mileage, etc.), we assume that each autonomous vehicle in the fleet has a per period operational cost of β. Furthermore, as the vehicles are electric, they

[1]For brevity of notation, we uniformly set $\ell_{\max}$ to be the maximum willingness to pay for all OD pairs without loss of generality. Our results can be derived in a similar fashion by replacing $\ell_{\max}$ with $\ell^{ij}_{\max}$, where $\ell^{ij}_{\max}$ is the maximum willingness to pay for OD pair (i, j).

have to sustain charge in order to operate. Without loss of generality, we assume there is a charging station placed at each node $i \in \mathcal{M}$. To charge at node i during time period t, the operator pays a price of electricity $p_i(t)$ per unit of energy. We assume that all EVs in the fleet have a battery capacity denoted as $v_{\max} \in \mathbb{Z}^+$; therefore, each EV has a discrete battery energy level $v \in \mathcal{V}$, where $\mathcal{V} = \{v \in \mathbb{N} | 0 \leq v \leq v_{\max}\}$. In our discrete-time model, we assume each vehicle takes one period to charge one unit of energy and τ_{ij} periods to travel between OD pair (i, j), while consuming v_{ij} units of energy[2].

Ride Hailing Model: The platform operator dynamically routes the fleet of EVs in order to serve the demand at each node. Customers that purchase a ride are not immediately matched with a ride, but enter the queue for OD pair (i, j). After the platform operator executes routing decisions for the fleet, the customers in the queue for OD pair (i, j) are matched with rides and served in a first-come, first-served discipline. A measure of the expected wait time is not available to each arriving customer. However, the operator knows that longer wait times will negatively affect their business and hence seeks to minimize the total wait time experienced by users. Denote the queue length for OD pair (i, j) by $q_{ij}(t)$. If after serving the customers, the queue length $q_{ij}(t) > 0$, the platform operator is penalized by a fixed cost of w per person at the queue to account for the value of time of the customers.

Platform Operator's Problem: We consider a profit-maximizing AMoD operator that manages a fleet of EVs that make trips to provide transportation services to customers. The operator's goal is to maximize profits by 1) setting prices for rides and hence managing customer demand at each node; 2) optimally operating the AMoD fleet (i.e., charging and routing) to minimize operational and charging costs. We will study two types of control policies the platform operator utilizes: 1) a static policy, where the pricing, routing and charging decisions

[2]In this section, we consider the travel times to be constant and exogenously defined for the time period the policy is developed for. This is because we assume that the number of AMoD vehicles is much less compared to the rest of the traffic. Also, to consider changing traffic conditions throughout the day, it is possible to train multiple static and real-time control policies for the different time intervals.

are time invariant and independent of the state of the system; 2) a real-time policy, where the pricing, routing and charging decisions are dependent on the system state.

4.2.2 Analysis of the Static Problem

In this subsection, we establish and discuss the static planning problem to provide a measure for comparison and demonstrate the efficacy of the real-time policy (which will be discussed in Subsection 4.2.3). To do so, we consider the fluid scaling of the dynamic network and characterize the static problem via a network flow formulation. Under this setting, we use the expected values of the variables (arrivals and prices of electricity) and ignore their time dependent dynamics, while allowing the vehicle routing decisions to be flows (real numbers) rather than integers. The static problem is convenient for determining the optimal static pricing, routing, and charging policy, under which the queueing network of the dynamic system is stable [157][3].

4.2.2.1 Static Profit Maximization Problem

We formulate the static optimization problem via a network flow model that aims to maximize the platform operator's profits. The platform operator maximizes its profits by setting prices and making routing and charging decisions such that the system remains stable.

Let ℓ_{ij} be the prices for rides for OD pair (i, j), x_{ij}^v be the number of vehicles at node i with energy level v being routed to node j, and x_{ic}^v be the number of vehicles charging at node i starting with energy level v. We state the platform operator's profit maximization problem as follows:

[3]The stability condition that we are interested in is rate stability of all queues. A queue for OD pair (i, j) is rate stable if $\lim_{t\to\infty} q_{ij}(t)/t = 0$.

$$\max_{x_{ic}^v, x_{ij}^v, \ell_{ij}} \quad \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{M}} \lambda_{ij}\ell_{ij}(1-F(\ell_{ij})) - \sum_{i\in\mathcal{M}}\sum_{v=0}^{v_{\max}-1}(\beta+p_i)x_{ic}^v - \beta\sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{M}}\sum_{v=v_{ij}}^{v_{\max}} x_{ij}^v\tau_{ij} \tag{4.1a}$$

$$\text{subject to} \quad \lambda_{ij}(1-F(\ell_{ij})) \le \sum_{v=v_{ij}}^{v_{\max}} x_{ij}^v \quad \forall i,j\in\mathcal{M}, \tag{4.1b}$$

$$x_{ic}^v + \sum_{j\in\mathcal{M}} x_{ij}^v = x_{ic}^{v-1} + \sum_{j\in\mathcal{M}} x_{ji}^{v+v_{ji}} \quad \forall i\in\mathcal{M},\ \forall v\in\mathcal{V}, \tag{4.1c}$$

$$x_{ic}^{v_{\max}} = 0 \quad \forall i\in\mathcal{M}, \tag{4.1d}$$

$$x_{ij}^v = 0 \quad \forall v < v_{ij},\ \forall i,j\in\mathcal{M}, \tag{4.1e}$$

$$x_{ic}^v \ge 0,\ x_{ij}^v \ge 0\ \ \forall i,j\in\mathcal{M},\ \forall v\in\mathcal{V}, \tag{4.1f}$$

$$x_{ic}^v = x_{ij}^v = 0 \quad \forall v\notin\mathcal{V},\ \forall i,j\in\mathcal{M}. \tag{4.1g}$$

The first term in the objective function in (4.1) accounts for the aggregate revenue the platform generates by providing rides for $\lambda_{ij}(1-F(\ell_{ij}))$ number of riders with a price of ℓ_{ij}. The second term is the operational and charging costs incurred by the charging vehicles (assuming that $p_i(t) = p_i\ \forall t$ under the static setting), and the last term is the operational costs of the trip-making vehicles (including rebalancing trips).

The constraint (4.1b) requires the platform to operate at least as many vehicles to serve all the induced demand between any two nodes i and j (The rest are the vehicles travelling without passengers, i.e., rebalancing vehicles). We will refer to this as the *demand satisfaction constraint*. The constraint (4.1c) is the *flow balance constraint* for each node and each battery energy level, which restricts the number of available vehicles at node i and energy level v to be the sum of arrivals from all nodes (including idle vehicles) and vehicles that are charging with energy level $v-1$. The constraint (4.1d) ensures that the vehicles with full battery do not charge further, and the constraint (4.1e) ensures the vehicles sustain enough charge to travel

between OD pair (i, j).

The solution to the optimization problem in (4.1) is the optimal static policy that consists of optimal prices as well as optimal vehicle routing and charging decisions. This policy can not directly be implemented in a real environment because it does not yield integer valued solutions. It is possible generate integer-valued solutions to be implemented in a real environment using the fractional flows (e.g., randomizing the vehicle decisions according to the flows, which we do in Subsection 4.2.4), yet the methodology is not the focus of the work presented in this section. Instead, we highlight a sufficient condition for a realizable policy (generating integer valued actions) to provide stability according to the feasible solutions of (4.1):

Proposition 4.2.1 *Let $\{\tilde{\ell}_{ij}, \tilde{x}^v_{ij}, \tilde{x}^v_{ic}\}$ be a feasible solution of* (4.1)*. Let μ be a policy that generates integer actions and can be implemented in the real environment. Then, μ guarantees stability of the system if for all OD pairs (i, j):*

1. *The time average of the induced arrivals equals $(1 - F(\tilde{\ell}_{ij}))$, and*
2. *The time average of the routed vehicles equals $\sum_{v=v_{ij}}^{v_{\max}} \tilde{x}^v_{ij}$.*

The proof of Proposition 4.2.1 is provided in Appendix C.1.1. According to Proposition 4.2.1, for a static pricing policy with the optimal prices ℓ^*_{ij}, there exists an integer-valued routing and charging policy that maintains stability of the system.

Corollary 4.2.1 *An example policy that generates integer-valued actions is randomizing according to the flows. Precisely, given a feasible solution $\{\tilde{\ell}_{ij}, \tilde{x}^v_{ij}, \tilde{x}^v_{ic}\}$ of* (4.1)*, integer-valued actions can be generated by routing a vehicle at node i with energy level v to node j with probability*

$$\psi^v_{ij} = \frac{\tilde{x}^v_{ij}}{\sum_{k=1}^m \tilde{x}^v_{ik} + x^v_{ic}},$$

and charging with probability

$$\psi^v_{ic} = \frac{\tilde{x}^v_{ic}}{\sum_{k=1}^m \tilde{x}^v_{ik} + x^v_{ic}},$$

$\forall i, j \in \mathcal{M}$ and $\forall v \in \mathcal{V}$. Combining this randomized policy with a static pricing policy of $\ell_{ij}(t) = \tilde{\ell}_{ij},\ \forall t$, results in a policy satisfying the criteria in Proposition 4.2.1.

The optimization problem in (4.1) is non-convex for a general $F(\cdot)$. Nonetheless, when the platform's profits are convex in the induced demand $\lambda_{ij}(1 - F(\cdot))$, it can be rewritten as a convex optimization problem and can be solved exactly. Hence, we assume that the rider's willingness to pay is uniformly distributed in $[0, \ell_{\max}]$, i.e., $F(\ell_{ij}) = \frac{\ell_{ij}}{\ell_{\max}}$[4].

Marginal Pricing: The prices for rides are a crucial component of the profits generated. The next proposition highlights how the optimal prices ℓ^*_{ij} for rides are related to the network parameters, prices of electricity, and the operational costs.

Proposition 4.2.2 *Let ν^*_{ij} be optimal the dual variable corresponding to the demand satisfaction constraint for OD pair (i, j). The optimal prices ℓ^*_{ij} are:*

$$\ell^*_{ij} = \frac{\ell_{\max} + \nu^*_{ij}}{2}. \tag{4.2}$$

These prices can be upper bounded by:

$$\ell^*_{ij} \leq \frac{\ell_{\max} + \beta(\tau_{ij} + \tau_{ji} + v_{ij} + v_{ji}) + v_{ij}p_j + v_{ji}p_i}{2} \tag{4.3}$$

[4]It is also possible to use other distributions that might reflect real willingness-to-pay distributions more accurately (such as pareto distribution, exponential distribution, triangular distribution, constant elasticity distribution, and normal distribution). Among these, pareto, exponential, and constant elasticity distributions preserve convexity and therefore the static planning problem can be solved efficiently. Triangular and normal distributions are not convex in their support and therefore the static planning problem is not a convex optimization problem. Nevertheless, it can still be solved numerically for the optimal static policy. Using these distributions however we cannot derive the closed-form results that allow us to interpret the pricing policy of the platform operator. The real-time policy proposed in Subsection 4.2.3 uses model-free Reinforcement Learning and therefore can be applied using other distributions or any other customer price response model.

Moreover, with these optimal prices ℓ^*_{ij}, the profits generated per period is:

$$P = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\lambda_{ij}}{\ell_{\max}} (\ell_{\max} - \ell^*_{ij})^2. \tag{4.4}$$

The proof of Proposition 4.2.2 is provided in Appendix C.1.2. Observe that the profits in Equation (4.4) are decreasing as the prices for rides increase. Thus expensive rides generate less profits compared to the cheaper rides and it is more beneficial if the optimal dual variables ν^*_{ij} are small and prices are close to $\ell_{\max}/2$. We can interpret the dual variables ν^*_{ij} as the cost of providing a single ride between i and j to the platform. In the worst case scenario, every single requested ride from node i requires rebalancing and charging both at the origin and the destination. Hence the upper bound on (4.3) includes the operational costs of passenger-carrying, rebalancing and charging vehicles (both at the origin and the destination); and the energy costs of both passenger-carrying and rebalancing trips multiplied by the price of electricity at the trip destinations. Similar to the taxes applied on products, whose burden is shared among the supplier and the customer; the costs associated with rides are shared among the platform operator and the riders (which is why the price paid by the riders include half of the cost of the ride).

Although the static policy guarantees stability (by appropriate implementation of integer-valued actions as dictated by Proposition 4.2.1), it does not perform well in a real dynamic setting because it does not acknowledge the stochastic dynamics of the system. On the other hand, a real-time policy that executes decisions based on the current state of the environment would likely perform better (e.g., if the queue length for OD pair (i, j) is very large, then it is probably better for the platform operator to set higher prices to prevent the queue from growing further). Accordingly, we present a practical way of implementing a real-time policy in the next subsection.

4.2.3 The Real-Time Policy

The static policy established in the previous subsection has three major issues:

1. Because it is based on a flow model, it generates static fractional flows that are not directly implementable in the real setting.

2. It neglects the stochastic events that occur in the dynamic setting (e.g., the induced arrivals), and assumes everything is deterministic. Hence, it does consider the unexpected occurrences (e.g., queues might build in the dynamic setting, whereas the static model assumes no queues) when executing actions.

3. It assumes perfect knowledge of the network parameters (arrivals, trip durations, energy consumptions of the trips, and prices of electricity).

Due to the above reasons, it is impractical to implement the static policy in the dynamic environment. A real-time policy that generates integer solutions and takes into account the current state of the network which is essential for decision making is necessary, and can be determined by solving the dynamic program that describes the system (with full knowledge of the network parameters) for the optimal policy. Such solutions would address issues 1 and 2 outlined above. Inspired by our theoretical model, the state information that describes the network fully consists of the vehicle states (locations, energy levels), queue lengths for each OD pair, and electricity prices at each node. Upon obtaining the full state information, the actions have to be executed for pricing for rides and fleet management (vehicle routing and charging). Consequent to taking actions, the platform operator observes a reward (consisting of revenue gained by arrivals, queue costs, and operational and charging costs), and the network transitions into a new state (Although the transition into the new state is stochastic, the random processes that govern this stochastic transition is known if the network parameters are known). The solution of this dynamic program is the optimal policy that determines which action to

take for each state the system is in, and can nominally be derived using classical exact dynamic programming algorithms (e.g., value iteration). However, the complexity and the scale of our dynamic problem presents a difficulty here: Aside from having a large dimensional state space (for instance, $m = 10,\ v_{\max} = 5,\ \tau_{ij} = 3\ \forall i, j$: the state has dimension 1240) and action space, the cardinality of these spaces are not finite (queues can grow unbounded, prices are continuous). Considering that the computational complexity per iteration for value iteration is $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2)$ and for policy iteration $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2 + |\mathcal{S}|^3)$ [158], where $\mathcal{S}$ and $\mathcal{A}$ are the state space and the action space, respectively, the problem is computationally intractable to solve using classical dynamic programming. Even if we did make them finite by putting a cap on the queue lengths and discretizing the prices, curse of dimensionality renders the problem intractable to solve with classical exact dynamic programming algorithms. As such, we resort to approximate dynamic programming methods. Specifically, we define the policy via a deep neural network that takes the full state information of the network as input and outputs the best action[5]. Subsequently, we apply a model-free reinforcement learning algorithm to train the neural network in order to improve the performance of the policy. Since it is model-free, it does not require a modeling of the network (hence, it does not require knowledge of the network parameters), which resolves the third issue associated with the static policy.

We adopted a practical policy gradient method, called Proximal Policy Optimization (PPO), developed in [120], which is effective for optimizing large nonlinear policies such as neural networks. We chose PPO mainly because it supports continuous state-action spaces and guarantees monotonic improvement.[6]

We note that it is possible to apply reinforcement learning to learn a policy in any environ-

[5]In general, the policy is a stochastic policy and determines the probabilities of taking the actions rather than deterministically producing an action.

[6]Although the policy outputs a continuous set of actions, integer actions can be generated by randomizing. This is done during both training and testing, therefore the RL agent observes the integer state transitions and learns as if the policy outputs integer actions. We discuss how to generate integer actions in more detail in Subsection 4.2.3.1.

ment, real or artificial, as long as there is data available. In this section, we use our theoretical model described in Subsection 4.2.1 to create the environment and generate data, mainly because there is no electric AMoD microsimulation environment available and also to verify our findings about the static policy. Developing a microsimulator for electric AMoD (like SUMO [159]) and integrating it with a deep reinforcement learning library to create a framework for real traffic experiments remains a future work. To ensure that our numerical experiments are reproducible, in the remainder of this subsection, we describe the Markov Decision Process (MDP) that governs this dynamic environment, which is a direct extension of our static model. It is also possible to enrich the environment and the MDP to reflect real life constraints more accurately such as road capacity and charging station constraints. Since the approach we adopt to develop the real-time policy is model-free, it can be applied identically.

In Subsection 4.2.4 we present numerical results on real-time policies developed through reinforcement learning based on dynamic environments generated through our theoretical model. The goal of the experiments is to primarily answer the following questions:

1. Can we develop a real-time control and pricing policy for AMoD using reinforcement learning and what are its potential benefits over the static policy?

2. How does the policy trained for a specific network perform, if the network parameters change?

3. Can we develop a global policy that can be utilized in any network with moderate fine tuning?

The reader may skip reading Subsection 4.2.3.1 if they are not interested in the details of the MDP model used in our numerical experiment.

4.2.3.1 The Real-Time Problem as MDP

We define the MDP by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the state transition operator and r is the reward function. We describe these elements as follows:

1. $\mathcal{S}$: The state space consists of prices of electricity at each node, the queue lengths for each origin-destination pair, and the number of vehicles at each node and each energy level. However, since travelling from node i to node j takes τ_{ij} periods of time, we need to define intermediate nodes. As such, we define $\tau_{ij} - 1$ number of intermediate nodes between each origin and destination pair, for each battery energy level v. Hence, the state space consists of $s_d = m^2 + (v_{\max} + 1)((\sum_{i=1}^{m} \sum_{j=1}^{m} \tau_{ij}) - m^2 + 2m)$ dimensional vectors in $\mathbb{R}_{\geq 0}^{s_d}$ (We include all the non-negative valued vectors, however, only $m^2 - m$ entries can grow to infinity because they are queue lengths, and the rest are always upper bounded by fleet size or maximum price of electricity). As such, we define the elements of the state vector at time t as $\boldsymbol{s}(t) = [\boldsymbol{p}(t)\ \boldsymbol{q}(t)\ \boldsymbol{s_{veh}}(t)]$, where $\boldsymbol{p}(t) = [p_i(t)]_{i \in \mathcal{M}}$ is the electricity prices state vector, $\boldsymbol{q}(t) = [q_{ij}(t)]_{i,j \in \mathcal{M}; i \neq j}$ is the queue lengths state vector, and $\boldsymbol{s_{veh}}(t) = [s_{ijk}^{v}(t)]_{\forall i,j,k,v}$ is the vehicle state vector, where $s_{ijk}^{v}(t)$ is the number of vehicles at vehicle state (i, j, k, v). The vehicle state (i, j, k, v) specifies the location of a vehicle that is travelling between OD pair (i, j) as the k'th intermediate node between nodes i and j, and specifies the battery energy level of a vehicle as v (The states of the vehicles at the nodes $i \in \mathcal{M}$ with energy level v is denoted by $(i, i, 0, v)$).

2. $\mathcal{A}$: The action space consists of prices for rides at each origin-destination pair and routing/charging decisions for vehicles at nodes $i \in \mathcal{M}$ at each energy level v. The price actions are continuous in range $[0, \ell_{\max}]$. Each vehicle at state $(i, i, 0, v)$ $(\forall i \in \mathcal{M},\ \forall v \in \mathcal{V})$ can either charge, stay idle or travel to one of the remaining $m - 1$ nodes. To allow for different transitions for vehicles at the same state (some might charge, some might travel to another node),

we define the action taken at time t for vehicles at state $(i, i, 0, v)$ as an $m + 1$ dimensional probability vector with entries in $[0, 1]$ that sum up to 1: $\boldsymbol{\alpha_i^v}(t) = [\alpha_{i1}^v(t) \dots \alpha_{im}^v(t)\ \alpha_{ic}^v(t)]$, where $\alpha_{ic}^{v_{\max}}(t) = 0$ and $\alpha_{ij}^v(t) = 0$ if $v < v_{ij}$. The action space is then all the vectors $\boldsymbol{a}$ of dimension $a_d = m^2 - m + (v_{\max} + 1)(m^2 + m)$, whose first $m^2 - m$ entries are the prices and the rest are the probability vectors satisfying the aforementioned properties. As such, we define the elements of the action vector at time t as $\boldsymbol{a}(t) = [\boldsymbol{\ell}(t)\ \boldsymbol{\alpha}(t)]$, where $\boldsymbol{\ell}(t) = [\ell_{ij}]_{i,j\in\mathcal{M},i\neq j}$ is the vector of prices and $\boldsymbol{\alpha}(t) = [\boldsymbol{\alpha_i^v}(t)]_{\forall i,v}$ is the vector of routing/charging actions.

3. $\mathcal{T}$: The transition operator is defined as $\mathcal{T}_{ijk} = Pr(\boldsymbol{s}(t + 1) = j|\boldsymbol{s}(t) = i,\ \boldsymbol{a}(t) = k)$. We can define the transition probabilities for electricity prices $\boldsymbol{p}(t+1)$, queue lengths $\boldsymbol{q}(t+1)$, and vehicle states $\boldsymbol{s_{veh}}(t + 1)$ as follows:

Electricity Price Transitions: Since we assume that the dynamics of prices of electricity are exogenous to our AMoD system, $Pr(\boldsymbol{p}(t + 1) = \boldsymbol{p_2}|\boldsymbol{p}(t) = \boldsymbol{p}_1,\ \boldsymbol{a}(t)) = Pr(\boldsymbol{p}(t + 1) = \boldsymbol{p}_2|\boldsymbol{p}(t) = \boldsymbol{p_1})$, i.e., the dynamics of the price are independent of the action taken. Depending on the setting, new prices might either be deterministic or distributed according to some probability density function at time t: $\boldsymbol{p}(t) \sim \mathcal{P}(t)$, which is determined by the electricity provider.

Vehicle Transitions: For each vehicle at node i and energy level v, the transition probability is defined by the action probability vector $\boldsymbol{\alpha_i^v}(t)$. Each vehicle transitions into state $(i, j, 1, v-v_{ij})$ with probability $\alpha_{ij}^v(t)$, stays idle in state $(i, i, 0, v)$ with probability $\alpha_{ii}^v(t)$ or charges and transitions into state $(i, i, 0, v + 1)$ with probability $\alpha_{ic}^v(t)$. The vehicles at intermediate states (i, j, k, v) transition into state $(i, j, k + 1, v)$ if $k < \tau_{ij} - 1$ or $(j, j, 0, v)$ if $k = \tau_{ij} - 1$ with probability 1. The total transition probability to the vehicle states $\boldsymbol{s_{veh}}(t+1)$ given $\boldsymbol{s_{veh}}(t)$ and $\boldsymbol{\alpha}(t)$ is the sum of all the probabilities of the feasible transitions from $\boldsymbol{s_{veh}}(t)$ to $\boldsymbol{s_{veh}}(t + 1)$ under $\boldsymbol{\alpha}(t)$, where the probability of a feasible transition is the multiplication of individual vehicle transition probabilities (since the vehicle transition probabilities are independent). Note

that instead of gradually dissipating the energy of the vehicles on their route, we immediately discharge the required energy for the trip from their batteries and keep them constant during the trip. This ensures that the vehicles have enough battery to complete the ride and does not violate the model, because the vehicles arrive to their destinations with true value of energy and a new action will only be taken when they reach the destination.

Queue Transitions: The queue lengths transition according to the prices and the vehicle routing decisions. For prices $\ell_{ij}(t)$ and induced arrival rate $\Lambda_{ij}(t)$, the probability that $A_{ij}(t)$ new customers arrive in the queue (i, j) is:

$$Pr(A_{ij}(t)) = \frac{e^{-\Lambda_{ij}(t)}\Lambda_{ij}(t)^{A_{ij}(t)}}{(A_{ij}(t))!}$$

Let us denote the total number of vehicles routed from node i to j at time t as $x_{ij}(t)$, which is given by:

$$x_{ij}(t) = \sum_{v=v_{ij}}^{v_{\max}} x_{ij}^{v}(t) = \sum_{v=v_{ij}}^{v_{\max}} s_{ij1}^{v-v_{ij}}(t+1). \tag{4.5}$$

Given $\boldsymbol{s_{veh}}(t+1)$ and $x_{ij}(t)$, the probability that the queue length $q_{ij}(t+1) = q$ is:

$$Pr(q_{ij}(t+1) = q|\boldsymbol{s}(t), \boldsymbol{a}(t), \boldsymbol{s_{veh}}(t+1)) = Pr(A_{ij}(t) = q - q_{ij}(t) + x_{ij}(t)),$$

if $q > 0$, and $Pr(A_{ij}(t) \leq -q_{ij}(t) + x_{ij}(t))$ if $q = 0$. Since the arrivals are independent, the total probability that the queue vector $\boldsymbol{q}(t+1) = \boldsymbol{q}$ is:

$$Pr(\boldsymbol{q}(t+1) = \boldsymbol{q}|\boldsymbol{s}(t), \boldsymbol{a}(t), \boldsymbol{s_{veh}}(t+1)) = \prod_{i\in\mathcal{M}} \prod_{\substack{j\in\mathcal{M} \\ j\neq i}} Pr(q_{ij}(t+1)|\boldsymbol{s}(t), \boldsymbol{a}(t), \boldsymbol{s_{veh}}(t+1)).$$

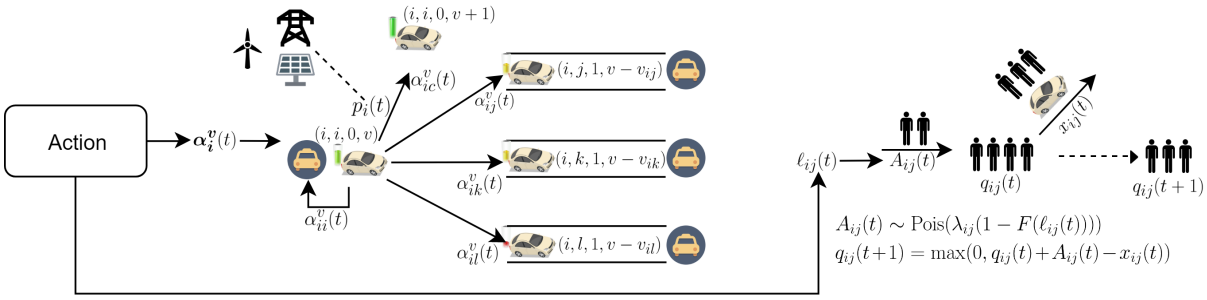


Figure 4.3: The schematic diagram representing the state transition of our MDP. Upon taking an action, a vehicle at state $(i,i,0,v)$ charges for a price of $p_i(t)$ and transitions into state $(i,i,0,v+1)$ with probability $\alpha_{ic}^v(t)$, stays idle at state $(i,i,0,v)$ with probability $\alpha_{ii}^v(t)$, or starts traveling to another node j and transitions into state $(i,j,1,v-v_{ij})$ with probability $\alpha_{ij}^v(t)$. Furthermore, $A_{ij}(t)$ new customers arrive to the queue (i,j) depending on the price $\ell_{ij}(t)$. After the routing and charging decisions are executed for all the EVs in the fleet, the queues are modified.

Hence, the transition probability is defined as:

$$\begin{aligned} Pr(\boldsymbol{s}(t+1)|\boldsymbol{s}(t),\boldsymbol{a}(t)) =& Pr(\boldsymbol{p}(t+1)|\boldsymbol{p}(t)) \times Pr(\boldsymbol{s_{veh}}(t+1)|\boldsymbol{s}(t),\boldsymbol{\alpha}(t)) \\ & \times Pr(\boldsymbol{q}(t+1)|\boldsymbol{s}(t),\boldsymbol{\alpha}(t),\boldsymbol{s_{veh}}(t+1)) \end{aligned} \tag{4.6}$$

We illustrate how the vehicles and queues transition into new states consequent to an action in Figure 4.3.

4. r: The reward function $r(t)$ is a function of state-action pairs at time t: $r(t) = r(\boldsymbol{a}(t),\boldsymbol{s}(t))$. Let $x_{ic}^v(t)$ denote the number of vehicles charging at node i starting with energy level v at time period t. The reward function $r(t)$ is defined as:

$$\begin{aligned} r(t) =& \sum_{i\in\mathcal{M}}\sum_{\substack{j\in\mathcal{M}\\ j\neq i}} \ell_{ij}(t)A_{ij}(t) - w\sum_{\in\mathcal{M}}\sum_{\substack{j\in\mathcal{M}\\ j\neq i}} q_{ij}(t) - \sum_{i\in\mathcal{M}}\sum_{v=0}^{v_{\max}-1}(\beta+p_i)x_{ic}^v(t) \\ & - \beta\sum_{i\in\mathcal{M}}\sum_{\substack{j\in\mathcal{M}\\ j\neq i}} x_{ij}(t) - \beta\sum_{i\in\mathcal{M}}\sum_{\substack{j\in\mathcal{M}\\ j\neq i}}\sum_{k=1}^{\tau_{ij}-1}\sum_{v=0}^{v_{\max}-1} s_{ijk}^v(t) \end{aligned} \tag{4.7}$$

The first term corresponds to the revenue generated by the passengers that request a ride for a price $\ell_{ij}(t)$, the second term is the queue cost of the passengers that have not yet been served, the third term is the charging and operational costs of the charging vehicles and the last two terms are the operational costs of the vehicles making trips. Note that revenue generated is immediately added to the reward function when the passengers enter the network instead of after the passengers are served. Since the reinforcement learning approach is based on maximizing the cumulative reward gained, all the passengers eventually have to be served in order to prevent queues from blowing up and hence it does not violate the model to add the revenues immediately.

Using the definitions of the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$, we model the dynamic problem as an MDP. Given large-dimensional state and action spaces with infinite cardinality, we can not solve the MDP using exact dynamic programming methods. As a solution, we characterize the real-time policy via a deep neural network and execute reinforcement learning in order to develop a real-time policy.

4.2.3.2 Reinforcement Learning Method

In this part, we go through the preliminaries of reinforcement learning and briefly explain the idea of the algorithm we adopted.

Preliminaries

The real-time policy associated with the MDP is defined as a function parameterized by θ:

$$\pi_\theta(\boldsymbol{a}|\boldsymbol{s}) = \pi : \mathcal{S} \times \mathcal{A} \to [0, 1],$$

i.e., a probability distribution in the state-action space. Given a state $\boldsymbol{s}$, the policy returns the probability for taking the action $\boldsymbol{a}$ (for all actions), and samples an action according to the

probability distribution. The goal is to derive the optimal policy π^*, which maximizes the discounted cumulative expected rewards J_π:

$$J_{\pi^*} = \max_{\pi} J_\pi = \max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(t) \right],$$
$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(t) \right],$$

where $\gamma \in (0, 1]$ is the discount factor. The value of taking an action $\boldsymbol{a}$ in state $\boldsymbol{s}$, and following the policy π afterwards is characterized by the value function $Q_\pi(\boldsymbol{s}, \boldsymbol{a})$:

$$Q_\pi(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(t) | \boldsymbol{s}(0) = \boldsymbol{s}, \boldsymbol{a}(0) = \boldsymbol{a} \right].$$

The value of being in state $\boldsymbol{s}$ is formalized by the value function $V_\pi(\boldsymbol{s})$:

$$V_\pi(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a}(0),\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(t) | \boldsymbol{s}(0) = \boldsymbol{s} \right],$$

and the advantage of taking the action $\boldsymbol{a}$ in state $\boldsymbol{s}$ and following the policy π thereafter is defined as the advantage function $A_\pi(\boldsymbol{s}, \boldsymbol{a})$:

$$A_\pi(\boldsymbol{s}, \boldsymbol{a}) = Q_\pi(\boldsymbol{s}, \boldsymbol{a}) - V_\pi(\boldsymbol{s}).$$

The methods used by reinforcement learning algorithms can be divided into three main groups: 1) critic-only methods, 2) actor-only methods, and 3) actor-critic methods, where the word critic refers to the value function and the word actor refers to the policy [160]. Critic-only (or value-function based) methods (such as Q-learning [161] and SARSA [162]) improve a

deterministic policy using the value function by iterating:

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} Q_\pi(\boldsymbol{s}, \boldsymbol{a}),$$
$$\pi(\boldsymbol{a}^*|\boldsymbol{s}) \longleftarrow 1.$$

Actor-only methods (or policy gradient methods), such as Williams' REINFORCE algorithm [163], improve the policy by updating the parameter θ by gradient ascent, without using any form of a stored value function:

$$\theta(t+1) = \theta(t) + \alpha \nabla_\theta \mathbb{E}_{\pi_{\theta(t)}} \left[\sum_\tau \gamma^\tau r(\tau) \right].$$

The advantage of policy gradient methods is their ability to generate actions from a continuous action space by utilizing a parameterized policy.

Finally, actor-critic methods [164, 165] make use of both the value functions and policy gradients:

$$\theta(t+1) = \theta(t) + \alpha \nabla_\theta \mathbb{E}_{\pi_{\theta(t)}} \left[Q_{\pi_{\theta(t)}}(\boldsymbol{s}, \boldsymbol{a}) \right].$$

Actor-critic methods are able to produce actions in a continuous action space, while reducing the high variance of the policy gradients by adding a critic (value function).

All of these methods aim to update the parameters θ (or directly update the policy π for critic-only methods) to improve the policy. In deep reinforcement learning, the policy π is defined by a deep neural network, whose weights constitute the parameter θ. To develop a real-time policy for our MDP, we adopt a practical policy gradient method called Proximal Policy Optimization (PPO).

Proximal Policy Optimization

PPO is a practical policy gradient method developed in [120], and is effective for optimizing

large non-linear policies such as deep neural networks. It preserves some of the benefits of trust region policy optimization (TRPO) [166] such as monotonic improvement, but is much simpler to implement because it can be optimized by a first-order optimizer, and is empirically shown to have better sample complexity.

In TRPO, an objective function (the "surrogate" objective) is maximized subject to a constraint on the size of the policy update so that the new policy is not too far from the old policy:

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \hat{A}_t \right] \tag{4.8a}$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t \left[\text{KL} \left[\pi_{\theta_{old}}(\cdot|\boldsymbol{s}_t), \pi_\theta(\cdot|\boldsymbol{s}_t) \right] \right] \leq \delta, \tag{4.8b}$$

where π_θ is a stochastic policy and $\hat{A}_t$ is an estimator of the advantage function at timestep t. The expectation $\hat{\mathbb{E}}_t[\dots]$ indicates the empirical average over a finite batch of samples and $\text{KL}\left[\pi_{\theta_{old}}(\cdot|\boldsymbol{s}_t), \pi_\theta(\cdot|\boldsymbol{s}_t)\right]$ denotes the Kullback–Leibler divergence between $\pi_{\theta_{old}}$ and π. Although TRPO solves the above constrained maximization problem using conjugate gradient, the theory justifying TRPO actually suggests using a penalty instead of a constraint, i.e., solving the unconstrained optimization problem

$$\underset{\theta}{\text{maximize}}\ \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \hat{A}_t - \beta \text{KL} \left[\pi_{\theta_{old}}(\cdot|\boldsymbol{s}_t), \pi_\theta(\cdot|\boldsymbol{s}_t) \right] \right], \tag{4.9}$$

for some penalty coefficient β. TRPO uses a hard constraint rather than a penalty because it is hard to choose a single value of β that performs well. To overcome this issue and develop a first-order algorithm that emulates the monotonic improvement of TRPO (without solving the constrained optimization problem), two PPO algorithms are constructed by: 1) clipping the surrogate objective and 2) using adaptive KL penalty coefficient [120].

1. *Clipped Surrogate Objective:* Let $r_t(\theta)$ denote the probability ratio $r_t(\theta) = \frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}$,

so $r(\theta_{old}) = 1$. TRPO maximizes

$$L(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \hat{A}_t \right] = \hat{\mathbb{E}}_t \left[r_t(\theta) \hat{A}_t \right]. \tag{4.10}$$

subject to the KL divergence constraint. Without a constraint however this would lead to a large policy update. To prevent this, PPO modifies the surrogate objective to penalize changes to the policy that move $r_t(\theta)$ away from 1:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right], \tag{4.11}$$

where ϵ is a hyperparameter, usually 0.1 or 0.2. The term $\text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$ modifies the surrogate objective by clipping the probability ratio, which removes the incentive for moving r_t outside of the interval $[1-\epsilon, 1+\epsilon]$. By taking the minimum of the clipped and the unclipped objective, the final objective becomes a lower bound on the unclipped objective.

2. *Adaptive KL Penalty Coefficient:* Another approach is to use a penalty on KL divergence and to adapt the penalty coefficient so that some target value of the KL divergence d_{targ} is achieved at each policy update. In each policy update, the following steps are performed:

   - Using several epochs of minibatch SGD, optimize the KL-penalized objective

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \hat{A}_t - \beta \text{KL}\left[\pi_{\theta_{old}}(\cdot|\boldsymbol{s}_t), \pi_\theta(\cdot|\boldsymbol{s}_t)\right] \right] \tag{4.12}$$

   - Compute $d = \hat{\mathbb{E}}_t \left[\text{KL}\left[\pi_{\theta_{old}}(\cdot|\boldsymbol{s}_t), \pi_\theta(\cdot|\boldsymbol{s}_t)\right]\right]$
     - If $d < d_{\text{targ}}/1.5,\ \beta \leftarrow \beta/2$
     - If $d > d_{\text{targ}} \times 1.5,\ \beta \leftarrow \beta \times 2$.

The updated β is then used for the next policy update. This scheme allows β to adjust if KL divergence is significantly different than d_{targ} so that the desired KL divergence between the old and the updated policy is attained.

A PPO algorithm using fixed-length trajectory segments is summarized in Algorithm 6. Each iteration, each of N (parallel) actors collect T timesteps of data. Then the surrogate loss on these NT timesteps of data is constructed and optimized with minibatch SGD for K epochs.

Algorithm 6: PPO, Actor-Critic Style

for iteration $= 0, 1, 2, \ldots$ **do**
  for actor $= 1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{old}}$ in environment for T timesteps.
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end
  Optimize surrogate L^{CLIP} or L^{KLPEN} w.r.t. θ, with K epochs and minibatch size $M \leq NT$.
  $\theta_{old} \leftarrow \theta$
end

In this section, we used the PPO algorithm with the clipped surrogate objective, because experimentally it it shown to have better performance than the PPO algorithm with adaptive KL penalty coefficient [120]. We refer the reader to [120] for a comprehensive study on PPO algorithms.

In the next subsection, we present our numerical studies demonstrating the performance of the RL policy.

4.2.4 Numerical Study

In this subsection, we discuss the numerical experiments and results for the performance of reinforcement learning approach to the dynamic problem and compare with the performance of several static policies, including the optimal static policy outlined in Subsection 4.2.2. We solved for the optimal static policy using CVX, a package for specifying and solving convex

programs [55]. To implement the dynamic environment compatible with reinforcement learning algorithms, we used Gym toolkit [167] developed by OpenAI to create an environment. For the implementation of the PPO algorithm, we used Stable Baselines toolkit [168].

We chose an operational cost of $\beta = \$0.1$ (by normalizing the average price of an electric car over 5 years [169]) and maximum willingness to pay $\ell_{\max} = \$30$. For prices of electricity $p_i(t)$, we generated random prices for different locations and different times using the statistics of locational marginal prices in [170]. We chose a maximum battery capacity of 20kWh. We discretrized the battery energy into 5 units, where one unit of battery energy is 4kWh. The time it takes to deliver one unit of charge is taken as one time epoch, which is equal to 5 minutes in our setup. The waiting time cost for one period is $w = \$2$ (average hourly wage is around $\$24$ in the United States [171]).

Note that the dimension of the state space grows significantly with battery capacity $v_{\max}$, because it expands the states each vehicle can have by $v_{\max}$.

Therefore, for computational purposes, we conducted two case studies: 1) Non-electric AMoD case study with a larger network in Manhattan, 2) Electric AMoD case study with a smaller network in San Francisco. We picked two different real world networks in order to demonstrate the universality of reinforcement learning method in establishing a real-time policy. In particular, our intention is to support the claim that the success of the reinforcement learning method is not restricted to a single network, but generalizes to multiple real world networks. Both experiments were performed on a laptop computer with Intel® Core™ i7-8750H CPU (6×2.20 GHz) and 16 GB DDR4 2666MHz RAM.

4.2.4.1 Case Study in Manhattan

In a non-electric AMoD network, the energy dimension v vanishes. Because there is no charging action[7], we can perform coarser discretizations of time. Specifically, we can allow

[7]The vehicles still refuel, however this takes negligible time compared to the trip durations.

each discrete time epoch to cover $5 \times \min_{i,j|i\neq j} \tau_{ij}$ minutes, and normalize the travel times τ_{ij} and w accordingly (For EV's, because charging takes a non-negligible but shorter time than travelling, in general we have $\tau_{ij} > 1$, and larger number of states). The static profit maximization problem in (4.1) for AMoD with non-electric vehicles can be rewritten as:

$$\begin{aligned} \max_{x_{ij},\ell_{ij}} \quad & \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{M}} \lambda_{ij}\ell_{ij}(1-F(\ell_{ij})) - \beta_g \sum_{i\in\mathcal{M}}\sum_{j\in\mathcal{M}} x_{ij}\tau_{ij} \\ \text{subject to} \quad & \lambda_{ij}(1-F(\ell_{ij})) \leq x_{ij} \quad \forall i,j \in \mathcal{M}, \\ & \sum_{j\in\mathcal{M}} x_{ij} = \sum_{j\in\mathcal{M}} x_{ji} \quad \forall i \in \mathcal{M}, \\ & x_{ij} \geq 0 \ \ \forall i,j \in \mathcal{M}. \end{aligned} \tag{4.13}$$

The operational costs $\beta_g = \$2.5$ (per 10 minutes, [172]) are different than those of electric vehicles. Because there is no "charging" (or refueling action, since it takes negligible time), β_g also includes fuel cost. The optimal static policy is used to compare and highlight the performance of the real-time policy[8].

Figure 4.4: Manhattan divided into $m = 10$ regions.

We divided Manhattan into 10 regions as in Figure 4.4, and using the yellow taxi data from

[8]The solution of the static problem yields vehicle flows. In order to make the policy compatible with our environment and to generate integer actions that can be applied in a dynamic setting, we randomized the actions by dividing each flow for OD pair (i, j) (and energy level v) by the total number of vehicles in i (and energy level v) and used that fraction as the probability of sending a vehicle from i to j (with energy level v).

the New York City Taxi and Limousine Commission dataset [173] for May 04, 2019, Saturday between 18.00-20.00, we extracted the average arrival rates for rides and trip durations between the regions (we exclude the rides occurring in the same region). We trained our model by creating new induced random arrivals with the same average arrival rate using prices determined by our policy. For the fleet size, we used a fleet of 1200 autonomous vehicles (according to the optimal fleet size emerging from the static problem).

For training, we used a neural network with 4 hidden layers and 128 neurons in each hidden layer. The rest of the parameters are left as default as specified by the Stable Baselines toolkit [168]. In order to get the best policy, we train 3 different models using DDPG[174], TRPO[166], and PPO. We trained the models for 10 million iterations, and the performances of the trained models are summarized in Table 4.1 using average rewards and queue lengths as metrics. Our experiments indicate that the model trained using PPO is performing the best among the three, hence we use that model as our real-time policy.

Metrics \ Algorithms	DDPG	TRPO	PPO
Average Rewards	9825.69	13142.47	**15527.34**
Average Queue Length	431.76	87.96	**68.11**

Table 4.1: Performances of RL policies trained with different algorithms.

We compare different policies' performance using the rewards and total queue length as metrics. The results are demonstrated in Figure 4.5. In Figure 4.5a we compare the rewards generated and the total queue length by applying the static and the real-time policies as defined in Subsections 4.2.2 and 4.2.3. We can observe that while the optimal static policy provides rate stability in a dynamic setting (since the queues do not blow up), it fails to generate profits as it is not able to clear the queues. On the other hand, the real-time policy is able to keep the total length of the queues 100 times shorter than the static policy while generating higher profits.

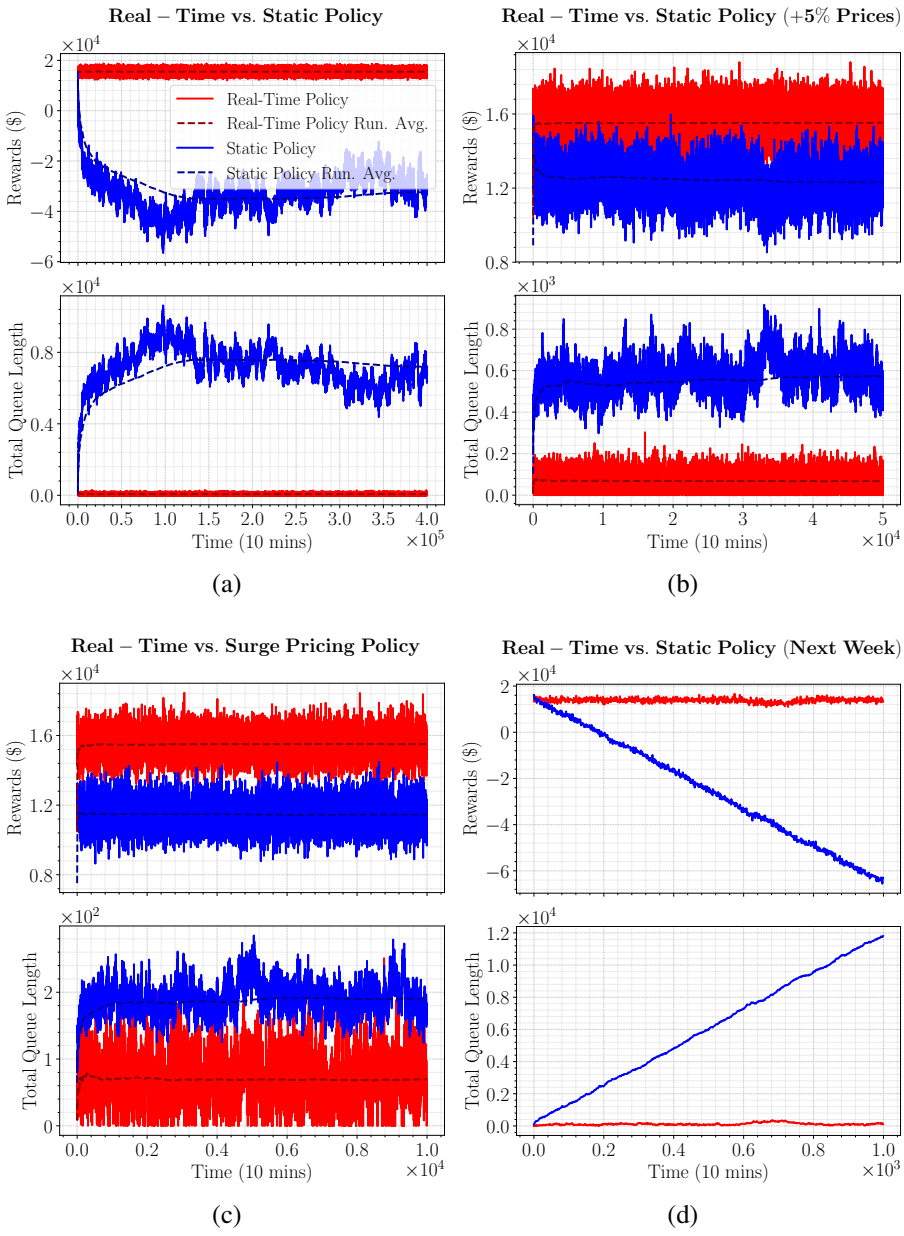


Figure 4.5: Comparison of different policies for the Manhattan case study. The legends for all figures are the same as the top left figure, where red lines correspond to the real-time policy and blue lines correspond to the static policies (We excluded the running averages for (d), because the static policy diverges). In all scenarios, we use the rewards generated and the total queue length as metrics. In (a), we demonstrate the results from applying the real-time policy and the optimal static policy. In (b), we compare the real-time policy with the static policy that utilizes 5% higher prices than the optimal static policy. In (c), we utilize a surge pricing policy along with the optimal static policy and compare with the real-time policy. In (d), we employ the real-time policy and static policy developed for May 4, 2019, Saturday for the arrivals on May 11, 2019, Saturday.

The optimal static policy fails to generate profits and is not necessarily the best static policy to apply in a dynamic setting. As such, in Figure 4.5b we demonstrate the performance of a sub-optimal static policy, where the prices are 5% higher than the optimal static prices to reduce the arrival rates and hence reduce the queue lengths. Observe that the profits generated are higher than the profits generated using optimal static policy for the static planning problem while the total queue length is less. This result indicates that under the stochasticity of the dynamic setting, a sub-optimal static policy can perform better than the optimal static policy. Furthermore, we summarize the performances of other static policies with higher static prices, namely with $5\%, 10\%, 20\%30\%$, and 40% higher prices than the optimal static prices in Table 4.2. Among these, an increase of 10% performs the best in terms of rewards. Nevertheless, this policy does still do worse in terms of rewards and total queue length compared to the real-time policy, which generates around 10% more rewards and results in 70% less queues. Lastly we note that although a 40% increase in prices results in minimum average queue length, this is a result of significantly reduced induced demand and therefore it generates very low rewards.

% of opt. static prices / Metrics	105%	110%	120%	130%	140%
Average Rewards	12234.13	**14112.77**	13739.35	12046.91	9625.82
Average Queue Length	584.05	231.93	74.64	30.88	**14.20**

Table 4.2: Performances of static pricing policies for Manhattan case study.

Next, we showcase that even some heuristic modifications which resemble what is done in practice can do better than the optimal static policy. We utilize the optimal static policy, but additionally utilize a surge-pricing policy. The surge-pricing policy aims to decrease the arrival rates for longer queues so that the queues will stay shorter and the rewards will increase. At each time period, for all OD pairs, the policy is to increase the price by 50% if the queue is longer than 100% of the induced arrival rate. The results are displayed in Figure 4.5c. New arrivals bring higher revenue per person and the total queue length is decreased, which

stabilizes the network while generating more profits than the optimal static policy. The surge pricing policy results in stable short queues and higher rewards compared to the optimal static policy for the static setting, however, both the real-time policy and the static pricing policy with 10% higher prices are superior. Performances of other surge pricing policies that multiply the prices by $1.25/1.5/2$ if the queue is longer than $50\%/100\%/200\%$ of the induced arrival rates can be found in Table 4.3. Accordingly, the best surge pricing policy maximizing the rewards is to multiply the prices by 1.25 if the queue is longer than 50% of the induced arrival rate. Yet, our real-time policy still generates around 20% more rewards and results in 32% less queues. We note that a surge pricing policy that multiplies the prices by 2 when the queues are longer than 50% of the induced arrival rates minimizes the queues by decreasing the induced arrival rates significantly, which results in substantially low rewards.

Queue Thr. / Surge	50%		100%		200%	
	Queue	Rewards	Queue	Rewards	Queue	Rewards
1.25×	101.25	**13022.83**	186.56	12897.30	380.34	12357.33
1.5×	91.89	12602.90	178.22	12589.71	370.18	12233.95
2×	**83.15**	5272.04	162.99	6224.69	337.01	7485.75

Table 4.3: Performances of surge pricing policies for Manhattan case study.

Finally, we test how the static and the real-time policies are robust to variations in input statistics. We compare the rewards generated and the total queue length applying the static and the real-time policies for the arrival rates of May 11, 2019, Saturday between 18.00-20.00. The results are displayed in Figure 4.5d. Even though the arrival rates between May 11 and May 4 do not differ much, the static policy is not resilient and fails to stabilize when there is a slight change in the network. The real-time policy, on the other hand, is still able to stabilize the network and generate profits. The neural-network based policy is able to determine the correct pricing and routing decisions by considering the current state of the network, even under different arrival rates.

These experiments show us that we can indeed develop a real-time policy using deep re-

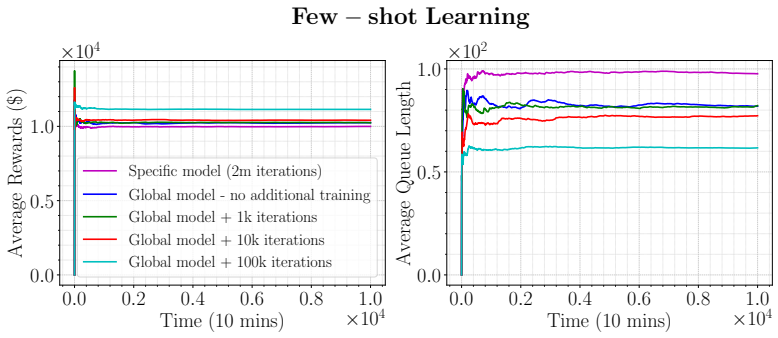


Figure 4.6: Performances of the specific model that is trained from scratch and fine-tuned global model (for different amounts of fine-tuning as specified in the legend): rewards (left) and queue lengths (right).

Figure 4.7: San Francisco divided into $m = 7$ regions. Map obtained from the San Francisco County Transportation Authority [175].

inforcement learning and this policy is resilient to small changes in the network parameters. The next study investigates the idea of generality, i.e., whether we can develop a global real-time policy and fine-tune it to a specific environment with *few-shots* of training, rather than developing a new policy from scratch.

Few-shot Learning: A common problem with reinforcement learning approaches is that because the agent is trained for a specific environment, it fails to respond to a slightly changed environment. Hence, one would need to train a different model for different environments (different network configurations, different arrival rates). However, this is not a feasible solution considering that training one model takes millions of iterations. As a more tractable solution, one could train a global model using different environments, and then calibrate it to the desired environment with fewer iterations rather than training a new model from scratch. We tested this phenomenon by training a global model for Manhattan using various arrival rates and network configurations that we extracted from different 2-hour intervals (We trained the global model for 10 million iterations). We then trained this model for the network configuration and arrival rates on May 6, 2019, Monday between 15.00-17.00. The results are displayed in Figure 4.6. Even with no additional training, the global model performs better than the specific

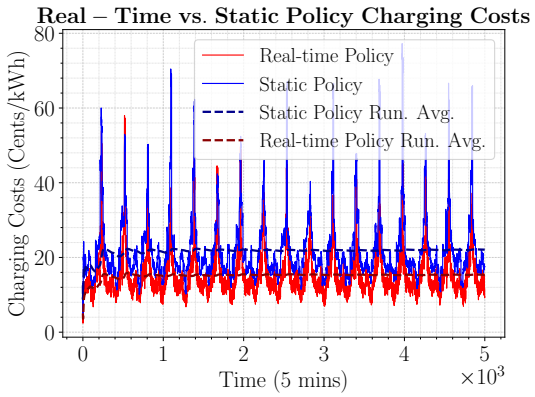


Figure 4.8: Charging costs for the optimal static policy and the real-time policy in San Francisco case study.

model trained from scratch for 2 million iterations. Furthermore, with only few iterations, it is possible to improve the performance of the global model significantly. This is an anticipated result, because although the network configurations and arrival rates for different 2-hour intervals are different, the environments are not fundamentally different (the state transitions are governed by similar random processes) and hence it is possible to generalize a global policy and fine-tune it to the desired environment with fewer number of iterations.

4.2.4.2 Case Study in San Francisco

We conducted the case study in San Francisco by utilizing an EV fleet of 420 vehicles. We divided San Francisco into 7 regions as in Figure 4.7, and using the traceset of mobility of taxi cabs data from CRAWDAD [176], we obtained the average arrival rates and travel times between regions (we exclude the rides occurring in the same region).

In Figure 4.8, we compare the charging costs paid under the real-time policy and the static policy. The static policy is generated by using the average value of the electricity prices, whereas the real-time policy takes into account the current electricity prices before executing an action. Therefore, the real-time policy provides cheaper charging options by utilizing smart charging strategies, decreasing the average charging costs by 25%.

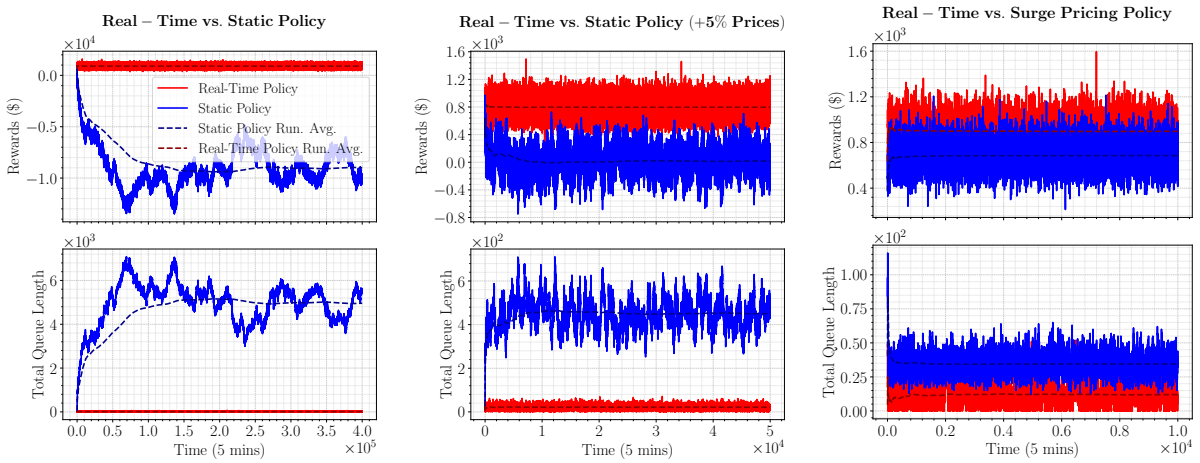


Figure 4.9: Comparison of different policies for San Francisco case study. The legends for all figures are the same as the top left figure, where red lines correspond to the real-time policy and blue lines correspond to the static policies. In all scenarios, we use the rewards generated and the total queue length as metrics. In (a), we demonstrate the results from applying the real-time policy and the optimal static policy. In (b), we compare the real-time policy with a sub-optimal static policy, where the prices are 5% higher than the optimal static policy. In (c), we utilize a surge pricing policy along with the optimal static policy and compare with the real-time policy.

In Figure 4.9a, we compare the rewards and the total queue length resulting from the real-time policy and the static policy. In Figure 4.9b, we compare the RL policy to the static policy with 5% higher prices than the optimal static policy, and summarize performances of several other static pricing policies in Table 4.5.

In Figure 4.9c, we use the static policy but also utilize a surge pricing policy that multiplies the prices by 1.5 if the queues are longer than 100% of the induced arrival rates. The performances of other surge pricing policies are also displayed in Table 4.4. Similar to the case study in Manhattan, the results demonstrate that the performance of the trained real-time policy is superior to the other policies. In particular, the RL policy is able to generate around 24% more rewards and result in around 75% less queues than the best heuristic policy, which utilizes 30% higher static prices than the optimal static policy.

Surge \ Queue Thr.	50%		100%		200%	
	Queue	Rewards	Queue	Rewards	Queue	Rewards
1.25×	67.62	**718.66**	75.92	715.02	99.56	687.45
1.5×	25.16	650.90	34.32	687.71	49.94	708.38
2×	**14.06**	331.21	20.55	455.25	44.44	611.23

Table 4.4: Performances of surge pricing policies for San Francisco case study.

Metrics \ % of opt. static prices	105%	110%	120%	130%	140%
Average Rewards	4.98	485.65	696.38	**721.89**	682.76
Average Queue Length	456.83	211.04	87.15	45.28	**25.66**

Table 4.5: Performances of static pricing policies for San Francisco case study.

4.2.5 Conclusion

In this section, we developed a real-time control policy based on deep reinforcement learning for operating an AMoD fleet of EVs as well as pricing for rides. Our real-time control policy jointly makes decisions for: 1) vehicle routing in order to serve passenger demand and to rebalance the empty vehicles, 2) vehicle charging in order to sustain energy for rides while exploiting geographical and temporal diversity in electricity prices for cheaper charging options, and 3) pricing for rides in order to adjust the potential demand so that the network is stable and the profits are maximized. Furthermore, we formulated the static planning problem associated with the dynamic problem in order to define the optimal static policy for the static planning problem. When implemented correctly, the static policy provides stability of the queues in the dynamic setting, yet it is not optimal regarding the profits and keeping the queues sufficiently low. Finally, we conducted case studies in Manhattan and San Francisco that demonstrate the performance of our developed policy. The two case studies on different networks indicate that reinforcement learning can be a universal method for establishing well performing real-time policies that can be applied to many real world networks. Lastly, by doing the Manhattan study with non-electric vehicles and San Francisco study with electric

vehicles, we have also demonstrated that a real-time policy using reinforcement learning can be established for both electric and non-electric AMoD systems.

4.3 Competition in Electric Autonomous Mobility on Demand Systems

In this section, we study the effects of competition in electric AMoD systems that are operated by profit-maximizing platform operators. Owing to the opportunities that autonomous electric vehicles create for efficient control schemes and cost-effective operation, it is possible for a single platform operator to provide cheap rides through optimizing the prices of rides for geographical load balancing as well as optimally routing and charging the fleet of electric vehicles. However, a monopolistic market with a single AMoD provider is in general disadvantageous for customer welfare. Therefore, introduction of another AMoD service provider to the market results in firms competing over the customers, hence forcing them to charge fairer prices and provide a higher quality of service. Our primary goal is to investigate the optimal behaviour of the firms in a monopoly and duopoly and quantify the impacts of competition on the customers as well as the firms.

Our contributions can be summarized as follows:

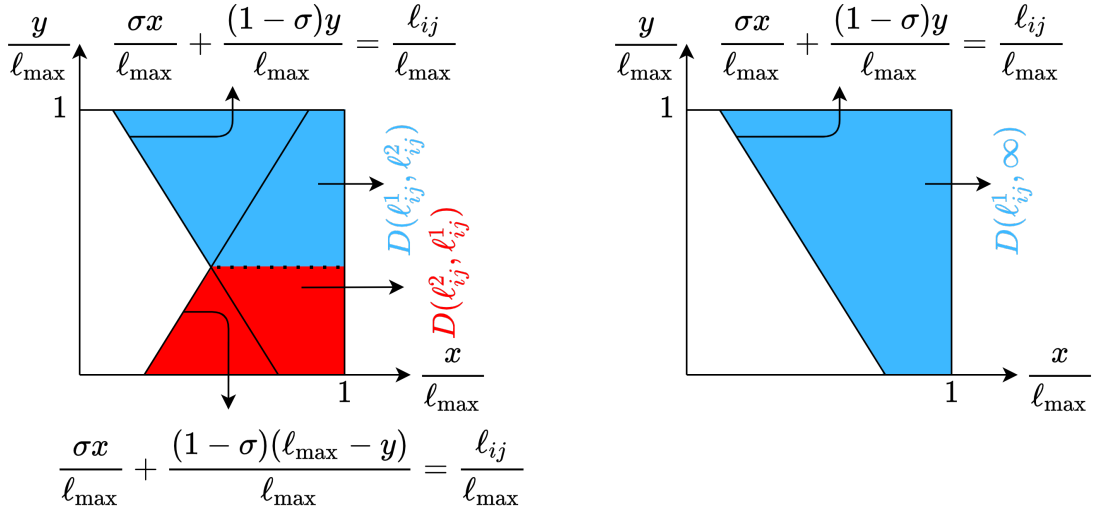
- We formalize the platform operator's profit maximization problem by adopting a static network-flow based model that captures the characteristics of an AMoD fleet, and derive expressions for the ride prices, profits, and consumer surplus under the optimal static policy.

- We prove that if the competitors have identical costs, then the duopoly equilibrium prices have to be symmetric. We show that under a mild sufficient condition on maximum travel costs that can be met with electric vehicles, the duopoly prices in equilibrium are never larger than the optimal monopoly prices. Furthermore, we derive theoretical bounds for the ratio of prices, induced demand, profits, and consumer surplus in the monopoly and the duopoly equilibrium.

- We study a real-time pricing and fleet management policy using model predictive control, and demonstrate the performance numerically on real network and demand data.

Related work: Research on competition in ride-sharing markets is relevant to ours. In terms of a broader scope on platform competition in two-sided markets, [177] and [178] introduce general frameworks and provide in-depth analysis. The impacts of single/multi-homing users on the market equilibria have been investigated in [179]. Theoretical studies on dynamic platform competition [180] and spatial platform competition [181] in two sided markets further provide insights towards competition in ride-sharing markets. Besides these, scholars examine the competition between ride-sharing and taxis [182, 183], where Uber is considered to be a monopoly. These works however do not capture the competition among ride-sharing platforms, yet ride-sharing markets are rather oligopolies in many countries[184]. Accordingly, a recent work [185] presents a head-to-head comparison of Uber, Lyft, and taxis using statistical methods. Another line of work related to ours focuses on the benefits of spatial price discrimination [186] and dynamic pricing in ride-sharing networks [187, 188]. These however do not study a competitive market. Closest to the work presented in this section is [189], which studies the effects of thickness (i.e., the mass of drivers) and competition on the equilibria of ride-sharing markets. It shows that competition always increases the welfare of the drivers, whereas it decreases the welfare of the customers if the market is not sufficiently thick.

To the best of our knowledge, there is no existing work on competition in electric AMoD systems. Our study aims to form the bridge between AMoD and competition literature with our theoretical findings. We hope that the closed form bounds quantifying the impacts of competition would help investors make informed policy decisions about competing AMoD platforms and investing in efficient AMoD technologies.



(a) Duopoly demand functions

(b) Monopoly demand functions

Figure 4.10: Graphical illustration of the demand functions for (a) duopoly, and (b) monopoly. The axes correspond to the uniform random variables x and y scaled by $1/\ell_{\max}$. In duopoly, the line $\sigma x + (1-\sigma)y = \ell_{ij}^1$ corresponds to the customers who earn 0 pay-off buying a ride from firm 1, and the line $\sigma x + (1-\sigma)(\ell_{\max} - y) = \ell_{ij}^2$ corresponds to the customers who earn 0 pay-off buying a ride from firm 2. As such, for the price tuple $(\ell_{ij}^1, \ell_{ij}^2)$, the blue shaded area corresponds to the demand function for firm 1, whereas the red corresponds to the demand function for firm 2. Monopoly is the special case of duopoly, where the prices for rides set by firm 2 are set to infinity: $\ell_{ij}^2 = \infty$.

4.3.1 System Model and Problem Definition

Network and Demand Models: We consider two fleets of AMoD EVs operated by two competitors within a transportation network characterized by a complete graph consisting of $\mathcal{N} = \{1, \dots, n\}$ nodes. Each of these nodes can serve as a trip origin or destination.

We study a discrete-time system with time periods normalized to integral units $t \in \{0, 1, 2, \dots\}$. In each period, potential riders of mass θ_{ij} seek rides between origin-destination (OD) pair (i, j), where $\theta_{ii} = 0$. We assume that customers have different valuations for riding with each firm, represented by the tuple (v_1, v_2) where v_f is the customer's valuation for firm f. To capture customer heterogeneity, we let $(v_1, v_2) \sim \mathcal{V}$, where $\mathcal{V}$ denotes the PDF of the joint distribution with support $[0, \ell_{\max}]^2$. Here, $\ell_{\max}$ is the maximum valuation of the

customers for both firms, i.e. the maximum willingness to pay[9]. To characterize the distribution $\mathcal{V}$, we adopt the model proposed by [189] and assume that the distribution of the random variables (v_1, v_2) is defined implicitly through:

$$v_1 = \sigma x + (1-\sigma)y, \tag{4.14}$$

$$v_2 = \sigma x + (1-\sigma)(\ell_{\max} - y), \tag{4.15}$$

where x and y are iid uniform random variables with support $[0, \ell_{\max}]$ and $\sigma \in [0,1]$. We refer to x as the *common value component* and y as the *idiosyncratic* component, with σ as the measure of correlation over customers' preferences[10]. In particular, x can be viewed as a customer's valuation of the ride itself and y (or $\ell_{\max} - y$) can be viewed as a customer's valuation of firm 1 itself (or firm 2 itself). A customer is identified by the draws from distributions of x and y, which are then mapped to that customer's valuations for riding with firms 1 and 2 via (4.14) and (4.15). A customer with valuations (v_1, v_2) makes a decision upon observing the prices for rides. If the prices for rides between OD pair (i,j) in period are set to be ℓ_{ij}^1 and ℓ_{ij}^2 by firm 1 and 2, respectively, the customer buys a ride from firm f if $v_f - \ell_{ij}^f > 0$ and $v_f - \ell_{ij}^f > v_{-f} - \ell_{ij}^{-f}$ (given firm f, $-f$ denotes the other firm), i.e., the customer gains a positive pay-off for purchasing a ride from firm f and this pay-off is higher than the pay-off that the customer would gain by buying from the other firm. Otherwise they do not buy a ride from either of the firms and leave the system. Hence, for a price tuple $(\ell_{ij}^1, \ell_{ij}^2)$ for OD pair (i,j), the induced mass of arrivals for firm f is given by $\Theta_{ij}^f := \theta_{ij} D(\ell_{ij}^f, \ell_{ij}^{-f})$, where $D : [0, \ell_{\max}]^2 \to [0,1]$ is the *demand function* of customers which determines the fraction of customers that would buy a ride from firm f upon observing the prices. This function has a

[9]For brevity of notation, we uniformly set $\ell_{\max}$ to be the maximum willingness to pay for all OD pairs without loss of generality. Our results can be derived in a similar fashion by replacing $\ell_{\max}$ with $\ell_{\max}^{ij}$, where $\ell_{\max}^{ij}$ is the maximum willingness to pay for OD pair (i,j).

[10]In the monopolistic setting, σ measures the correlation between customers' valuation of riding with the monopolistic firm and customers' valuation of riding with outside options (e.g., public transport).

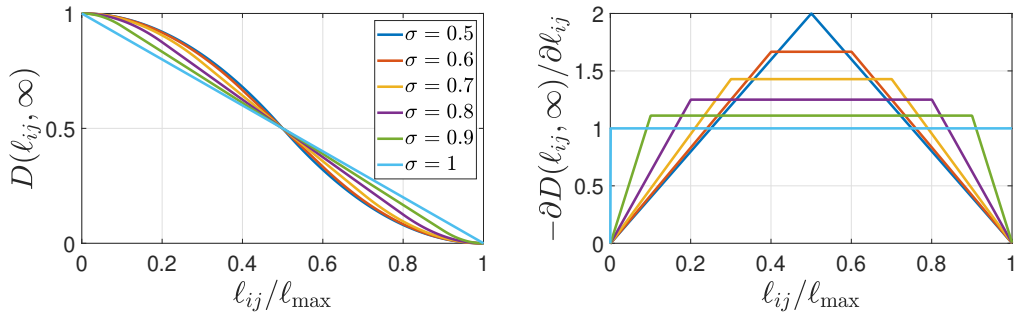


Figure 4.11: Demand function (left) and willingness to pay distribution (right) as a function of ride prices for several values of $\sigma \in [0.5, 1]$.

simple geometric interpretation depicted in Figure 4.10. We plot the demand function and the willingness to pay distribution as a function of ride prices for several values of σ in the monopolistic setting in Figure 4.11. Note that the demand function is concave if $\ell_{ij}^1 < (1-\sigma)\ell_{\max}$, is linear if $(1-\sigma)\ell_{\max} \leq \ell_{ij}^1 < \sigma\ell_{\max}$, and is convex if $\sigma\ell_{\max} \leq \ell_{ij}^1$.

Vehicle Model: In order to best serve its customers and maximize its profits, each operator needs to dispatch its fleet, including vehicle routing and charging. To implicitly capture the effect of trip demand and the associated charging and routing decisions on the fleet size and hence the operational costs incurred by each operator, we assume that each vehicle in charging or trip-making mode has a per period operational cost of β_c and β_t, respectively. A trip-making vehicle can either be occupied by a customer, which we refer to as a customer carrying vehicle; or can be empty, which we refer to as a rebalancing vehicle. We note that in the work presented in this section, we set the capacity of a vehicle to be one passenger. Furthermore, as the vehicles are electric, they have to sustain charge in order to operate, which needs to be purchased from the power grid. Without loss of generality, we assume there is a charging station placed at each node $i \in \mathcal{N}$. To charge at node i, the operator pays a price of electricity p_i per unit of energy. We assume that all EVs in the fleet have a battery capacity denoted as $e_{\max} \in \mathbb{Z}^+$; therefore, each EV has a discrete battery energy level $e \in \mathcal{E}$, where $\mathcal{E} = \{e \in \mathbb{N} | 0 \leq e \leq e_{\max}\}$. In our discrete-time model, we assume each vehicle takes one period to charge one unit of energy and

τ_{ij} periods to travel between OD pair (i, j), while consuming e_{ij} units of energy.

Platform Operator's Problem: We consider a profit-maximizing AMoD operator that manages a fleet of EVs that make trips to provide transportation services to customers. The operator's goal is to maximize profits by 1) setting prices for rides and hence managing customer demand at each node; 2) optimally operating the AMoD fleet (i.e., charging and routing) to minimize operational and charging costs. Next, we study the static planning problem for both the monopoly and the duopoly settings in order to characterize the optimal static prices and to examine the effects of competition in electric AMoD systems.

4.3.2 Analysis of the Static Problem

In this subsection, we establish and discuss the static planning problems considering a single operator (i.e., monopoly) and two competing operators (i.e., duopoly) in order to study the effect of competition in an electric AMoD system. We consider the fluid scaling of the network and characterize the static planning problem via a network flow formulation. The static problem is convenient for determining the optimal static pricing, routing, and charging policy of the platform operator.

4.3.2.1 Monopoly Static Planning Problem

We define the monopoly to be the setting where the firm 2 is removed. In order to make the comparison between the monopoly and the duopoly consistent, we keep the customer behaviour and the demand function D same. Hence, removing firm 2 from the system is equivalent to setting prices for rides posted by firm 2 to be ∞, and the induced demand for rides for OD pair (i, j) to be $D(\ell_{ij}^1, \infty)$ for a given ℓ_{ij}^1.

The goal of the platform operator is to maximize its profits by setting prices for rides and making routing and charging decisions such that the induced demand is served. Let x_{ij}^e be the

number of vehicles at node i with energy level e being routed to node j and x_{ic}^e be the number of vehicles charging at node i and currently at energy level e. We state the platform operator's problem as follows:

$$\max_{x_{ic}^e, x_{ij}^e, \ell_{ij}^1} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij}\ell_{ij}^1 D(\ell_{ij}^1, \infty) \sum_{i=1}^{n}\sum_{e=0}^{e_{\max}-1} (\beta_c + p_i) x_{ic}^e - \beta_t \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e \tau_{ij} \tag{4.16a}$$

$$\text{subject to} \quad \theta_{ij} D(\ell_{ij}^1, \infty) \leq \sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e \quad \forall i, j \in \mathcal{N}, \tag{4.16b}$$

$$x_{ic}^e + \sum_{j=1}^{n} x_{ij}^e = x_{ic}^{e-1} + \sum_{j=1}^{n} x_{ji}^{e+e_{ji}}, \quad \forall i \in \mathcal{N},\ \forall e \in \mathcal{E}, \tag{4.16c}$$

$$x_{ic}^{e_{\max}} = 0, \quad \forall i \in \mathcal{N}, \tag{4.16d}$$

$$x_{ij}^e = 0, \quad \forall e < e_{ij},\ \forall i, j \in \mathcal{N}, \tag{4.16e}$$

$$x_{ic}^e \geq 0,\ x_{ij}^e \geq 0, \ \forall i, j \in \mathcal{N},\ \forall e \in \mathcal{E}, \tag{4.16f}$$

$$x_{ic}^e = x_{ij}^e = 0, \quad \forall e \notin \mathcal{E},\ \forall i, j \in \mathcal{N}. \tag{4.16g}$$

The objective function (4.16a) corresponds to the profits earned by the firm per period. In particular, the first term in (4.16a) accounts for the aggregate revenue the platform generates by providing rides for $\theta_{ij} D(\ell_{ij}^1, \infty)$ number of riders with a price of ℓ_{ij}^1. The second term is the operational and charging costs incurred by the charging vehicles, and the last term is the operational costs of the trip-making vehicles.

The constraint (4.16b) requires the platform to operate at least as many vehicles to serve all the induced demand between any two nodes i and j (The rest are the vehicles travelling without passengers, i.e., rebalancing vehicles). We will refer to this as the *demand satisfaction constraint*. We let λ_{ij} be the dual variable associated with (4.16b) and λ_{ij}^m be the optimal dual variable. The constraint (4.16c) is the *flow balance constraint* for each node and each battery energy level, which restricts the number of available vehicles at node i and energy level e to

be the sum of arrivals from all nodes and vehicles that are charging with energy level $e-1$. The constraint (4.16d) ensures that the vehicles with full battery do not charge further, and the constraint (4.16e) ensures the vehicles sustain enough charge to travel between OD pair (i,j).

It is worthwhile to mention that unlike traditional minimum-cost flow problems, where the objective is to minimize total travel cost, the objective of (4.16) is to maximize the total revenue minus the costs, i.e., profits. Furthermore, in traditional minimum-cost flow problems, demand elasticity in response to price is not explicit and the elasticity is often modeled in response to travel times [190, 191], whereas the explicit dependency of the induced demand to prices via $D(\ell_{ij}^1, \infty)$ results in a more challenging task. The prices affect the induced demand, which affects the routing decisions and this causes a complex interplay between the decision variables.

Optimal Pricing: The prices for rides are a crucial component of the profits generated. The next proposition highlights how the optimal prices $\ell_{ij}^m := \ell_{ij}^{1*}$ for rides are related to the network parameters, prices of electricity, and the operational costs. In the following results, we investigate this interconnection by providing upper bounds on the prices that a profit-maximizing monopolist may charge customers, as well the corresponding profits generated. We highlight the fact that the monopolist's profits are in fact a decreasing function of the optimal prices for rides. The higher the monopolist has to charge its customers, the lower its generated profits. This could be a motivation for the monopolist to invest in efficient vehicle technology and cheap charging solutions.

Proposition 4.3.1 *Define*

$$\overline{\lambda}_{ij} := \beta_t(\tau_{ij} + \tau_{ji}) + e_{ij}(p_j + \beta_c) + e_{ji}(p_i + \beta_c).$$

Let λ_{ij}^m be the optimal dual variable corresponding to the demand satisfaction constraint

(4.16b) *for OD pair* (i, j)*. The optimal monopoly prices* ℓ_{ij}^m *are:*

$$\ell_{ij}^m = \begin{cases} \frac{\lambda_{ij}^m + \sqrt{(\lambda_{ij}^m)^2 + 6\sigma(1-\sigma)\ell_{\max}^2}}{3} & , \frac{\lambda_{ij}^m}{\ell_{\max}} < \frac{3-5\sigma}{2} \\ \frac{(1+\sigma)\ell_{\max} + 2\lambda_{ij}^m}{4} & , \frac{3-5\sigma}{2} \leq \frac{\lambda_{ij}^m}{\ell_{\max}} < \frac{3\sigma-1}{2} \\ \frac{2\lambda_{ij}^m + \ell_{\max}}{3} & , \frac{3\sigma-1}{2} \leq \frac{\lambda_{ij}^m}{\ell_{\max}} \leq 1. \end{cases} \tag{4.17}$$

These prices can be upper bounded by:

$$\ell_{ij}^m \leq \begin{cases} \frac{\overline{\lambda}_{ij} + \sqrt{(\overline{\lambda}_{ij})^2 + 6\sigma(1-\sigma)\ell_{\max}^2}}{3} & , \frac{\overline{\lambda}_{ij}}{\ell_{\max}} < \frac{3-5\sigma}{2} \\ \frac{(1+\sigma)\ell_{\max} + 2\overline{\lambda}_{ij}}{4} & , \frac{3-5\sigma}{2} \leq \frac{\overline{\lambda}_{ij}}{\ell_{\max}} < \frac{3\sigma-1}{2} \\ \frac{2\overline{\lambda}_{ij} + \ell_{\max}}{3} & , \frac{3\sigma-1}{2} \leq \frac{\overline{\lambda}_{ij}}{\ell_{\max}} \leq 1. \end{cases} \tag{4.18}$$

The proof can be found in Appendix C.2.1. We can interpret the dual variables λ_{ij}^m as the cost of providing a single ride between i and j to the platform. In the worst case scenario, every single requested ride from node i requires rebalancing and charging both at the origin and the destination. Hence the upper bounds on (4.18) include the operational costs of passenger-carrying, rebalancing and charging vehicles (both at the origin and the destination); and the energy costs of both passenger-carrying and rebalancing trips multiplied by the price of electricity at the trip destinations (This is exactly what $\overline{\lambda}_{ij}$ consists of).

Similar to the taxes applied on products, whose burden is shared among the supplier and the customer; the costs associated with rides are shared among the platform operator and the riders (which is why the price paid by the riders include some fraction of the cost of the ride).

We note that if the optimal dual variables λ_{ij}^m fall in the region $[(3-5\sigma)\ell_{\max}/2, (3\sigma-1)\ell_{\max}/2]$, then the optimal prices given by (4.17) fall in the region $[(1-\sigma)\ell_{\max}, \sigma\ell_{\max}]$. In this region, the demand function $D(\ell_{ij}^m, \infty)$ is linear. Hence, the optimization problem (4.16) (with the additional constraint $(1-\sigma)\ell_{\max} \leq \ell_{ij}^1 \leq \sigma\ell_{\max},\ \forall i, j \in \mathcal{N}$,

without losing global optimality) becomes a convex quadratic program and can be solved in polynomial time. The following assumptions guarantee this:

Assumption 4.3.1 *Assume that $\sigma \geq 3/5$, i.e., the customers' preferences over the two firms are highly correlated.*

Assumption 4.3.2 *We assume $\max_{i,j} \overline{\lambda}_{ij} \leq \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)} \ell_{\max}$ as an upper bound on the maximum cost of a ride in the network.*

Remark 4.3.1 *Assumption 4.3.1 implies that at least $3/5$ (60%) of the customers' valuations between the firms are correlated. Higher correlation implies that riders' valuations of the rides provided by a firm depend less on the identity of the firm. This is reasonable for ride-sharing platforms, where majority of the customers decide depending heavily on the price rather than the identity of the firm.*

Assumption 4.3.2 imposes an upper bound on the maximum cost of a ride. This can be satisfied in practice, especially with electric vehicles. Observe that the bound is increasing with σ, hence it is tightest when $\sigma = 3/5$. To give numbers with a simple calculation, consider a network with farthest OD pair of 15 miles and 30 minutes away (with average speed $30mph$), $\sigma = 3/5$ and $\ell_{\max} = \$50$. An average EV consumes 34kWh energy to drive for 100 miles. For an average price of electricity of $\$0.11$ per kWh and a charger with $20kW$ charging speed, the EV charges $10kWh$ in 30 minutes for $\$1.1$, that allows for 30 miles of range. If we amortize the cost of a very expensive EV of $\$100k$ over 5 years, we get per minute operational cost of $\$0.04$. In total, to do the trip and the rebalancing, the vehicle drives for 30 miles for 1 hour and charges for 30 minutes. In total, this yields a cost of $90 \times \$0.04 + \$1.1 = \$4.7 = \max_{i,j} \overline{\lambda}_{ij} \leq \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)} = \$7.5$. Whereas the fuel for gasoline vehicles costs about 4 times more (around $\$0.16$ per mile), which would yield $\max_{i,j} \overline{\lambda}_{ij} = \$8.00$.

Next, we relate the optimal prices ℓ_{ij}^m to the profits generated by the operator and the consumer surplus. The profits are defined by the objective function in (4.16a). The consumer

surplus is defined as the difference between the price that customers pay and the price that they are willing to pay, i.e., the aggregate pay-off of the customers.

Proposition 4.3.2 *Suppose that Assumptions 4.3.1 and 4.3.2 hold. With the optimal monopoly prices ℓ_{ij}^m, the profits per period are:*

$$P^m = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\theta_{ij}}{4\sigma\ell_{\max}}(\ell_{\max}(1+\sigma) - 2\ell_{ij}^m)^2. \tag{4.19}$$

The consumer surplus with the optimal prices is:

$$\text{CS}^m = \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij} \frac{\ell_{\max}(\sigma^2+\sigma+1) - 3\ell_{ij}^m(1+\sigma-\frac{\ell_{ij}^m}{\ell_{\max}})}{6\sigma}. \tag{4.20}$$

The proof can be found in Appendix C.2.2. Notice that the profits in (4.19) are decreasing as the prices for rides increase. Thus expensive rides generate less profits compared to the cheaper rides and it is more beneficial if the optimal dual variables λ_{ij}^m are small and prices are close to $\ell_{\max}(1+\sigma)/4$. Thus, the operator has incentive to use more efficient routing and charging policies so they can lower ride prices as much as possible. Moreover, by computing $\frac{\partial \text{CS}^m}{\partial \ell_{ij}^m}$ using (4.20), one identifies that lower prices generate higher consumer surplus, which is an intuitive result.

4.3.2.2 Duopoly Static Planning Problem

We study the duopoly as a game between two firms. At a high level, the game is described by firm f observing firm $-f$'s prices and solving the optimization problem (4.16) (by considering firm $-f$'s prices to be ℓ_{ij}^{-f} rather than ∞ for the demand function). We consider two competitors with identical operational costs β_t and β_c, and study the optimal pricing strategy when the firms are at an equilibrium. In an equilibrium, no firm benefits from unilaterally changing the prices for any number of OD pairs (and as a result the optimal solution to their

static planning problem). Given $\{\ell_{ij}^{-f}\}_{\forall i,j\in\mathcal{N}}$, the best response of firm f is the best pricing, routing and charging strategy of f, which is the solution of (4.16) (with ℓ_{ij}^{-f} instead of ∞ in the demand function). Since the operational costs and the prices of electricity are identical for both of the firms, their best response to their competitor's prices are the same. As such, it is intuitive that there exists an equilibrium in which both firms set the prices equal ($\ell_{ij}^{f} = \ell_{ij}^{-f}, \forall i,j \in \mathcal{N}$), and we show that this is in fact the case. Such an equilibrium is commonly referred to as a *symmetric* duopoly equilibrium. Furthermore, we show that no asymmetric equilibria can exist under this setting, i.e., identical firms will not set different prices for the same OD pair at equilibrium.

Let the following static planning problem characterize the state in which both firms serve equal number of customers for all OD pairs and have identical pricing strategies:

$$\max_{x_{ic}^e, x_{ij}^e, \ell_{ij}^1} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij}\ell_{ij}^1 D(\ell_{ij}^1, \ell_{ij}^2)\Big|_{\ell_{ij}^1=\ell_{ij}^2} - \sum_{i=1}^{n}\sum_{e=0}^{e_{\max}-1}(\beta_c + p_i)x_{ic}^e - \beta_t \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e \tau_{ij} \tag{4.21a}$$

$$\text{subject to} \quad \theta_{ij} D(\ell_{ij}^1, \ell_{ij}^2)\Big|_{\ell_{ij}^1=\ell_{ij}^2} \leq \sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e, \quad \forall i,j \in \mathcal{N}, \tag{4.21b}$$

$$(4.16c) - (4.16g).$$

We note that the optimization problem (4.21) is in general non-convex due to $D(\ell_{ij}^1, \ell_{ij}^2)$. Since there are no constraints on the fleet size and furthermore prices that control the demand are decision variables, a feasible solution to the above optimization problem always exists. Moreover, the optimal solution to (4.21) specifies an equilibrium of the duopoly.

Proposition 4.3.3 *Suppose that $\sigma \geq 1/2$ and Assumption 4.3.2 holds. The firms are in an equilibrium when their routing, charging, and symmetric pricing strategy follows the solution of* (4.21).

Proof outline: We first determine the optimal pricing strategy $\{\ell_{ij}^d\}_{i,j\in\mathcal{N}}$ of (4.21) using the first and second order optimality conditions (similar to proof of Proposition 1). Then, by stating the first order optimality condition for firm f we show that when firm $-f$ sets prices as $\ell_{ij}^{-f} = \ell_{ij}^d$, $\forall i, j \in \mathcal{N}$, then the best response of firm f is to set $\ell_{ij}^f = \ell_{ij}^d$, $\forall i, j \in \mathcal{N}$. Hence, they are in an equilibrium. □

The complete proof can be found in Appendix C.2.3. Accordingly, there exists a duopoly equilibrium characterized as the optimal solution of (4.21), in which the firms set identical prices. The optimal solution to (4.21) is however not necessarily unique and there can be many solutions yielding the same profits. For instance, if $p_i = p_j, \forall i, j \in \mathcal{N}$, then the optimal charging strategy is not unique. We let $\{\ell_{ij}^d\}_{i,j\in\mathcal{N}}$ to be the equilibrium prices determined as an optimal solution of (4.21) and say that the firms are in a symmetric duopoly equilibrium as long as $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d, \forall i, j \in \mathcal{N}$. Furthermore, in the next proposition, we state that if both firms serve all OD pairs, equilibrium prices can not be asymmetric.

Proposition 4.3.4 *Suppose that $\sigma \geq 1/2$ and Assumption 4.3.2 holds. There exists no asymmetric equilibrium prices, in which both firms serve nonzero demand for all OD pairs with nonzero potential riders.*

Proof outline: We let $\ell_{ij}^1 = \ell_{ij}^2 + \delta$ for some $\delta > 0$ and show by contradiction that the first-order optimality condition can not simultaneously be satisfied for both firms. Since the demand function $D(\ell_{ij}^1, \ell_{ij}^2)$ has different expressions for $\ell_{ij}^1 \leq (1-\sigma)\ell_{\max}$ and $\ell_{ij}^1 > (1-\sigma)\ell_{\max}$, we separately study three cases: (i) $\ell_{ij}^1, \ell_{ij}^2 \leq (1-\sigma)\ell_{\max}$, (ii) $\ell_{ij}^1, \ell_{ij}^2 > (1-\sigma)\ell_{\max}$, and (iii) $\ell_{ij}^1 > (1-\sigma)\ell_{\max}$, $\ell_{ij}^2 \leq (1-\sigma)\ell_{\max}$. For all cases, we first assume that the first-order optimality condition hold for both firms and bound the difference between the dual variables leading to ℓ_{ij}^1 and ℓ_{ij}^2 in terms of δ. For cases (i) and (ii), we show by using the bound on the dual variables that if the first-order condition for firm 2 is satisfied (i.e., is equal to 0), then the first-order condition for firm 1 is always less than 0, which is a contradiction. For case (iii), we

show that with first-order condition satisfying prices, $\ell_{ij}^2 + \delta$ is always less than ℓ_{ij}^1, which is a contradiction. □

The complete proof is provided in Appendix C.2.4. As we have identified that the duopoly can only be in a symmetric equilibrium, we analyze the effects of competition in state of a symmetric equilibrium.

The next set of results characterize the effects of competition on the ride prices, the operators' profits, the total societal ride demand served, and the consumer surplus. In the first result, we provide lower and upper bounds on the price reduction the customers will see with the introduction of the second firm and moving from a monopoly to a symmetric duopoly equilibrium.

Proposition 4.3.5 *Suppose that Assumptions 4.3.1 and 4.3.2 hold. Let $\overline{\lambda}_{ij}$ be defined as in Proposition 4.3.1. Define*

$$\Delta_1(\lambda_{ij}) := 4\ell_{\max}^2 + (2\lambda_{ij} + (15\sigma - 3)\ell_{\max})(2\lambda_{ij} + (1-\sigma)\ell_{\max}),$$

$$\Delta_2(\lambda_{ij}) := 2(\sigma\ell_{\max} - \lambda_{ij})^2 + 2(\ell_{\max} - \lambda_{ij})^2 + 11(\sigma-1)^2\ell_{\max}^2.$$

Let λ_{ij}^d be the optimal dual variable corresponding to the demand satisfaction constraint (4.21b)*. The symmetric duopoly equilibrium prices are determined as:*

$$\ell_{ij}^d = \begin{cases} \frac{(3-5\sigma)\ell_{\max} + 2\lambda_{ij}^d + \sqrt{\Delta_1(\lambda_{ij}^d)}}{8} & , \frac{\lambda_{ij}^d}{\ell_{\max}} \leq \frac{3(1-\sigma)^2}{2(\sigma+1)} \\ \frac{(5-3\sigma)\ell_{\max} + 2\lambda_{ij}^d - \sqrt{\Delta_2(\lambda_{ij}^d)}}{4} & , \text{o.w.}, \end{cases} \tag{4.22}$$

Moreover, denote the difference between optimal monopoly and symmetric duopoly equilibrium

prices for OD pair (i,j) *as* $\Delta\ell_{ij} := \ell^m_{ij} - \ell^d_{ij}$. *Then:*

$$\Delta\ell_{ij} \geq \begin{cases} \frac{(7\sigma-1)\ell_{\max}-2\overline{\lambda}_{ij}-\sqrt{\Delta_1(\overline{\lambda}_{ij})}}{8} & , \frac{\overline{\lambda}_{ij}}{\ell_{\max}} \leq \frac{3(1-\sigma)^2}{2(\sigma+1)} \\ \frac{(4\sigma-4)\ell_{\max}-2\overline{\lambda}_{ij}+\sqrt{\Delta_2(\overline{\lambda}_{ij})}}{4} & , \text{o.w.}, \end{cases} \tag{4.23}$$

and

$$\Delta\ell_{ij} \leq \frac{(7\sigma-1)\ell_{\max}+4\overline{\lambda}_{ij}-\ell_{\max}\sqrt{-15\sigma^2+18\sigma+1}}{8}. \tag{4.24}$$

Proof outline: We state the first and the second order optimality conditions on (4.21) to get the duopoly equilibrium prices. To lower bound the price difference, we evaluate the monopoly prices at $\lambda^m_{ij} = 0$ and the duopoly equilibrium prices at $\lambda^d_{ij} = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ (and to upper bound, vice versa). □

The complete proof can be found in Appendix C.2.5. An interesting observation is how σ affects the prices. For the optimal monopoly prices, $\partial\ell^m_{ij}/\partial\sigma > 0$, i.e., the monopolist serving a population with higher σ charges more for the rides with identical costs (i.e., identical λ^m_{ij}). The reason is that larger σ shifts the distribution of customers' valuations for the monopolist from intermediate to extreme values (as σ increases from $1/2$ to 1, the distribution shifts from triangular to uniform). This shift in the distribution modifies the demand function $D(\ell^1_{ij}, \infty)$, which leads to an increase on the optimal prices. Simply put, larger σ, i.e., lack of firm loyalty, leads to an increase in the prices for the monopoly. On the contrary for the duopoly equilibrium prices, $\partial\ell^d_{ij}/\partial\sigma < 0$. That is, the duopoly serving a population with higher σ charges less for the rides with identical costs (i.e., identical λ^d_{ij}). The intuition behind is that larger σ indicates a lack of firm loyalty (when $\sigma = 1$, the customers buy from the firm that offers lower prices). Hence, higher σ strengthens the competition and causes the firms to charge less. The reader can observe that when $\sigma = 1$, $\ell^d_{ij} = \lambda^d_{ij}$, i.e., the equilibrium prices are equal to the costs of providing the rides to the platform, which is the lowest the firms can go without losing money but make no profit.

Observe that the lower bounds in (4.23) are decreasing functions of $\overline{\lambda}_{ij}$. Given the maximum value of $\overline{\lambda}_{ij}$ equal to $\overline{\lambda}_{ij} = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$, the lower bound on the price difference is 0. Hence, we can conclude that the duopoly prices are never higher than the monopoly prices, for all OD pairs.

Proposition 4.3.5 characterizes the effect of competition on the prices depending on the network parameters and therefore the dual variables. The next series of results aim to determine universal bounds on the ratio of prices, induced demand, profits and consumer surplus in the monopoly and the duopoly, *independent of the network parameters*.

Proposition 4.3.6 (Price Bounds) *Suppose that Assumptions 4.3.1 and 4.3.2 hold. For all OD pairs, the optimal monopoly prices obey the following:*

$$2\ell_{\max}/5 \leq \underline{\ell}^m \leq \ell_{ij}^m \leq \overline{\ell}^m \leq 3\ell_{\max}/4, \tag{4.25}$$

where $\underline{\ell}^m := \frac{1+\sigma}{4}\ell_{\max}$ *and* $\overline{\ell}^m := \frac{7+14\sigma-9\sigma^2}{40-24\sigma}\ell_{\max}$. *Furthermore, the symmetric duopoly equilibrium prices obey:*

$$0 \leq \underline{\ell}^d \leq \ell_{ij}^d \leq \overline{\ell}^d \leq \ell_{\max}/2, \tag{4.26}$$

where $\underline{\ell}^d := \frac{3-5\sigma+\sqrt{-15\sigma^2+18\sigma+1}}{8}\ell_{\max}$ *and* $\overline{\ell}^d := \frac{1+\sigma}{4}\ell_{\max}$. *Moreover for all OD pairs* (i,j)*, the ratio between the symmetric duopoly equilibrium prices and the optimal monopoly prices obey the following:*

$$\frac{\underline{\ell}^d}{\overline{\ell}^m} \leq \frac{\ell_{ij}^d}{\ell_{ij}^m} \leq \frac{\overline{\ell}^d}{\underline{\ell}^m} = 1. \tag{4.27}$$

Proof outline: The proof is done by evaluating the optimal monopoly prices given by (4.17) at $\lambda_{ij}^m = 0$ and $\lambda_{ij}^m = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ as well as the duopoly equilibrium prices given by (4.22) at $\lambda_{ij}^d = 0$ and $\lambda_{ij}^d = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ to get the bounds on the prices in terms of σ. Then, we impose the condition $\sigma \in [3/5, 1]$ to get the uniform bounds.

The complete proof can be found in Appendix C.2.6. An observation is that increasing

σ increases both the upper and the lower bounds for the optimal monopoly prices, whereas decreases the lower bound on the duopoly equilibrium prices and increases the upper bound. This is because for the optimal monopoly prices, $\partial \ell_{ij}^m / \partial \sigma > 0$. However, because it strengthens the competition between the firms in the duopoly, it can cause the prices to go much lower, hence decreasing the lower bound (when $\sigma = 1$: if $\lambda_{ij}^d = 0$, then $\ell_{ij}^d = 0$). The upper bound on the duopoly equilibrium prices still increases, because according to Assumption 4.3.2 a larger σ permits a larger λ_{ij}^d and hence higher prices. Consequently, the upper bound on the price ratio is always 1 independent of σ while the lower bound is decreasing with σ.

The next result characterizes the effect of competition on the total customer demand for rides that are served by either firm. We show that the aggregate demand served by the duopoly is at least equal to and can be up to 4 times higher than the demand served by the monopoly.

Proposition 4.3.7 (Demand Bounds) *Suppose that Assumptions 4.3.1 and 4.3.2 hold. For all OD pairs (i, j), the monopoly demand functions evaluated at the optimal monopoly prices obey:*

$$1/4 \leq \underline{D}^m \leq D(\ell_{ij}^m, \infty) \leq \overline{D}^m \leq 2/3, \tag{4.28}$$

where $\underline{D}^m := \frac{13-3\sigma^2-6\sigma}{40\sigma-24\sigma^2}$ and $\overline{D}^m := \frac{1+\sigma}{4\sigma}$. The duopoly demand functions at the duopoly equilibrium prices obey:

$$1/4 \leq \underline{D}^d \leq D(\ell_{ij}^d, \ell_{ij}^d) \leq \overline{D}^d \leq 1/2, \tag{4.29}$$

where $\underline{D}^d := \frac{1}{4\sigma}$ and $\overline{D}^d := \frac{1}{2} - \frac{\left(-(1+\sigma)+\sqrt{-15\sigma^2+18\sigma+1}\right)^2}{128\sigma(1-\sigma)}$. Furthermore, the ratio between the total demand served between any OD pair $\frac{2D(\ell_{ij}^d, \ell_{ij}^d)}{D(\ell_{ij}^m, \infty)}$ obeys the following:

$$1 \leq \frac{2}{1+\sigma} = \frac{2\underline{D}^d}{\overline{D}^m} \leq \frac{2D(\ell_{ij}^d, \ell_{ij}^d)}{D(\ell_{ij}^m, \infty)} \leq \frac{2\overline{D}^d}{\underline{D}^m} \leq 4 \tag{4.30}$$

Proof outline: The proof is done by evaluating the demand functions for the monopoly and the duopoly at the price bounds given by (4.25) and (4.26), and then imposing the condition $\sigma \in [3/5, 1]$ to get uniform bounds.

The complete proof can be found in Appendix C.2.7. Taking into account that induced demand is inversely proportional to prices, the impact of σ on the demand function bounds is in accordance with price bounds in Proposition 4.3.6.

Remark 4.3.2 *The upper bound in* (4.30) *is achieved when* $\sigma = 1$, $\lambda_{ij}^m = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ *and* $\lambda_{ij}^d = 0$. *Although it is achievable for some OD pairs, it is not possible to achieve it for all OD pairs simultaneously. This is because for* λ_{ij}^d *to be 0, constraint* (4.21b) *has to be slack, meaning node* i *has excess supply of vehicles that are being rebalanced to node* j. *This however can not hold simultaneously for all OD pairs, since that would mean there are empty vehicles being routed between all OD pairs, which would not be optimal.*

Interestingly, we see that this potential increase in the aggregate demand never translates into a profit increase for the firms because of the competition. As expected, profits decrease in the presence of competition. According to the next result, the profits generated by a single firm in duopoly is always less than 85% of the profits generated by the monopoly.

Proposition 4.3.8 (Profit Bounds) *Suppose that Assumptions 4.3.1 and 4.3.2 hold. Let profits earned by serving the induced demand between OD pair* (i, j) *in the monopoly be* P_{ij}^m. *With the optimal monopoly prices,* P_{ij}^m *for all* (i, j) *obey the following:*

$$\theta_{ij}\ell_{\max}/16 \leq \theta_{ij}\underline{P}^m \leq P_{ij}^m \leq \theta_{ij}\overline{P}^m \leq \theta_{ij}\ell_{\max}/4, \tag{4.31}$$

where

$$\underline{P}^m = \frac{(3\sigma^2+6\sigma-13)^2}{64\sigma(5-3\sigma)^2}\ell_{\max}, \quad \overline{P}^m = \frac{(1+\sigma)^2}{16\sigma}\ell_{\max}.$$

Similarly, let profits earned by serving the induced demand between OD pair (i,j) by a single firm in the duopoly be P^d_{ij}. With the duopoly equilibrium prices, P^d_{ij} for all (i,j) obey:

$$0 \leq \theta_{ij}\underline{P}^d \leq P^d_{ij} \leq \theta_{ij}\overline{P}^d \leq (4+\sqrt{10})\ell_{\max}\theta_{ij}/48 \tag{4.32}$$

where

$$\underline{P}^d = \left(\overline{\ell}^d - \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}\right) \times \underline{D}^d = \frac{1-\sigma}{2\sigma(5-3\sigma)}\ell_{\max},$$

$$\overline{P}^d = \underline{\ell}^d \overline{D}^d.$$

Furthermore, for all OD pairs, the ratio $\frac{P^d_{ij}}{P^m_{ij}}$ obeys:

$$\frac{8(1-\sigma)}{(\sigma+1)^2(5-3\sigma)} = \frac{\underline{P}^d}{\overline{P}^m} \leq \frac{P^d_{ij}}{P^m_{ij}} \leq \frac{\overline{P}^d}{\underline{P}^m} \lessapprox 0.85. \tag{4.33}$$

Proof outline: The proof is done by evaluating the profits for the monopoly given by (4.19) at the price bounds given by (4.25). For the duopoly, we first derive the dual objective, show that it decreases with ℓ^d_{ij}, and evaluate at the duopoly equilibrium price bounds given by (4.26). Then, we impose the condition $\sigma \in [3/5, 1]$ to get the uniform bounds.

The complete proof can be found in Appendix C.2.8. Since lower prices generate more profits in the monopoly and the price bounds are increasing with σ, the profit bounds of the monopoly are decreasing with σ. Similarly, the duopoly profit bounds are decreasing with σ too. Since σ increases the upper bound on prices, the lower bound on the profits decrease. However, although σ decreases the lower bound on the prices, the upper bound on the profits still decrease. This is because competition in the duopoly is a downward driving force on the prices. Consequently, lower prices in the duopoly do not only result from lower λ^d_{ij}, but also stronger competition. Hence, although lower prices increase the aggregate demand, because

the firms are now competing over the customers, neither of the firms serve enough customers to compensate for the decrease in the prices. Hence, the profits decrease.

The upper bound in (4.33) is achieved when $\sigma = 3/5$, $\lambda_{ij}^m = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$, and $\lambda_{ij}^d = 0$. Due to the same argument in Remark 4.3.2, it can not be achieved simultaneously by all the OD pairs. Consequently, the ratio of total profits can not achieve this upper bound with equality.

How do the customers benefit from the introduction of competition? We saw that a reduction in ride prices is expected. Next, we show that the consumer surplus in the duopolistic setting is at least equal to and can be up to 16 times the consumer surplus in the monopoly.

Proposition 4.3.9 (Consumer Surplus Bounds) *Suppose that Assumptions 4.3.1 and 4.3.2 hold. Let the consumer surplus of customers requesting a ride between OD pair (i,j) in the monopoly be CS_{ij}^m. With the optimal monopoly prices, CS_{ij}^m for all (i,j) obey:*

$$\theta_{ij}\frac{\ell_{\max}}{32} \leq \theta_{ij}\underline{\mathrm{CS}}^m \leq \mathrm{CS}_{ij}^m \leq \theta_{ij}\overline{\mathrm{CS}}^m \leq \theta_{ij}\frac{13}{90}\ell_{\max}, \tag{4.34}$$

where

$$\underline{\mathrm{CS}}^m = \frac{171\sigma^4 - 660\sigma^3 + 1378\sigma^2 - 1748\sigma + 907}{384\sigma(5-3\sigma)^2}\ell_{\max},$$

$$\overline{\mathrm{CS}}^m = (7\sigma^2 - 2\sigma + 7)\ell_{\max}/(96\sigma).$$

Similarly, let the consumer surplus of customers requesting a ride between OD pair (i,j) in the duopoly be CS_{ij}^d. With the duopoly equilibrium prices, CS_{ij}^d for all (i,j) obey:

$$\theta_{ij}\frac{\ell_{\max}}{8} \leq \theta_{ij}\underline{\mathrm{CS}}^d \leq \mathrm{CS}_{ij}^d \leq \theta_{ij}\overline{\mathrm{CS}}^d \leq \theta_{ij}\frac{\ell_{\max}}{2}, \tag{4.35}$$

where

$$\underline{\mathrm{CS}}^d = (\sigma^2 - 2\sigma + 13)\ell_{\max}/(96\sigma),$$

$$\overline{\mathrm{CS}}^d = \frac{\ell_{\max}}{24\sigma(1-\sigma)}\Big((2\sigma)^3 - (\sigma+1-2\frac{\underline{\ell}^d}{\ell_{\max}})^3 - 24\sigma(1-\frac{\underline{\ell}^d}{\ell_{\max}})(\sigma-1+\frac{\underline{\ell}^d}{\ell_{\max}})\Big).$$

Furthermore, for all OD pairs, the ratio $\frac{\mathrm{CS}_{ij}^d}{\mathrm{CS}_{ij}^m}$ *obeys:*

$$1 \leq \frac{\sigma^2-2\sigma+13}{7\sigma^2-2\sigma+7} \leq \frac{\underline{\mathrm{CS}}^d}{\overline{\mathrm{CS}}^m} \leq \frac{\mathrm{CS}_{ij}^d}{\mathrm{CS}_{ij}^m} \leq \frac{\overline{\mathrm{CS}}^d}{\underline{\mathrm{CS}}^m} \leq 16. \tag{4.36}$$

Proof outline: The proof is done by evaluating the consumer surplus for the monopoly given by (4.20) at the price bounds given by (4.25). For the duopoly, we compute the consumer surplus at the price bounds given by (4.26) in a similar fashion to the the proof of Proposition 4.3.2. Then, we impose the condition $\sigma \in [3/5, 1]$ to get the uniform bounds.

The complete proof can be found in Appendix C.2.9. Considering the fact that lower prices (both in the duopoly and the monopoly) increase the consumer surplus by inducing more customers and increasing the surplus per customer, the dependency of the price bounds on σ reflects to the consumer surplus bounds.

Remark 4.3.2 applies for the upper bound in (4.36) too, and thus it can not be achieved for all OD pairs simultaneously. Therefore, the ratio of total consumer surplus cannot achieve this upper bound with equality.

So far, we have studied the effects of competition in an electric AMoD system by adopting a static network-flow formulation. Although very convenient for analysis, this formulation does not reflect the randomness in arrivals nor constrains vehicles dispatch decisions to be integer valued (e.g., 0.25 customer may be served). To address these discrepancies with the real environment, in the next subsection, we modify our model to account for the randomness in arrivals and furthermore design a control policy that can be implemented in real-time.

4.3.3 Real-Time Control

To accommodate for the stochastic nature of the arrivals, we model the arrival of the potential customers OD pair (i, j) as a Poisson process with an arrival rate of θ_{ij}. Moreover, we allow the firms to set prices real-time and use the same price-responsive demand model. In particular, during period t, for a price tuple $(\ell^1_{ijt}, \ell^2_{ijt})$ for OD pair (i, j), the induced arrival rate for firm f is given by $\Theta^f_{ijt} = \theta_{ij} D(\ell^f_{ijt}, \ell^{-f}_{ijt})$. Thus, the number of new ride requests in time period t for firm f is $A^f_{ijt} \sim \text{Pois}(\Theta^f_{ijt})$ for OD pair (i, j). As a consequence of this randomness in the customer arrivals, the platform operator might not be able to assign every customer to a ride immediately (if the number of induced arrivals exceed the number of available vehicles). In order to address this nuance, we adopt the following ride-sharing model:

Ride Hailing Model: Customers that purchase a ride during period t are not immediately matched with a ride, but enter the queue for OD pair (i, j) to be served at the beginning of period $t+1$. After the platform operator executes routing decisions for the fleet at the beginning of period $t + 1$, the customers in the queue for OD pair (i, j) are matched with rides and served on a first-come, first-served basis.

Under these additional modeling modifications, our goal is to establish a real-time pricing and fleet management policy that can be implemented in a real environment and provides stability of the queues[11]. In fact, the model studied in Subsection 4.3.2 is the static planning problem associated with this real environment, where we ignored the stochasticity of the arrivals and used the expected values, while allowing the vehicle routing decisions to be flows (real numbers) rather than integers. For the monopoly (or the symmetric duopoly), the solution to this static planning problem in (4.16) (or (4.21)) is the optimal static policy that consists of optimal prices as well as optimal vehicle routing and charging decisions. This policy can not directly be implemented in a real environment because it does not yield integer-valued solu-

[11]The stability condition that we are interested in is rate stability of all queues. A queue for OD pair (i, j) is rate stable if $\lim_{t\to\infty} q_{ij}(t)/t = 0$.

tions. In Section 4.2, it was proven that randomizing the vehicle decisions according to the optimal solution of the static problem to get integer-valued actions guarantees the stability of the queues. However, considering random arrivals, this method may not execute the most profitable actions since it does not take the real-time queue lengths into consideration. Although it guarantees stability of the queues, it does not seek to minimize the queue lengths and hence the wait time of the passengers, which would negatively affect the business.

Instead of using the randomized solution to implement real-time actions, it is possible to realize a real-time policy that acknowledges the queue lengths and hence aims to maximize the profits while minimizing the total wait time of the customers. To achieve this, we propose to apply finite-horizon model predictive control (MPC) in our numerical experiment (albeit with no performance guarantee).

MPC Procedure: The idea of finite-horizon MPC is to observe the current state of the environment and determine the best control strategy for a planning horizon of T by predicting the state path of the environment. Then, only the control strategy at the initial time period is implemented and the process is repeated. Specifically, let $\mathcal{S}$ be the state of the vehicles (locations, energy levels) and $\{Q_{ij}\}_{i,j\in\mathcal{N}}$ be the outstanding customer demand (i.e., people who have requested a ride but not yet served) at the beginning of planning time. The MPC Algorithm is summarized as follows:

Algorithm 7: MPC Procedure

1: $\mathcal{S} \leftarrow$ Get vehicle states (locations, energy levels)
2: $Q_{ij} \leftarrow$ Count outstanding customers
3: $\{x^e_{ijt}, x^e_{ict}, \ell_{ijt}\}_{\forall i,j,e,t} \leftarrow$ Solve (4.37)
4: Execute $\{x^e_{ij0}, x^e_{ic0}, \ell_{ij0}\}_{\forall i,j,e}$

At each period, Algorithm 7 is run and the system state is observed. Using this information, the optimal fleet management and pricing strategy is computed for the next T periods by solving (4.37). Vehicle routing/charging and pricing decisions are executed for the initial time

period and the environment transitions into next state. Then, Algorithm 7 is re-run and this process is repeated during the entire operation of the system.

Next, we state the optimization problem (4.37) for the controller using a dynamic pricing scheme in monopoly. Let the decision variable ℓ^1_{ijt} be the price for rides between OD pair (i, j) in period t, x^e_{ijt} be the number of vehicles at node i with energy level e being routed to node j in period t, x^e_{ict} be the number of vehicles charging at node i starting with energy level e in period t, and q_{ijt} be the people waiting in the queue for OD pair (i, j) in period t. We state the problem as follows:

$$\max_{x^e_{ict}, x^e_{ijt}, q_{ijt}, \ell^1_{ijt}} \quad \sum_{ijt} \ell^1_{ijt}\theta_{ij} D(\ell^1_{ijt}, \infty) - \sum_{ijt} w_{ijt} q_{ijt} - \beta_t \sum_{ijet} \tau_{ij} x^e_{ijt} - \sum_{iet} (\beta_c + p_i) x^t_{ic} \tag{4.37a}$$

$$\text{s.t.} \quad q_{ijt_0} \geq Q_{ij} - \sum_e x^e_{ijt_0}, \quad \forall i, j \in \mathcal{N} \tag{4.37b}$$

$$q_{ijt} \geq q_{ijt-1} + \theta_{ij} D(\ell^1_{ijt-1}, \infty) - \sum_e x^e_{ijt}, \quad \forall i, j \in \mathcal{N},\ \forall t > t_0, \tag{4.37c}$$

$$\sum_j x^e_{ijt} + x^e_{ict} - \sum_j x^{e+e_{ji}}_{jit-\tau_{ji}} - x^{e-1}_{ict-1} = s^e_{it}, \quad \forall i \in \mathcal{N},\ \forall e \in \mathcal{E}, \quad \forall t \geq t_0 \tag{4.37d}$$

$$x^{e_{\max}}_{ict} = 0, \quad \forall i \in \mathcal{N},\ \forall t \geq t_0, \tag{4.37e}$$

$$x^e_{ijt} = 0, \quad \forall e < e_{ij}, \forall i, j \in \mathcal{N}, \forall t \geq t_0, \tag{4.37f}$$

$$x^e_{ijt}, x^e_{ict}, q_{ijt} \geq 0,\ x^e_{ijt}, x^e_{ict} \in \mathbb{N}, \quad \forall i, j \in \mathcal{N},\ \forall e \in \mathcal{E},\ \forall t \geq t_0, \tag{4.37g}$$

$$x^e_{ijt} = x^e_{ict} = 0, \forall e \notin \mathcal{E}, \forall t < t_0, \forall i, j \in \mathcal{N}. \tag{4.37h}$$

The first term in the objective function (4.37a) corresponds to the expected revenue gained by setting prices ℓ^1_{ijt}. The second term assigns a cost to the queue lengths, where w_{ijt} is the cost per person in the queue for OD pair (i, j) at the time period t. The third term is the operational costs of the trip-making vehicles, and the last term is the operational and the charging costs of

the charging vehicles. Hence, the objective is to maximize the profits minus the queue penalty.

The state variable s^e_{it} denotes the number of vehicles at node i with energy level e, at the beginning of time period t. At the beginning of the planning time $t = t_0$, $s^e_{it_0}$ is simply the number of available vehicles at node i with energy level e. For $t > t_0$, s^e_{it} denotes the number of vehicles that will be available at the beginning of time period t, at node i with energy level e. These are the vehicles that are en route to another node at the time of planning. Hence, (4.37d) is the vehicle balance constraint. The constraints (4.37c) along with the non-negativity constraint (4.37g), implement the queue length transition $q_{ijt} = \max\{0, q_{ijt-1} + \theta_{ij} D(\ell^1_{ijt-1}, \infty) - \sum_e x^e_{ijt}\}$ as two linear inequalities. For $t = t_0$, the queue length is modified via (4.37b), where Q_{ij} denotes the number of passengers waiting to be served at the planning time.

The MPC controller using a dynamic pricing scheme for the duopoly can be stated in a similar way to the monopoly. We exclude it here and refer the reader to the Appendix C.2.10.

We end this subsection by noting that it is possible to implement a model predictive controller with static prices in monopoly simply by adding the constraint $\ell^1_{ijt} = \ell^m_{ij}, \forall t \geq t_0$ to (4.37). For the duopoly, we replace $D(\ell^1_{ijt}, \infty)$ with $D(\ell^1_{ijt}, \ell^2_{ijt})$ and add the constraint $\ell^1_{ijt} = \ell^2_{ijt} = \ell^d_{ij}, \forall t \geq t_0$.

4.3.4 Numerical Study

In this subsection, we discuss the effects of competition and the performances of the real-time controllers via numerical examples. To solve the optimization problems we used the Gurobi Optimizer [192].

In our discrete-time system, we chose one period to be equal to $\Delta t = 5$ minutes, which is equal to the time it takes to deliver one unit of battery energy. We chose operational costs of $\beta_t = \$0.2$ and $\beta_c = \$0.1$ (by taking the amortized average price of an electric car over 5 years [169] as a reference), maximum willingness to pay $\ell_{\max} = \$50$, and $\sigma = 3/5$. We chose

Figure 4.12: Manhattan divided into $n = 20$ regions.

a battery capacity of 24kWh, and discretized the battery energy into $e_{\max} = 6$ units, where one unit of battery energy is 4kWh. Price of electricity per unit of energy (4kWh) ranges from $\$0.32$ to $\$1.2$[193], and we randomly sampled p_i for all locations uniformly from this range.

For the network and demand data, we divided Manhattan into 20 regions as in Figure 4.12. Using the yellow taxi data from the New York City Taxi and Limousine Commission dataset [173] for May 09, 2019, Thursday between 15.00-17.00, we extracted the average arrival rates for rides, average trip durations, and average distances between the regions (we excluded the rides occurring in the same region). Note that the demand data used is not the data of potential riders, but the data of realized rides. Although it is not ideal to impose a demand function on the data of realized rides, this is the best data we could use due to lack of available data on potential riders. This is a common approach in the literature of pricing schemes in ride-sharing platforms [186, 194], as the realized rides leaving a location can be seen as a reasonable proxy for the potential riders at that location.

4.3.4.1 Effects of Competition Under Static Setting

In this study, we analyze the effects of competition using prices for rides, induced demand, profits, and consumer surplus as metrics. To get the values of the aforementioned metrics in

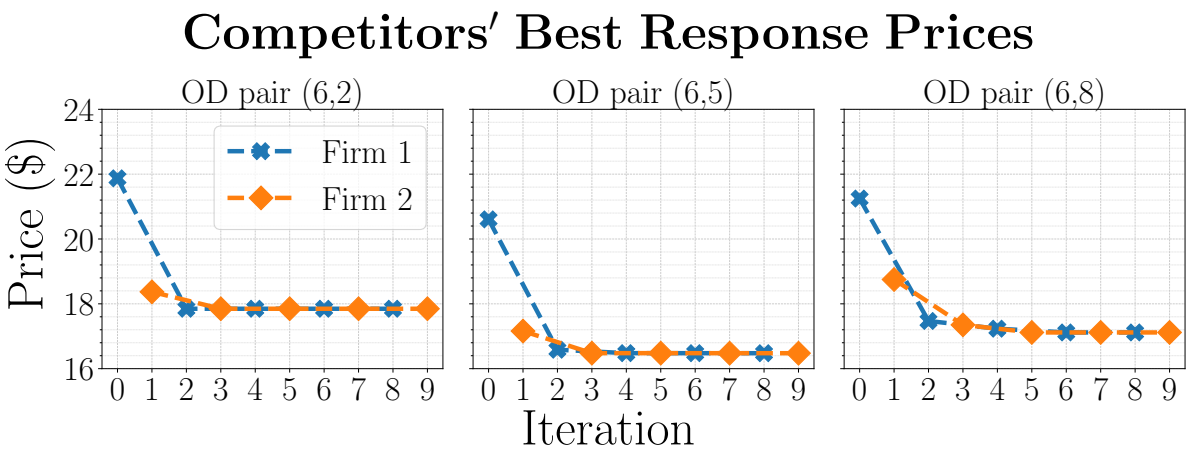


Figure 4.13: Best response prices for some rides originating from node 6.

the monopoly, we solved (4.16). For the duopoly, we can not solve (4.21) since the problem is non-convex. Therefore, we implemented best-response dynamics to see empirically whether this process would converge to an equilibrium of the duopoly so that we could numerically compare the monopoly and the duopoly. Although we do not have a theoretical guarantee for convergence of best response dynamics, we know that only symmetric equilibria exist according to Proposition 4.3.4. Fortunately, our experiment converged to a symmetric equilibrium in a couple of iterations as demonstrated in Figure 4.13.

In Table 4.6, we display the ratios of performance metrics in the monopoly and the symmetric duopoly equilibrium. Moreover, we compute the theoretic upper and lower bounds derived in Subsection 4.3.2 for $\sigma = 3/5$ for comparison. To summarize the table, competition results in a 20% decrease in the average prices of rides, a 44% increase in the total induced demand, a 43% decrease in the profits of a single firm, and a 100% increase in the consumer surplus.

Impact of σ: The correlation over customers' preferences is measured by σ, and the effects of competition depend on the value of σ. To study how σ influences the effects of competition, we present the ratios of performance metrics in the monopoly and the symmetric duopoly equilibrium for $\sigma = 0.8$ and $\sigma = 1$ in Table 4.7.

Metrics	Empirical	Theoretic LB	Theoretic UB
$\ell^d_{\text{avg}}/\ell^m_{\text{avg}}$	0.80	0.67	1
D^d/D^m	1.44	1.25	2.26
P^d/P^m	0.57	0.39	0.85
CS^d/CS^m	2.00	1.46	5.89

Table 4.6: Ratios of average prices, induced demand, profits, and consumer surplus in the monopoly and the symmetric duopoly equilibrium for $\sigma = 3/5$.

Metrics	Empirical		Theoretic LB		Theoretic UB	
	$\sigma = 0.8$	$\sigma = 1$	$\sigma = 0.8$	$\sigma = 1$	$\sigma = 0.8$	$\sigma = 1$
$\ell^d_{\text{avg}}/\ell^m_{\text{avg}}$	0.42	0.11	0.29	0	1	1
D^d/D^m	1.73	2.04	1.11	1	2.55	4
P^d/P^m	0.32	0	0.19	0	0.74	0
CS^d/CS^m	2.95	4.18	1.22	1	9.22	16

Table 4.7: Ratios of average prices, induced demand, profits, and consumer surplus in the monopoly and the symmetric duopoly equilibrium for $\sigma = 0.8$ and $\sigma = 1$.

The results in Tables 4.6 and 4.7 indicate that the higher the σ, the stronger the competition between the firms. A larger σ indicates higher correlation over customers' preferences, which means that the customers care less about the identity of the firm and more about lower prices when buying a ride ($\sigma = 1$ means they buy from the firm that offers the lower price). Hence, a stronger competition requires the firms to drop their prices further, which in turn decreases their profits more. This is in favor of the customers, since lower prices induce more demand while generating higher consumer surplus.

4.3.4.2 Real-Time Control

In this study, we demonstrate the performances of the model predictive controllers utilizing static and dynamic pricing schemes using profits (minus the queue penalty) and the average wait time of the customers as metrics. To quantify the queue penalty, we set queue penalty per person to be $w_{ijt} = \$4$ (by doubling the average hourly wage of $\$24$ in the U.S.[171]).

We computed the instantaneous profits in one period as:

$$\text{Profits} = \text{Revenue} - (\text{Operational + Charging Costs}), \tag{4.38}$$

the queue penalty in one period as:

$$\text{Queue Penalty} = w \times \text{Outstanding Customers}, \tag{4.39}$$

and used the objective value of (4.16) as an upper bound on the average profits for comparison. We define

$$\text{Normalized Queue Length} := \frac{\text{Outstanding Customers}}{\text{Induced Demand}} \tag{4.40}$$

and compute the instantaneous average wait time of customers in one period as:

$$\text{Avg. Wait Time} = \text{Normalized Queue Length} \times \Delta t. \tag{4.41}$$

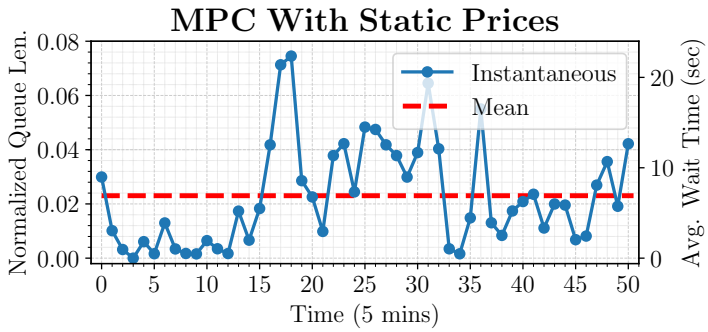
We implemented the MPC with $T = 10 \times \Delta t$ as the planning horizon, and ran the environment for $50 \times \Delta t$.

Monopoly

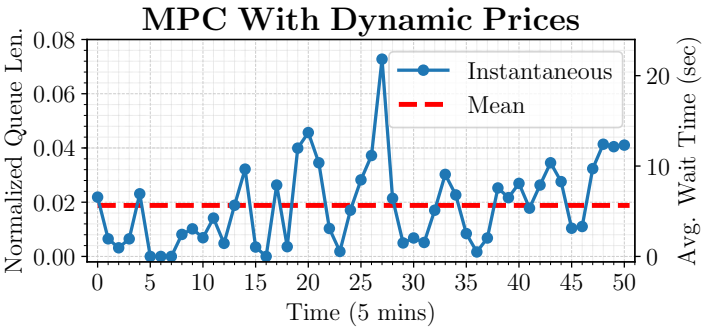
We plot the instantaneous average wait time for MPC with static prices (MPC-SP) and dynamic prices (MPC-DP) in Figure 4.14, and summarize the results in Table 4.8.

Metrics	MPC-SP	MPC-DP	% Impr.
Mean Profits-Queue Penalty ($)	11700.86	11778.13	0.66%
% of static	98.36%	99.02%	
Mean Avg. Wait Time (sec)	6.91	5.64	18.38%
Var. Avg. Wait Time (sec)	32.58	20.95	35.7%

Table 4.8: MPC results in the monopoly. Mean and variance are computed over time. The static objective value is 11894.9.



(a)



(b)

Figure 4.14: MPC results. We plot the normalized queue length for the MPC with static prices (a)/MPC with dynamic prices (b).

We observe that both controllers are able to keep the queue lengths very short (around 2% of the induced demand), and still generate substantial amount of profits that is close to the static objective. In particular, MPC-SP generates 98.36% and MPC-DP generates 99.02% of the static profits, including the queue penalty. Although the marginal benefits of using dynamic pricing might seem low, a 0.66% increase in average profits would make a considerable difference in the long run (e.g., from Table 4.8, a $\$77$ increase in profits per period adds up to more than an increase of $\$900$ per hour). Moreover, we observe that the mean of average wait time for MPC-SP is 6.91 seconds, while that of MPC-DP is 5.64 seconds which is an improvement of 18.38%. Lastly, a dynamic pricing scheme reduces the variance of the average wait time by 35.7%, which indicates a more robust system with predictable wait times.

We furthermore generated integer actions by randomizing according to the flows of the

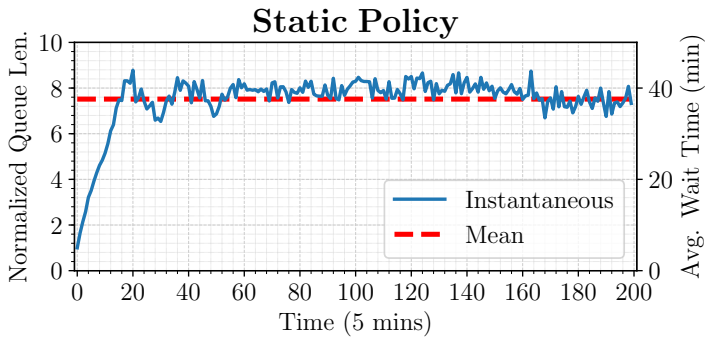


Figure 4.15: Monopoly Static Policy Queues

static solution and implemented the static policy in the real environment to compare its performance. In Figure 4.15 we plot the average wait time using the static policy. Although it provides stability of the queues, it results in bad wait times with a mean of 36.9 minutes, which is more than 300 times longer than both MPC-SP and MPC-DP.

Duopoly

We computed the mean value of the metrics over both firms to get the performances of the controllers. The results are summarized in Table 4.9.

Metrics	MPC-SP	MPC-DP	% Impr.
Mean Profits-Queue Penalty ($)	6670.89	6729.2	0.87%
% of static	98.56%	99.42%	
Mean Avg. Wait Time (sec)	7.27	5.01	31.08%
Var. Avg. Wait Time (sec)	44.68	17.92	59.89%

Table 4.9: MPC results in the duopoly. Mean and variance are computed over time. The static objective value is 6768.2.

Similar to the monopoly, both controllers are able to keep the queues short while generating profits close to the static objective, with dynamic pricing scheme increasing the efficiency.

4.3.5 Conclusion

In this section, we studied the impacts of competition on electric AMoD systems by comparing the monopoly and the duopoly in equilibrium. By formalizing the optimal strategies of profit-maximizing platform operators, we show that the identical competitors can only be in a symmetric equilibrium. In state of a symmetric duopoly equilibrium, the prices for rides and the profits of the firms are always less than those in the monopolistic setting, whereas the aggregate demand served and the consumer surplus are always higher. The closed-form universal bounds we provide quantify the amount of increase/reduction on the said metrics. These bounds depend heavily on the correlation between customers' preferences and therefore the strength of the competition. The numerical studies using network and demand data of Manhattan indicate that stronger competition boosts the amount of increase/reduction on the metrics. Lastly, we experimentally demonstrate that it is possible to implement a real-time control policy for fleet management using model predictive control, and show that a real-time pricing policy further improves the performance.

Chapter 5

Conclusions

5.1 Review

In this thesis, we have explored a multitude of challenges encompassed within the overarching domain of optimization in Human-Cyber-Physical Systems, with a keen focus on elevating robustness, safety, and efficiency.

Namely in Chapter 2, we showcased the potential shortcomings of vanilla distributed optimization algorithms in the presence of malicious agents and proposed novel robust distributed optimization algorithms based on robust estimation techniques, with applications on distributed resource allocation and distributed learning. In Chapter 3, we introduced a novel algorithm for solving resource allocation problems over networks through pricing only and without any two-way communications with agents as common in distributed optimization methods. The prices produced by our algorithm ensure that the induced demand satisfies the constraints of the system during the optimization process, which promotes safety. In Chapter 4 we developed a real-time control policy based on deep reinforcement learning for operating an AMoD fleet of EVs as well as pricing for rides. Additionally, we provided theoretical studies quantifying the impacts of competition on ride prices, profits of the platform operators, aggregate demand

served, and consumer surplus by comparing the monopoly and the duopoly.

5.2 Future Directions

The high-level methodology for the research presented in this thesis is to 1) identify the real-world challenge, 2) characterize a mathematical model that captures the real-world challenge accurately, and 3) develop a solution that addresses the challenge using performance metrics quantifiable by the mathematical model. Inevitably, the mathematical modeling of real-world problems through assumptions leads to some level of abstraction. Furthermore, while we were interested in certain performance metrics in this thesis (e.g., convergence rate, safety), other researchers might put importance on other performance metrics (e.g., complexity of the algorithm). Accordingly, here we discuss some future research directions that would take the work presented in this thesis to the next level by addressing the potential shortcomings.

5.2.1 Improving Memory & Time Complexity of Robust Distributed Optimization Algorithms

One of the novel ideas used in Chapter 2 to improve the robustness of the distributed optimization algorithms is to exploit the temporal dynamics of user behavior to detect anomalies through robust statistics. This necessitates the storage of a large number of high-dimensional vectors for the robust mean estimator, which can be seen as a trade-off between memory and robustness. It would be an interesting future direction to develop a robust mean estimator that can save on memory costs. Another novel idea for robustness in Chapter 2 is the normalization of the gradient and it is well known that normalized gradient-based optimization benefits significantly from using momentum [80], which reduces the time complexity. Given this, another potential future direction is to study whether momentum can be incorporated into the

algorithms presented in Chapter 2 while preserving their robustness properties. Although this thesis focused on whether robustness can be achieved without compromising efficiency too much, this trade-off could be addressed by improving the memory and time complexities of the presented algorithms.

5.2.2 Safe Pricing for Resource Allocation in Non-stationary Environments

Chapter 3 tackles the resource allocation problem in settings where users' resource demand can only be impacted through prices using a commonly adopted mathematical model of Network Utility Maximization problems. Although this model captures the essence of the real-world problem, it does not take into account the temporally changing behavior of the users and the system overall. In particular, the user utility for the resources as well as the safety-critical constraints of the system can change over time, which would mathematically be captured using time-dependent utility and constraint functions, e.g., [195]. Safe pricing algorithms for resource allocation in such non-stationary environments would reduce the level of abstraction and have a wider range of real-world applications.

5.2.3 Global Control Policies for Electric AMoD Systems

The studies in Chapter 4 demonstrated that it is possible to develop a real-time control and pricing policy for electric AMoD systems using reinforcement learning. Over the past few years, scholars have studied the fleet management problem for fleets of autonomous vehicles using various reinforcement learning approaches. One example is [196], which reveals that reinforcement learning agents can, with the aid of graph neural networks, attain behavior policies marked by significantly enhanced transferability, generalizability, and scalability. This success is owed to graph neural networks' ability to exploit the connectivity encoded by the transporta-

tion network. As such, the utilization of graph neural networks to develop global a joint pricing and fleet management policy for electric AMoD systems would be a promising future direction of research.

Appendix A

Supplements to Chapter 2

A.1 Proofs for Results in Section 2.2

A.1.1 Proof of Lemma 2.2.1

Observe that in both (2.9) and (2.12), the decision variables of the optimization problems are $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}$. Hence, it suffices to show that any given set of $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}$ satisfying constraints of (2.12) also satisfies constraints of (2.9). Let $\overline{\boldsymbol{\theta}}_{\mathcal{A}} := \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\boldsymbol{\theta}_i$. Since g_t is L-smooth, the following holds:

$$\begin{aligned}
&\max_{\boldsymbol{\theta}_j\in\mathcal{C}_j, j\in\mathcal{A}} g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^{N}\boldsymbol{\theta}_i\big) = \max_{\boldsymbol{\theta}_j\in\mathcal{C}_j, j\in\mathcal{A}} g_t\big(\tfrac{|\mathcal{H}|}{N}\overline{\boldsymbol{\theta}}_{\mathcal{H}} + \tfrac{|\mathcal{A}|}{N}\overline{\boldsymbol{\theta}}_{\mathcal{A}}\big)\\
&= \max_{\boldsymbol{\theta}_j\in\mathcal{C}_j, j\in\mathcal{A}} g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}} + (\alpha_1 - \tfrac{|\mathcal{A}|}{N})\overline{\boldsymbol{\theta}}_{\mathcal{H}} + \tfrac{|\mathcal{A}|}{N}\overline{\boldsymbol{\theta}}_{\mathcal{A}}\big) \qquad \text{(A.1)}\\
&\leq \max_{\boldsymbol{\theta}_j\in\mathcal{C}_j, j\in\mathcal{A}} \Big(g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\big) + \Big\langle \widetilde{\boldsymbol{\theta}}, \nabla g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\big)\Big\rangle + \tfrac{L}{2}\|\widetilde{\boldsymbol{\theta}}\|^2 \Big),
\end{aligned}$$

where we defined $\widetilde{\boldsymbol{\theta}} := (\alpha_1 - \frac{|\mathcal{A}|}{N})\overline{\boldsymbol{\theta}}_{\mathcal{H}} + \frac{|\mathcal{A}|}{N}\overline{\boldsymbol{\theta}}_{\mathcal{A}}$. Observe that

$$\|\widetilde{\boldsymbol{\theta}}\| \leq (\alpha_1 - \tfrac{|\mathcal{A}|}{N})\|\overline{\boldsymbol{\theta}}_{\mathcal{H}}\| + \tfrac{|\mathcal{A}|}{N}\|\overline{\boldsymbol{\theta}}_{\mathcal{A}}\| \leq \alpha_1 R. \qquad \text{(A.2)}$$

Furthermore, since the gradient of g_t is uniformly bounded by B and $\alpha_1^2 \leq \alpha_1$, (A.1) can be upper bounded by:

$$g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\big) + \alpha_1\big(RB + \tfrac{1}{2}LR^2\big) \tag{A.3}$$

Hence, we have shown that for any given set of $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}$, the following holds:

$$\max_{\boldsymbol{\theta}_j \in \mathcal{C}_j, j\in\mathcal{A}} g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^N \boldsymbol{\theta}_i\big) \leq g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\big) + \alpha_1\big(RB + \tfrac{1}{2}LR^2\big). \tag{A.4}$$

As such defining $c_t := \alpha_1\big(RB + \frac{1}{2}LR^2\big)$, it can be seen that if a set of $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}$ satisfies

$$g_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}\big) + c_t \leq 0,\ t = 1, ..., T, \tag{A.5}$$

the same set of $\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}}$ satisfies the desired constraint (2.9).

A.1.2 Proof of Proposition 2.2.2

Fix any $j \in [d]$. The assumption implies that for all $i \in \mathcal{H}$, one has:

$$|[\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j| \leq r. \tag{A.6}$$

We observe that $|\mathcal{H}| \geq (1-\alpha_1)N$. Applying [40, Lemma 1] shows that the median estimator[1] satisfies

$$|[\boldsymbol{x}_{\mathsf{med}} - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j| \leq (1-\alpha_1)\sqrt{\frac{1}{1-2\alpha_1}}r. \tag{A.7}$$

The above implies that for all $i \in \mathcal{H}$, we have

$$|[\boldsymbol{x}_i - \boldsymbol{x}_{\mathsf{med}}]_j| \leq \left(1 + \sqrt{\frac{(1-\alpha_1)^2}{1-2\alpha_1}}\right) r. \tag{A.8}$$

[1] At each coordinate, the median is the geometric median estimator of one dimension in [40].

This implies that $r_j \leq \left(1+\sqrt{\frac{(1-\alpha_1)^2}{1-2\alpha_1}}\right) r$, since $|\mathcal{H}| \geq (1-\alpha_1)N$. We then bound the performance of $\widehat{\boldsymbol{x}}_{\mathcal{H}}$:

$$(1-\alpha_1)N[\widehat{\boldsymbol{x}}_{\mathcal{H}}]_j = \sum_{i\in\mathcal{N}_j}[\boldsymbol{x}_i]_j = \sum_{i\in\mathcal{H}}[\boldsymbol{x}_i]_j - \sum_{i\in\mathcal{H}\setminus\mathcal{N}_j}[\boldsymbol{x}_i]_j + \sum_{i\in\mathcal{A}\cap\mathcal{N}_j}[\boldsymbol{x}_i]_j, \tag{A.9}$$

thus

$$\begin{aligned}(1-\alpha_1)N[\widehat{\boldsymbol{x}}_{\mathcal{H}} - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j &= -\sum_{i\in\mathcal{H}\setminus\mathcal{N}_j}[\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j + \sum_{i\in\mathcal{A}\cap\mathcal{N}_j}[\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j \\ &= -\sum_{i\in\mathcal{H}\setminus\mathcal{N}_j}[\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j + \sum_{i\in\mathcal{A}\cap\mathcal{N}_j}[\boldsymbol{x}_i - \boldsymbol{x}_{\mathsf{med}} + \boldsymbol{x}_{\mathsf{med}} - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j\end{aligned} \tag{A.10}$$

Notice that $|\mathcal{A}\cap\mathcal{N}_j| \leq \alpha_1 N$ and thus $|\mathcal{H}\setminus\mathcal{N}_j| \leq \alpha_1 N$. Gathering terms shows

$$|[\widehat{\boldsymbol{x}} - \overline{\boldsymbol{x}}_{\mathcal{H}}]_j| \leq \frac{2\alpha_1 N}{(1-\alpha_1)N}\left(1+\sqrt{\frac{(1-\alpha_1)^2}{1-2\alpha_1}}\right) r. \tag{A.11}$$

The above holds for all $j \in [d]$. Applying the norm equivalence shows the desired bound.

A.1.3 Proof of Lemma 2.2.2

Let $\left[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\right]_i$ denote the ith block of $\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}$, and $\left[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)}\right]_i$ denote the ith block of $\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)}$. From Equation (2.18a):

$$\left[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)}\right]_i = \frac{1}{N}\big(\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)} + \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}\big). \tag{A.12}$$

Furthermore, we replace $\widehat{\boldsymbol{g}}_{\mathcal{H}}^{(k)}$ from Algorithm 2 Step 3(c):

$$\left[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)}\right]_i = \frac{1}{N}\Big(\sum_{t=1}^{T}\lambda_t^{(k)}\nabla_{\boldsymbol{\theta}}\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}) + \nabla_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{(k)}) + \upsilon\boldsymbol{\theta}_i^{(k)}\Big). \tag{A.13}$$

The perturbation $\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_i = \big[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)}\big]_i - \nabla_{\boldsymbol{\theta}_i}\overline{\mathcal{L}}_v(\{\boldsymbol{\theta}_i\}_{i\in\mathcal{H}};\boldsymbol{\lambda};\mathcal{H})$ is given by the difference between (A.13) and (2.14a):

$$\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_i = \frac{1}{N}\sum_{t=1}^{T}\lambda_t^{(k)}\Big(\nabla_{\boldsymbol{\theta}}\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}) - \frac{(1-\alpha_1)N}{|\mathcal{H}|}\nabla_{\boldsymbol{\theta}}\overline{g}_t((1-\alpha_1)\bar{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)})\Big). \tag{A.14}$$

By adding and subtracting $\frac{1}{N}\Big(\sum_{t=1}^{T}\lambda_t^{(k)}\nabla_{\boldsymbol{\theta}}\overline{g}_t((1-\alpha_1)\bar{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\Big)$, the above expression becomes:

$$\begin{aligned}\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_i = \frac{1}{N}\sum_{t=1}^{T}\lambda_t^{(k)}\Big(&\nabla_{\boldsymbol{\theta}}\overline{g}_t\big((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\big) - \nabla_{\boldsymbol{\theta}}\overline{g}_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\big)\\ &+ \frac{|\mathcal{H}|-(1-\alpha_1)N}{|\mathcal{H}|}\nabla_{\boldsymbol{\theta}}\overline{g}_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\big)\Big)\end{aligned} \tag{A.15}$$

Similarly, comparing (2.19b) with (2.18b) and (2.14b), we identify that:

$$\big[\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\big]_t = \overline{g}_t\big((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\big) - \overline{g}_t\big((1-\alpha_1)\overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\big). \tag{A.16}$$

Using Assumption 2.2.1 and the said assumptions, we immediately see that

$$\|\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_i\| \leq (1-\alpha_1)\frac{\overline{\lambda}LT}{N}\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\| + \frac{|\mathcal{H}|-(1-\alpha_1)N}{|\mathcal{H}|}\frac{\overline{\lambda}BT}{N} \tag{A.17}$$

which then implies (2.20). Assumption 2.2.1 implies that $\overline{g}_t$ is B-Lipschitz continuous, therefore

$$|\big[\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\big]_t| \leq B(1-\alpha_1)\|\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)} - \overline{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}\|, \tag{A.18}$$

which implies (2.21).

A.1.4 Proof of Theorem 2.2.1

Based on Lemma 2.2.2, our idea is to perform a perturbation analysis on the PDA algorithm. Without loss of generality, we assume $N = 1$ and denote $\boldsymbol{\theta} = \boldsymbol{\theta}_1$. To simplify notations

we define $\upsilon' := (1-\alpha_1)\upsilon$. We also drop the subscript, denote the modified and regularized Lagrangian function as $\mathcal{L} = \overline{\mathcal{L}}_\upsilon$. Furthermore, we denote the saddle point to (P'_υ) as $\boldsymbol{z}^\star = (\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)$.

Using the fact that $\boldsymbol{\theta}^\star = \mathcal{P}_{\mathcal{C}}(\boldsymbol{\theta}^\star) = \mathcal{P}_{\mathcal{C}}\big(\boldsymbol{\theta}^\star - \gamma\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\big)$, we observe that in the primal update:

$$\begin{aligned}\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^\star\|^2 \overset{(a)}{\leq} &\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\|^2 - 2\gamma\langle\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star), \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\rangle \\ &+ \gamma^2\|\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(k)} - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\|^2 \end{aligned} \tag{A.19}$$

where (a) is due to the projection inequality $\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{x}-\boldsymbol{y})\| \leq \|\boldsymbol{x}-\boldsymbol{y}\|$. Furthermore, using the Young's inequality, for any $c_0, c_1 > 0$, we have

$$\begin{aligned}&\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^\star\|^2 \leq \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\|^2 - 2\gamma\langle\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star), \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\rangle \\ &\quad+\gamma^2(1+c_0)\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\|^2 - 2\gamma\langle\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}, \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\rangle + \gamma^2\big(1+\frac{1}{c_0}\big)\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 \\ &\leq (1+2c_1\gamma)\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\|^2 - 2\gamma\langle\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star), \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^\star\rangle \\ &\quad+ \gamma^2(1+c_0)\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\|^2 + \Big(\frac{2\gamma}{c_1} + \gamma^2 + \frac{\gamma^2}{c_0}\Big)\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2. \end{aligned} \tag{A.20}$$

Similarly, in the dual update we get,

$$\begin{aligned}&\|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^\star\|^2 \leq \|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\|^2 + \gamma^2\|\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} - \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\|^2 \\ &\quad+ 2\gamma\langle\widehat{\boldsymbol{g}}_{\boldsymbol{\lambda}}^{(k)} - \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star), \boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\rangle \\ &\leq (1+2c_1\gamma)\|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\|^2 + 2\gamma\langle\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star), \boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^\star\rangle \\ &\quad+ \gamma^2(1+c_0)\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\lambda}^{(k)}) - \nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star)\|^2 + \Big(\frac{2\gamma}{c_1} + \gamma^2 + \frac{\gamma^2}{c_0}\Big)\|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2. \end{aligned} \tag{A.21}$$

Summing up the two inequalities gives:

$$\begin{aligned}\|\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^\star\|^2 &\leq (1+2c_1\gamma)\|\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\|^2+\Big(\frac{2\gamma}{c_1}+\gamma^2+\frac{\gamma^2}{c_0}\Big)E_k\\ &\quad-2\gamma\langle\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})-\boldsymbol{\Phi}(\boldsymbol{z}^\star),\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\rangle+\gamma^2(1+c_0)\|\boldsymbol{\Phi}(\boldsymbol{z}^{(k)})-\boldsymbol{\Phi}(\boldsymbol{z}^\star)\|^2\\ &\overset{(a)}{\leq}\Big(1+2\gamma(c_1-\upsilon')+\gamma^2(1+c_0)L_\Phi^2\Big)\|\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\|^2+\Big(\frac{2\gamma}{c_1}+\gamma^2+\frac{\gamma^2}{c_0}\Big)E_k,\end{aligned}\tag{A.22}$$

where (a) uses the strong monotonicity and smoothness of the map $\boldsymbol{\Phi}$, cf. [24, Lemma 3.4]. Setting $c_1=\upsilon'/2$ yields

$$\|\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^\star\|^2\leq\Big(1-\gamma\upsilon'+\gamma^2(1+c_0)L_\Phi^2\Big)\|\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\|^2+\Big(\frac{4\gamma}{\upsilon'}+\gamma^2+\frac{\gamma^2}{c_0}\Big)E_k.\tag{A.23}$$

Observe that we can choose γ such that $1-\gamma\upsilon'+\gamma^2(1+c_0)L_\Phi^2<1$. Moreover, the above inequality implies that $\|\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\|^2$ evaluates to

$$\begin{aligned}\|\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^\star\|^2&\leq(1-\gamma\upsilon'+\gamma^2(1+c_0)L_\Phi^2)^k\|\boldsymbol{z}^{(0)}-\boldsymbol{z}^\star\|^2\\ &\quad+\sum_{\ell=1}^{k}(1-\gamma\upsilon'+\gamma^2(1+c_0)L_\Phi^2)^{k-\ell}\Big(\frac{4\gamma}{\upsilon'}+\gamma^2+\frac{\gamma^2}{c_0}\Big)E_\ell.\end{aligned}\tag{A.24}$$

If $E_k\leq\overline{E}$ for all k, then $\boldsymbol{z}^{(k)}$ converges to a neighborhood of $\boldsymbol{z}^\star$ of radius

$$\limsup_{k\to\infty}\|\boldsymbol{z}^{(k)}-\boldsymbol{z}^\star\|^2\leq\frac{\frac{4\gamma}{\upsilon'}+\gamma^2+\frac{\gamma^2}{c_0}}{\gamma\upsilon'-\gamma^2(1+c_0)L_\Phi^2}\overline{E}.\tag{A.25}$$

Setting $c_0=1$ concludes the proof.

A.1.5 Proof of Lemma 2.2.3

Comparing the equations in (2.27) with (2.26a) and (2.26b), we identify that:

$$\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_j = \frac{1}{N}\sum_{t=1}^{T}\lambda_t^{(k)}\big(\nabla_{\boldsymbol{\theta}} g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^{N}\widehat{\boldsymbol{\theta}}_i^{(k)}\big) - \nabla_{\boldsymbol{\theta}} g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^{N}\boldsymbol{\theta}_i^{(k)}\big)\big), \tag{A.26}$$

$$\big[\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\big]_t = g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^{N}\widehat{\boldsymbol{\theta}}_i^{(k)}\big) - g_t\big(\tfrac{1}{N}\textstyle\sum_{i=1}^{N}\boldsymbol{\theta}_i^{(k)}\big), \tag{A.27}$$

where $\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_j$ denotes the jth block of $\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}$. Using Assumption 2.2.1, we immediately see that:

$$\begin{aligned}\|\big[\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\big]_j\| &\leq \frac{\overline{\lambda}LT}{N}\|\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)})\| \\ &\leq \frac{\overline{\lambda}LT}{N^2}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|,\end{aligned} \tag{A.28}$$

which then implies (2.28a). Assumption 2.2.1 implies that g_t is B-Lipschitz continuous, therefore

$$\begin{aligned}|\big[\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\big]_t| &\leq B\|\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)})\| \\ &\leq \frac{B}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|,\end{aligned} \tag{A.29}$$

which implies (2.28b).

A.1.6 Proof of Lemma 2.2.4

Observe that the gradient perturbation in both dual and primal variables is upper bounded by some constant times $\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|$ in (2.28). Thus, we would like to upper bound this term. Let $\mathcal{H}_i^{(k)}$ be the set of $(1-\alpha_2)m$ trustworthy parameters of agent i out of the last m parameters at iteration k, i.e., $(1-\alpha_2)m$ trustworthy parameters from set $\{\boldsymbol{r}_i^{(k-\ell)}\}_{\ell=0}^{m-1}$. Note that if a parameter is trustworthy, then $\boldsymbol{r}_i^{(k-\ell)} = \boldsymbol{\theta}_i^{(k-\ell)}$. Hence we define the mean of the

iterates in set $\mathcal{H}_i^{(k)}$ as:

$$\overline{\boldsymbol{\theta}}_i^{(k)} := \frac{1}{(1-\alpha_2)m} \sum_{\boldsymbol{\theta}_i^{(k-\ell)} \in \mathcal{H}_i^{(k)}} \boldsymbol{\theta}_i^{(k-\ell)}. \tag{A.30}$$

Using triangular inequality, we can write:

$$\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| = \|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)} + \overline{\boldsymbol{\theta}}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \leq \|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| + \|\overline{\boldsymbol{\theta}}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|. \tag{A.31}$$

Let $\widehat{\boldsymbol{\theta}}_i^{(k)}$ be the estimated mean using median-based estimator. Using norm equivalence:

$$\begin{aligned} \max_{\boldsymbol{\theta}_i^{(k-\ell)} \in \mathcal{H}_i^{(k)}} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\|_\infty &\leq \max_{\boldsymbol{\theta}_i^{(k-\ell)} \in \mathcal{H}_i^{(k)}} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| \\ &\leq \max_{0 \leq \ell \leq m-1} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\|. \end{aligned} \tag{A.32}$$

Thus, under Assumption 2.2.2, Proposition 2.2.2 suggests:

$$\|\overline{\boldsymbol{\theta}}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \leq C_\alpha \max_{0 \leq \ell \leq m-1} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\|, \tag{A.33}$$

where $C_\alpha = \frac{2\alpha_2}{1-\alpha_2} \left(1 + \sqrt{\frac{(1-\alpha_2)^2}{1-2\alpha_2}}\right) \sqrt{d}$.

Let $\ell^\star = \underset{0 \leq \ell \leq m-1}{\arg\max} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\|$. Then:

$$\begin{aligned} &\max_{0 \leq \ell \leq m-1} \|\boldsymbol{\theta}_i^{(k-\ell)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| = \|\boldsymbol{\theta}_i^{(k-\ell^\star)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| \\ &= \|\boldsymbol{\theta}_i^{(k-\ell^\star)} - \boldsymbol{\theta}_i^{(k-\ell^\star+1)} + \boldsymbol{\theta}_i^{(k-\ell^\star+1)} - \ldots - \boldsymbol{\theta}_i^{(k-1)} + \boldsymbol{\theta}_i^{(k+1)} - \boldsymbol{\theta}_i^{(k)} + \boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| \\ &\leq \|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| + \sum_{j=k-\ell^\star}^{k-1} \|\boldsymbol{\theta}_i^{(j)} - \boldsymbol{\theta}_i^{(j+1)}\| \\ &\leq \|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| + \sum_{j=k-\ell^\star}^{k-1} \|\gamma [\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i\| \\ &\leq \|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| + \gamma \sum_{j=k-m+1}^{k-1} \|[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i\|, \end{aligned} \tag{A.34}$$

where $[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i$ denotes the ith block of $\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}$. Using equations (A.33) and (A.34), we can rewrite (A.31):

$$\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \leq (1 + C_\alpha)\|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| + \gamma C_\alpha \sum_{j=k-m+1}^{k-1} \|[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i\|. \tag{A.35}$$

Next step is to bound the $\|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\|$ term:

$$\begin{aligned}
\|\boldsymbol{\theta}_i^{(k)} - \overline{\boldsymbol{\theta}}_i^{(k)}\| &= \|\boldsymbol{\theta}_i^{(k)} - \frac{1}{(1-\alpha_2)m} \sum_{\boldsymbol{\theta}_i^{(k-\ell)} \in \mathcal{H}_i^{(k)}} \boldsymbol{\theta}_i^{(k-\ell)}\| \\
&\leq \frac{1}{(1-\alpha_2)m} \sum_{\boldsymbol{\theta}_i^{(k-\ell)} \in \mathcal{H}_i^{(k)}} \|\boldsymbol{\theta}_i^{(k)} - \boldsymbol{\theta}_i^{(k-\ell)}\| \\
&\leq \frac{1}{(1-\alpha_2)m} \sum_{\ell=0}^{m-1} \|\boldsymbol{\theta}_i^{(k)} - \boldsymbol{\theta}_i^{(k-\ell)}\| \\
&= \frac{1}{(1-\alpha_2)m} \sum_{\ell=0}^{m-1} \|\boldsymbol{\theta}_i^{(k)} - \boldsymbol{\theta}_i^{(k-1)} + \boldsymbol{\theta}_i^{(k-1)} - \ldots - \boldsymbol{\theta}_i^{(k-\ell+1)} + \boldsymbol{\theta}_i^{(k-\ell+1)} - \boldsymbol{\theta}_i^{(k-\ell)}\| \\
&\leq \frac{1}{(1-\alpha_2)m} \sum_{\ell=0}^{m-1} \sum_{j=k-\ell}^{k-1} \|\boldsymbol{\theta}_i^{(j+1)} - \boldsymbol{\theta}_i^{(j)}\| \\
&\leq \frac{m}{(1-\alpha_2)m} \sum_{j=k-m+1}^{k-1} \|\boldsymbol{\theta}_i^{(j+1)} - \boldsymbol{\theta}_i^{(j)}\| \\
&\leq \frac{1}{1-\alpha_2} \gamma \sum_{j=k-m+1}^{k-1} \|[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i\|.
\end{aligned} \tag{A.36}$$

Plugging (A.36) into (A.35):

$$\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \leq \left(\frac{1 + C_\alpha}{1 - \alpha_2} + C_\alpha\right) \gamma \sum_{j=k-m+1}^{k-1} \|[\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}]_i\|. \tag{A.37}$$

For brevity of notation, let $\left(\frac{1+C_\alpha}{1-\alpha_2} + C_\alpha\right) = \overline{C}_\alpha$ and let $\nabla_{\boldsymbol{\theta}} \mathcal{L}_\upsilon^{(k)} := \nabla_{\boldsymbol{\theta}} \mathcal{L}_\upsilon(\boldsymbol{\theta}^{(k)}; \boldsymbol{\lambda}^{(k)})$. Summing up (A.37) for all agents and using norm equivalence:

$$\begin{aligned}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(k)}-\widehat{\boldsymbol{\theta}}_i^{(k)}\| &\leq \overline{C}_\alpha\sqrt{N}\gamma\sum_{j=k-m+1}^{k-1}\|\widehat{\boldsymbol{g}}_{\boldsymbol{\theta}}^{(j)}\| \\ &\overset{(2.27a)}{=} \overline{C}_\alpha\sqrt{N}\gamma\sum_{j=k-m+1}^{k-1}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_\upsilon^{(j)}+\boldsymbol{e}_{\boldsymbol{\theta}}^{(j)}\| \\ &\leq \overline{C}_\alpha\sqrt{N}\gamma\sum_{j=k-m+1}^{k-1}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_\upsilon^{(j)}\|+\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(j)}\|.\end{aligned} \tag{A.38}$$

Using (2.28a):

$$\begin{aligned}\|\boldsymbol{e}_{\boldsymbol{\theta}}^{(j)}\| &\leq \frac{\overline{\lambda}LT}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(j)}-\widehat{\boldsymbol{\theta}}_i^{(j)}\| \\ &= \frac{\overline{\lambda}LT}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(j)}-\boldsymbol{\theta}_i^{\star}+\boldsymbol{\theta}_i^{\star}-\widehat{\boldsymbol{\theta}}_i^{(j)}\| \\ &\leq \frac{\overline{\lambda}LT}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{(j)}-\boldsymbol{\theta}_i^{\star}\|+\|\widehat{\boldsymbol{\theta}}_i^{(j)}-\boldsymbol{\theta}_i^{\star}\| \\ &\overset{(*)}{\leq} (1+\sqrt{d})\frac{\overline{\lambda}LT}{N}\sum_{i=1}^{N}\max_{0\leq\ell_i\leq m-1}\|\boldsymbol{\theta}_i^{(j-\ell_i)}-\boldsymbol{\theta}_i^{\star}\| \\ &\leq (1+\sqrt{d})\overline{\lambda}LT\max_{\substack{i\\0\leq\ell_i\leq m-1}}\|\boldsymbol{\theta}_i^{(j-\ell_i)}-\boldsymbol{\theta}_i^{\star}\| \\ &\leq (1+\sqrt{d})\overline{\lambda}LT\max_{0\leq\ell\leq m-1}\|\boldsymbol{z}^{(j-\ell)}-\boldsymbol{z}^{\star}\|,\end{aligned} \tag{A.39}$$

where $(*)$ is obtained by:

$$\begin{aligned}\|\widehat{\boldsymbol{\theta}}_i^{(j)}-\boldsymbol{\theta}_i^{\star}\| &\leq \sqrt{d}\|\widehat{\boldsymbol{\theta}}_i^{(j)}-\boldsymbol{\theta}_i^{\star}\|_\infty \\ &\leq \sqrt{d}\max_{0\leq\ell\leq m-1}\|\boldsymbol{\theta}_i^{(j-\ell)}-\boldsymbol{\theta}_i^{\star}\|_\infty \\ &\leq \sqrt{d}\max_{0\leq\ell\leq m-1}\|\boldsymbol{\theta}_i^{(j-\ell)}-\boldsymbol{\theta}_i^{\star}\|,\end{aligned} \tag{A.40}$$

and $\|\boldsymbol{\theta}_i^{(j)}-\boldsymbol{\theta}_i^{\star}\| \leq \max_{0\leq\ell\leq m-1}\|\boldsymbol{\theta}_i^{(j-\ell)}-\boldsymbol{\theta}_i^{\star}\|$.

Furthermore, by using the L_Φ-Lipschitz property of $\overline{\mathbf{\Phi}}(\boldsymbol{z})$:

$$\begin{aligned}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_\upsilon^{(j)}\| \leq \|\overline{\mathbf{\Phi}}(\boldsymbol{z}^{(j)})\| &= \|\overline{\mathbf{\Phi}}(\boldsymbol{z}^{(j)}) - \overline{\mathbf{\Phi}}(\boldsymbol{z}^\star)\| \\ &\leq \frac{1}{L_\Phi}\|\boldsymbol{z}^{(j)} - \boldsymbol{z}^\star\| \\ &\leq \frac{1}{L_\Phi}\max_{0\leq\ell\leq m-1}\|\boldsymbol{z}^{(j-\ell)} - \boldsymbol{z}^\star\|\end{aligned} \tag{A.41}$$

We can rewrite (A.38) using equations (A.39) and (A.41):

$$\begin{aligned}&\sum_{i=1}^N \|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\| \leq \overline{C}_\alpha\sqrt{N}\gamma\sum_{j=k-m+1}^{k-1}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_\upsilon^{(j)}\| + \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(j)}\| \\ &\leq \overline{C}_\alpha\sqrt{N}C_0\gamma\sum_{j=k-m+1}^{k-1}\max_{0\leq\ell\leq m-1}\|\boldsymbol{z}^{(j-\ell)} - \boldsymbol{z}^\star\| \\ &\leq \overline{C}_\alpha\sqrt{N}C_0(m-1)\gamma\max_{1\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|,\end{aligned} \tag{A.42}$$

where $C_0 = \frac{1}{L_\Phi} + (1+\sqrt{d})\overline{\lambda}LT$. Finally, using (2.28) and letting $C_1 = \left(\frac{\overline{\lambda}LT}{N}\right)^2 + \left(\frac{BT}{N}\right)^2$:

$$\begin{aligned}E_k = \|\boldsymbol{e}_{\boldsymbol{\theta}}^{(k)}\|^2 + \|\boldsymbol{e}_{\boldsymbol{\lambda}}^{(k)}\|^2 &\leq C_1\left(\sum_{i=1}^N\|\boldsymbol{\theta}_i^{(k)} - \widehat{\boldsymbol{\theta}}_i^{(k)}\|\right)^2 \\ &\leq C_1(\overline{C}_\alpha\sqrt{N}C_0(m-1))^2\gamma^2\max_{1\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2 \\ &\leq C_1(\overline{C}_\alpha\sqrt{N}C_0(m-1))^2\gamma^2\max_{0\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2 \\ &\leq \overline{C}\gamma^2\max_{0\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2,\end{aligned} \tag{A.43}$$

where $\overline{C} = \left(\frac{T^2(\overline{\lambda}^2L^2+B^2)}{N}\right) \times \left(\frac{1}{L_\Phi} + (1+\sqrt{d})\overline{\lambda}LT\right)^2 \times \left(\frac{1+C_\alpha}{1-\alpha_2} + C_\alpha\right)^2 \times (m-1)^2$.

A.1.7 Proof of Theorem 2.2.2

Based on Lemma 2.2.3, our idea is to perform a perturbation analysis on the PDA algorithm. The first part of the proof is analogous to that of Theorem 2.2.1, and then upper bounding E_k by Lemma 2.2.4. This yields:

$$\begin{aligned}\|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^\star\| \leq & (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2)\|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 \\ & + \left(\frac{4\gamma}{\upsilon} + 2\gamma^2\right)\gamma^2\overline{C}\max_{0\leq\ell\leq 2(m-1)}\|\boldsymbol{z}^{(k-\ell)} - \boldsymbol{z}^\star\|^2.\end{aligned} \tag{A.44}$$

For the second part of the proof, we use the following Lemma:

Lemma A.1.1 *[197, Lemma 3] Let $\{V(t)\}$ be a sequence of real numbers satisfying*

$$V(t+1) \leq pV(t) + q\max_{t-\tau(t)\leq s\leq t} V(s) + r, \quad t \in \mathbb{N}_0,$$

for some nonnegative constants p,q, and r. If $p + q < 1$ and

$$0 \leq \tau(t) \leq \tau_{\max}, \quad t \in \mathbb{N}_0,$$

then

$$V(t) \leq \rho^t V(0) + \epsilon, \quad t \in \mathbb{N}_0,$$

where $\rho = (p+q)^{\frac{1}{1+\tau_{\max}}}$ and $\epsilon = r/(1-p-q)$.

We apply Lemma A.1.1 on (A.44) for $t = k \geq 2(m-1)$, $V(t) = V(k) = \|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2$, $p = 1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2$, $q = \left(\frac{4\gamma}{\upsilon} + 2\gamma^2\right)\gamma^2\overline{C}$, $r = 0$, and $\tau_{\max} = 2(m-1)$ to get:

$$V(k) \leq \rho^{k-2(m-1)}V(2(m-1)), \quad k \geq 2(m-1), \tag{A.45}$$

where $\rho = (1 - \gamma\upsilon + 2\gamma^2 L_\Phi^2 + \frac{4\overline{C}\gamma^3}{\upsilon} + 2\overline{C}\gamma^4)^{\frac{1}{1+2(m-1)}}$. The condition $p + q < 1$ is met when:

$$f(\gamma) = \upsilon - 2\gamma L_\Phi^2 - \frac{4\overline{C}\gamma^2}{\upsilon} - 2\overline{C}\gamma^3 > 0$$

Observe that $f(\gamma)$ is a continuous function in γ, and $f(0) = \upsilon > 0$. Hence, there exists a small $\gamma > 0$ such that $f(\gamma) > 0$, which satisfies the required condition. Taking the limit as k goes to infinity in (A.45):

$$\lim_{k\to\infty} V(k) \leq \lim_{k\to\infty} \rho^{k-2(m-1)} V(2(m-1)) = 0, \tag{A.46}$$

since $\rho < 1$. Finally, since $V(k) \geq 0$, we conclude that $\lim_{k\to\infty} V(k) = \lim_{k\to\infty} \|\boldsymbol{z}^{(k)} - \boldsymbol{z}^\star\|^2 = 0$.

A.1.8 Proof of Bounded Dual Variables

The proof is the same for statements in both Lemma 2.2.2 and Lemma 2.2.3. The update rule given by (2.18b):

$$\lambda_t^{(k+1)} = \big[\lambda_t^{(k)} + \gamma\big(\overline{g}_t((1-\alpha_1)\widehat{\boldsymbol{\theta}}_{\mathcal{H}}^{(k)}) - \upsilon\lambda_t^{(k)}\big)\big]_+. \tag{A.47}$$

The update rule given by (2.26b):

$$\lambda_t^{(k+1)} = \big[\lambda_t^{(k)} + \gamma\big(g_t(\widehat{\boldsymbol{\theta}}^{(k)}) - \upsilon\lambda_t^{(k)}\big)\big]_+. \tag{A.48}$$

Let $\overline{g}_t(\cdot) \leq M$ and $0 \leq \lambda_t^{(k)} \leq \frac{M}{\upsilon}$. We can upper bound both (A.47) and (A.48) as:

$$\lambda_t^{(k+1)} \leq (1 - \gamma\upsilon)\lambda_t^{(k)} + \gamma M \leq \frac{M}{\upsilon}. \tag{A.49}$$

Hence, one can set $\lambda_t^{(0)} \leq \frac{M}{\upsilon}$ to guarantee assumption. Anyhow if $\frac{M}{\upsilon} \leq \lambda_t^{(k)}$:

$$\lambda_t^{(k+1)} \leq (1-\gamma\upsilon)\lambda_t^{(k)} + \gamma M \leq \lambda_t^{(k)}. \tag{A.50}$$

Thus, if $\lambda_t^{(0)} \leq \frac{M}{\upsilon}$ then $\overline{\lambda} = \frac{M}{\upsilon}$. If $\lambda_t^{(0)} \geq \frac{M}{\upsilon}$ then $\overline{\lambda} = \lambda_t^{(0)}$. This guarantees the assumption.

A.2 Proofs for Results in Section 2.3

A.2.1 Proof of Theorem 2.3.1

The proof will use the following theorem in [92] as an auxiliary result, which offers a convenient bound for the expected value of the maximum of finitely many exponentially integrable random variables.

Theorem A.2.1 *[92, Theorem 2.5] Let $Z_1, \cdots, Z_N$ be real-valued random variables such that for every $\lambda \in (0,a)$ and $i = 1, \ldots, N$, the logarithm of the moment generating function of Z_i satisfies $\mathbb{E}[e^{\lambda Z_i}] \leq \phi(\lambda)$ where ϕ is a convex and continuously differentiable function on $[0,a)$ with $0 < a \leq \infty$ such that $\phi(0) = \phi'(0) = 0$. Then*

$$\mathbb{E}[\max_{i=1,\ldots,N} Z_i] \leq \inf_{\lambda\in(0,a)} \left[\frac{\log N + \phi(\lambda)}{\lambda}\right]. \tag{A.51}$$

Using the update rule, we write

$$\|x_{t+1} - x^\star\|^2 = \|\Pi_{\mathcal{X}}\{x_t - \gamma\hat{\hat{g}}_t/\|\hat{\hat{g}}_t\|\} - x^\star\|^2 \leq \|x_t - x^\star - \gamma\hat{\hat{g}}_t/\|\hat{\hat{g}}_t\|\|^2 \tag{A.52}$$

$$= \|x_t - x^\star\|^2 - 2\gamma\langle\hat{\hat{g}}_t/\|\hat{\hat{g}}_t\|, x_t - x^\star\rangle + \gamma^2 \tag{A.53}$$

$$= \|x_t - x^\star\|^2 - 2\gamma(1-Z_t)\langle\frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|}, x_t - x^\star\rangle - 2\gamma Z_t\langle\frac{\hat{\hat{g}}_t^{Z_t=1}}{\|\hat{\hat{g}}_t^{Z_t=1}\|}, x_t - x^\star\rangle + \gamma^2 \tag{A.54}$$

$$\leq \|x_t - x^\star\|^2 - 2\gamma(1-Z_t)\langle \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|}, x_t - x^\star\rangle \\ - 2\gamma(1-Z_t)\langle \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} - \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|}, x_t - x^\star\rangle + 2\gamma Z_t\|x_t - x^\star\| + \gamma^2 \tag{A.55}$$

$$\leq \|x_t - x^\star\|^2 - 2\gamma(1-Z_t)\frac{\mu}{L}\|x_t - x^\star\| \\ - 2\gamma(1-Z_t)\langle \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} - \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|}, x_t - x^\star\rangle + 2\gamma Z_t\|x_t - x^\star\| + \gamma^2 \tag{A.56}$$

$$\leq \|x_t - x^\star\|^2 - \frac{2\gamma}{\kappa}(1 - Z_t(1+\kappa))\|x_t - x^\star\| + 2\gamma\|e_t\|\|x_t - x^\star\| + \gamma^2, \tag{A.57}$$

where

$$e_t = \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} - \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} \tag{A.58}$$

In case of the event $Z_t = 0$, we know that there are at least $(1-\alpha_2)N$ agents for which $Y_{i,t} = 0$. Therefore, we define the following sets:

• Define $\mathcal{T}^t$ as the set of $(1-\alpha_2)N$ agents with smallest indices $i \in [N]$ for which $Y_{i,t} = 0$, i.e.,

$$\mathcal{T}^t = \{i | i \in [N], Y_{i,t} = 0, \sum_i 1 = (1-\alpha_2)N\} \tag{A.59}$$

such that $\sum_{i\in\mathcal{T}^t} i$ is minimized.

• For all agents $i \in \mathcal{T}^t$, define $\mathcal{T}_i^t$ as the set of $(1-\alpha_1)m$ smallest time indices $\tau \in [t-m+1, t]$ for which $W_{i,\tau} = 0$, i.e.,

$$\mathcal{T}_i^t = \{\tau | \tau \in [t-m+1, t], W_{i,\tau} = 0, \sum_\tau 1 = (1-\alpha_1)m\} \tag{A.60}$$

such that $\sum_{\tau\in\mathcal{T}_i^t}\tau$ is minimized.

Using the above sets, the following Lemma, whose proof can be found in Appendix A.2.2, bounds the norm of e_t:

Lemma A.2.1 *Suppose that $F(\cdot)$ and $F_{i,\tau}(\cdot)$, $\forall i, \tau$, are L-smooth. Define $\hat{\hat{g}}_t^{Z_t=0}$ as the robus-*

tified gradient at iteration t when $Z_t = 0$. Define the error e_t as

$$e_t = \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} - \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|}$$

Then for all $t \geq m$, the following bounds the norm of the error:

$$\begin{aligned}
\|e_t\| \leq & \frac{2}{\|\nabla F(x_t)\|} \Bigg(L\gamma(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1)) \\
& + \left\| \nabla F(x_t) - \frac{1}{(1-\alpha_1)m(1-\alpha_2)N} \sum_{i \in \mathcal{T}^t} \sum_{\tau \in \mathcal{T}_i^t} \nabla F_{i,\tau}(x_t) \right\| \\
& + 2C_{\alpha_2} \max_{i \in \mathcal{T}^t} \left\| \frac{1}{(1-\alpha_1)m} \sum_{\tau \in \mathcal{T}_i^t} \nabla F_{i,\tau}(x_t) - \nabla F(x_t) \right\|_\infty \\
& + \frac{1}{(1-\alpha_2)N} \sum_{i \in \mathcal{T}^t} \max_{\tau, \tau' \in \mathcal{T}_i^t} C_{\alpha_1} \|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty \\
& + 2C_{\alpha_2} C_{\alpha_1} \max_{i \in \mathcal{T}^t, \tau, \tau' \in \mathcal{T}_i^t} \|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty \Bigg)
\end{aligned} \tag{A.61}$$

We plug (A.61) into (A.57) and take expectation of both sides with respect to $z \sim \mathcal{D}$ noting that $\nabla F_{i,\tau}(x_t) = \nabla F_{i,\tau'}(x_t), \forall t, \tau, \tau'$ for the SAA, and use μ-strong convexity of $F(\cdot)$:

$$\begin{aligned}
\mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\|x_{t+1} - x^\star\|^2] \leq & \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\|x_t - x^\star\|^2] + \gamma^2 \\
& - \frac{2\gamma}{\kappa}(1 - Z_t(1+\kappa)) \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\|x_t - x^\star\|] \\
& + \frac{4\gamma^2 L(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1))}{\mu} \\
& + \frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)Nb}} \\
& + \frac{8\gamma C_{\alpha_2}}{\mu} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\max_{i \in \mathcal{T}^t} \|\nabla F_{i,t}(x_t) - \nabla F(x_t)\|_\infty]
\end{aligned} \tag{A.62}$$

$$
\begin{aligned}
&\leq \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|x_t - x^\star\|^2] + \gamma^2 \\
&\quad - \frac{2\gamma}{\kappa}(1 - Z_t(1+\kappa)) \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|x_t - x^\star\|] \\
&\quad + \frac{4\gamma^2 L(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1))}{\mu} \\
&\quad + \frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)Nb}} \\
&\quad + \frac{8\gamma C_{\alpha_2}}{\mu} \inf_{\lambda\in(0,b/a)} \left[\frac{\log\left(2(1-\alpha_2)Nd\right) + b\phi(\lambda/b)}{\lambda}\right],
\end{aligned}
\tag{A.63}
$$

where the second inequality uses Assumption 2.3.1 and Theorem A.2.1 with

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t} \|\nabla F_{i,t}(x_t) - \nabla F(x_t)\|_\infty] = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t, k\in[d]} |[\nabla F_{i,t}(x_t) - \nabla F(x_t)]_k| \\
&= \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t, k\in[d]} \left\{[\nabla F_{i,t}(x_t) - \nabla F(x_t)]_k, [\nabla F(x_t) - \nabla F_{i,t}(x_t)]_k\right\}
\end{aligned}
\tag{A.64}
$$

and

$$
\mathop{\mathbb{E}}_{z\sim D}[e^{\lambda[\nabla F_{i,t}(x_t) - \nabla F(x_t)]_k}] \leq b\phi(\lambda/b), \tag{A.65}
$$

$\forall x_t \in \mathcal{X}, k \in [d], |\lambda| \leq b/a$, with $\phi(\lambda) = \frac{\lambda^2\sigma^2}{2(1-a|\lambda|)}$. Note that the maximization is taken over $2|\mathcal{T}^t|d = 2(1-\alpha_2)Nd$ sub-gamma random variables. The $\inf_{\lambda\in(0,b/a)}[\cdot]$ term attains its minimum at

$$
\lambda^\star = \frac{2\sqrt{\log\left(2(1-\alpha_2)Nd\right)}}{2a\sqrt{\log\left(2(1-\alpha_2)Nd\right)}/b + \sqrt{2\sigma^2/b}} \tag{A.66}
$$

and the expression evaluated at $\lambda^\star$ becomes

$$
\left[\frac{\log\left(2(1-\alpha_2)Nd\right) + b\phi(\lambda/b)}{\lambda}\right]_{\lambda=\lambda^\star} = \frac{a}{b}\log\left(2(1-\alpha_2)Nd\right) + \frac{\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\sqrt{b}} \tag{A.67}
$$

The next step is to take expectation of (A.63) with respect to all randomness, where the challenge is to compute $\mathbb{E}[Z_t\|x_t - x^\star\|]$. However, Z_t is a random variable that depends on

$\{Y_{i,t}\}_{i\in[N]}$, and $Y_{i,t}$ is a random variable which depends on $\{W_{i,\tau}\}_{\tau\in[t-m+1,t]}$. All in all, Z_t is a random variable which depends on the agents' state at time $t-m+1$ (which also not independent of history). Since x_t is also dependent on the agents' state at time $t-m+1$, Z_t and x_t are dependent random variables. Therefore, we can not directly compute $\mathbb{E}[Z_t\|x_t - x^\star\|]$. But considering the fact that the two-state Markov chain governing the agents' states converges exponentially fast to its stationary distribution, the idea is to use total law of expectation by conditioning on the state at time $t-m+1-m_0$ for some $m_0 \geq 0$, i.e.,

$$\mathbb{E}[Z_t\|x_t - x^\star\|] = \mathbb{E}[\mathbb{E}[Z_t\|x_t - x^\star\||\mathcal{S}_{t-m+1-m_0}]], \tag{A.68}$$

where $\mathcal{S}_{t-m+1-m_0} = \{x_{t-m+1-m_0}, \{\pi^i_{t-m+1-m_0}\}_{i\in[N]}\}$ and $\pi^i_{t-m+1-m_0}$ is the distribution of the state of agent i at time $t-m+1-m_0$. Note that due to normalized updates:

$$\|x_t - x^\star\| \leq \|x_{t-m+1-m_0} - x^\star\| + \gamma(m-1+m_0), \tag{A.69}$$

and therefore (A.68) can be rewritten as

$$\begin{aligned}&\mathbb{E}[\mathbb{E}[Z_t\|x_t - x^\star\||\mathcal{S}_{t-m+1-m_0}]]\\ &\leq \mathbb{E}[\|x_{t-m+1-m_0} - x^\star\|\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]] + \mathbb{E}[\gamma(m-1+m_0)\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]].\end{aligned} \tag{A.70}$$

We now use Lemma 2.3.1 (or Lemma A.2.2 in Appendix A.2.8) to establish uniform bounds on $\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]$:

$$\mathbb{E}[\|x_t - x^\star\|Z_t] \leq \mathbb{E}[\|x_{t-m+1-m_0} - x^\star\|]P_Z^m(m_0) + \gamma(m-1+m_0)P_Z^m(m_0) \tag{A.71}$$

$$\leq P_Z^m(m_0)\left(\mathbb{E}[\|x_t - x^\star\|] + 2\gamma(m-1+m_0)\right), \tag{A.72}$$

where the last inequality follows from the normalized updates. Now we take expectation of

(A.63) with respect to all randomness and use (A.72):

$$\begin{aligned}\mathbb{E}[\|x_{t+1}-x^\star\|^2] \leq & \mathbb{E}[\|x_t-x^\star\|^2]+\gamma^2 \\ & -\frac{2\gamma}{\kappa}(1-P_Z^m(m_0)(1+\kappa))\mathbb{E}[\|x_t-x^\star\|] \\ & +4\gamma^2 P_Z^m(m_0)(1+1/\kappa)(m-1+m_0) \\ & +4\gamma^2\kappa(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1)) \\ & +\frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)Nb}}+\frac{8\gamma C_{\alpha_2}}{\mu}\frac{a}{b}\log\left(2(1-\alpha_2)Nd\right) \\ & +\frac{8\gamma C_{\alpha_2}}{\mu}\frac{\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\sqrt{b}}.\end{aligned} \tag{A.73}$$

Since $P_Z^m(m_0) < 1/(1+\kappa)$, the coefficient of $\mathbb{E}[\|x_t - x^\star\|]$ term is negative. Therefore, to upper bound the inequality, we lower bound $\mathbb{E}\left[\|x_t - x^\star\|\right]$ as:

$$\mathbb{E}\left[\|x_t-x^\star\|\right]=\mathbb{E}\left[\frac{\|x_t-x^\star\|^2}{\|x_t-x^\star\|}\right]\geq\frac{\mathbb{E}\left[\|x_t-x^\star\|^2\right]}{R}. \tag{A.74}$$

Using above, we rewrite (A.73) for all $m_0 \in \mathbb{N}_0$ such that $P_Z^m(m_0) < 1/(1+\kappa)$:

$$\begin{aligned}\mathbb{E}[\|x_{t+1}-x^\star\|^2] \leq & \mathbb{E}[\|x_t-x^\star\|^2]\left(1-\frac{2\gamma}{\kappa R}(1-P_Z^m(m_0)(1+\kappa))\right) \\ & +\gamma^2\overline{C}(m_0)+\frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)Nb}} \\ & +\frac{8\gamma C_{\alpha_2}}{\mu}\frac{a}{b}\log\left(2(1-\alpha_2)Nd\right) \\ & +\frac{8\gamma C_{\alpha_2}}{\mu}\frac{\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\sqrt{b}},\end{aligned} \tag{A.75}$$

where

$$\overline{C}(m_0)=1+4P_Z^m(m_0)(1+1/\kappa)(m-1+m_0)+4\kappa(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1)), \tag{A.76}$$

$$c_0(m_0) = \frac{2}{\kappa R}\left(1 - P_Z^m(m_0)\left(1+\kappa\right)\right). \tag{A.77}$$

Note that (A.75) holds for all $t \geq m + m_0$. Hence, we can write:

$$\begin{aligned}
\mathbb{E}\left[\|x_{T+m+m_0} - x^\star\|^2\right] \leq & \mathbb{E}\left[\|x_{m+m_0} - x^\star\|^2\right](1 - c_0(m_0)\gamma)^T \\
& + \frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)Nb}} \sum_{t=m+m_0}^{T+m+m_0-1} \prod_{i=t+1}^{T+m+m_0-1} (1 - c_0(m_0)\gamma) \\
& + \frac{8\gamma C_{\alpha_2}}{\mu} \sum_{t=m+m_0}^{T+m+m_0-1} \prod_{i=t+1}^{T+m+m_0-1} (1 - c_0(m_0)\gamma) \times \\
& \left(\frac{a}{b}\log\left(2(1-\alpha_2)Nd\right) + \frac{\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\sqrt{b}}\right) \\
& + \overline{C}(m_0)\gamma^2 \sum_{t=m+m_0}^{T+m+m_0-1} \prod_{i=t+1}^{T+m+m_0-1} (1 - c_0(m_0)\gamma)
\end{aligned} \tag{A.78}$$

$$\begin{aligned}
= & \mathbb{E}\left[\|x_{m+m_0} - x^\star\|^2\right](1 - c_0(m_0)\gamma)^T \\
& + \frac{4\sigma}{\mu\sqrt{(1-\alpha_2)Nb}} \frac{(1 - (1 - c_0(m_0)\gamma)^T)}{c_0(m_0)} \\
& + \frac{8C_{\alpha_2}}{\mu}\frac{a}{b}\log\left(2(1-\alpha_2)Nd\right)\frac{(1 - (1 - c_0(m_0)\gamma)^T)}{c_0(m_0)} \\
& + \frac{8C_{\alpha_2}}{\mu}\frac{\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\sqrt{b}}\frac{(1 - (1 - c_0(m_0)\gamma)^T)}{c_0(m_0)} \\
& + \overline{C}(m_0)\gamma\frac{(1 - (1 - c_0(m_0)\gamma)^T)}{c_0(m_0)}.
\end{aligned} \tag{A.79}$$

Next, we observe that $(1 - (1 - c_0(m_0)\gamma)^T) \leq 1$ and $\|x_{m+m_0} - x^\star\| \leq (\|x_1 - x^\star\| + (m + m_0 - 1)\gamma)^2$ to get the desired result:

$$\begin{aligned}
\mathbb{E}[\|x_{T+m+m_0} - x^\star\|^2] \leq & \left(\|x_1 - x^\star\| + \gamma(m + m_0 - 1)\right)^2 (1 - c_0(m_0)\gamma)^T \\
& + \frac{4\sigma}{\mu\sqrt{(1-\alpha_2)Nb}c_0(m_0)} + \frac{\overline{C}(m_0)\gamma}{c_0(m_0)} \\
& + \frac{8aC_{\alpha_2}\log\left(2(1-\alpha_2)Nd\right)}{\mu c_0(m_0)b} + \frac{8C_{\alpha_2}\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\mu c_0(m_0)\sqrt{b}}.
\end{aligned} \tag{A.80}$$

A.2.2 Proof of Lemma A.2.1

$$\|e_t\| = \left\| \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} - \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} \right\| \tag{A.81}$$

$$= \left\| \frac{\nabla F(x_t)\|\hat{\hat{g}}_t^{Z_t=0}\| - \hat{\hat{g}}_t^{Z_t=0}\|\nabla F(x_t)\|}{\|\nabla F(x_t)\|\|\hat{\hat{g}}_t^{Z_t=0}\|} \right\| \tag{A.82}$$

$$= \left\| \frac{\|\hat{\hat{g}}_t^{Z_t=0}\|(\nabla F(x_t) - \hat{\hat{g}}_t^{Z_t=0})}{\|\nabla F(x_t)\|\|\hat{\hat{g}}_t^{Z_t=0}\|} + \frac{\hat{\hat{g}}_t^{Z_t=0}(\|\hat{\hat{g}}_t^{Z_t=0}\| - \|\nabla F(x_t)\|)}{\|\nabla F(x_t)\|\|\hat{\hat{g}}_t^{Z_t=0}\|} \right\| \tag{A.83}$$

$$\leq 2\frac{\|\nabla F(x_t) - \hat{\hat{g}}_t^{Z_t=0}\|}{\|\nabla F(x_t)\|} \tag{A.84}$$

Using the sets $\mathcal{T}^t$ and $\{\mathcal{T}_i^t\}_{i\in\mathcal{T}^t}$, we define:

$$\bar{\bar{g}}_t = \frac{1}{(1-\alpha_1)m(1-\alpha_2)N} \sum_{i\in\mathcal{T}^t} \sum_{\tau\in\mathcal{T}_i^t} \nabla F_{i,\tau}(x_\tau), \tag{A.85}$$

$$\bar{\hat{g}}_t = \frac{1}{(1-\alpha_2)N} \sum_{i\in\mathcal{T}^t} \hat{g}_{i,t}. \tag{A.86}$$

Here, $\bar{\bar{g}}$ is the true mean of $(1-\alpha_1)m(1-\alpha_2)N$ trustworthy gradients with time indices $\tau \in \mathcal{T}_t^i$ from agents $i \in \mathcal{T}^t$, and $\bar{\hat{g}}_t$ is the mean of the robustified gradients of agents $i \in \mathcal{T}^t$.

We split the numerator of (A.84) as follows:

$$\|\nabla F(x_t) - \hat{\hat{g}}_t^{Z_t=0}\| \leq \|\nabla F(x_t) - \bar{\bar{g}}_t\| + \|\bar{\bar{g}}_t - \bar{\hat{g}}_t\| + \|\bar{\hat{g}}_t - \hat{\hat{g}}_t^{Z_t=0}\|. \tag{A.87}$$

Next, we upper bound each term above using smoothness of F, normalized updates, and triangular inequality:

1.

$$\|\nabla F(x_t) - \bar{\bar{g}}_t\| = \|\nabla F(x_t) - \frac{1}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\nabla F_{i,\tau}(x_\tau)\| \tag{A.88}$$

$$\begin{aligned}\leq &\|\nabla F(x_t) - \frac{1}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\nabla F_{i,\tau}(x_t)\| \\ &+ \|\frac{1}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\nabla F_{i,\tau}(x_t) - \nabla F_{i,\tau}(x_\tau)\|\end{aligned} \tag{A.89}$$

$$\begin{aligned}\leq &\|\nabla F(x_t) - \frac{1}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\nabla F_{i,\tau}(x_t)\| \\ &+ \frac{L}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\|x_t - x_\tau\|\end{aligned} \tag{A.90}$$

$$\begin{aligned}\leq &\|\nabla F(x_t) - \frac{1}{(1-\alpha_1)m(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\sum_{\tau\in\mathcal{T}_i^t}\nabla F_{i,\tau}(x_t)\| \\ &+ L\gamma(m-1).\end{aligned} \tag{A.91}$$

2.

$$\|\bar{\bar{g}}_t - \bar{\hat{g}}_t\| = \|\frac{1}{(1-\alpha_1)N}\sum_{i\in\mathcal{T}^t}\bar{g}_{i,t} - \hat{g}_{i,t}\| \leq \frac{1}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\|\bar{g}_{i,t} - \hat{g}_{i,t}\| \tag{A.92}$$

$$\begin{aligned}\leq &\frac{1}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\max_{\tau\in\mathcal{T}_i^t}C_{\alpha_1}\|\nabla F_{i,\tau}(x_\tau) \\ &- \frac{1}{(1-\alpha_1)m}\sum_{\tau'\in\mathcal{T}_i^t}\nabla F_{i,\tau'}(x_{\tau'})\|_\infty\end{aligned} \tag{A.93}$$

$$\leq \frac{1}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\max_{\tau,\tau'\in\mathcal{T}_i^t}C_{\alpha_1}\|\nabla F_{i,\tau}(x_\tau) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty \tag{A.94}$$

$$\begin{aligned}\leq &\frac{1}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\max_{\tau,\tau'\in\mathcal{T}_i^t}C_{\alpha_1}\|\nabla F_{i,\tau}(x_\tau) - \nabla F_{i,\tau}(x_{\tau'})\|_\infty \\ &+ \frac{1}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}\max_{\tau,\tau'\in\mathcal{T}_i^t}C_{\alpha_1}\|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty\end{aligned} \tag{A.95}$$

$$
\begin{aligned}
&\leq C_{\alpha_1} L\gamma(m-1) \\
&\quad + \frac{1}{(1-\alpha_2)N} \sum_{i\in\mathcal{T}^t} \max_{\tau,\tau'\in\mathcal{T}_i^t} C_{\alpha_1} \|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty.
\end{aligned} \tag{A.96}
$$

3.

$$
\|\bar{\hat{g}}_t - \hat{\hat{g}}_t^{Z_t=0}\| \leq C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\hat{g}_{i,t} - \frac{1}{(1-\alpha_2)N} \sum_{j\in\mathcal{T}^t} \hat{g}_{j,t}\|_\infty \leq C_{\alpha_2} \max_{i,j\in\mathcal{T}^t} \|\hat{g}_{i,t} - \hat{g}_{j,t}\|_\infty \tag{A.97}
$$

$$
\leq C_{\alpha_2} \max_{i,j\in\mathcal{T}^t} (\|\hat{g}_{i,t} - \bar{g}_{i,t}\|_\infty + \|\hat{g}_{j,t} - \bar{g}_{j,t}\|_\infty + \|\bar{g}_{i,t} - \bar{g}_{j,t}\|_\infty) \tag{A.98}
$$

$$
\leq 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\hat{g}_{i,t} - \bar{g}_{i,t}\| + 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\bar{g}_{i,t} - \nabla F(x_t)\|_\infty \tag{A.99}
$$

$$
\begin{aligned}
&\leq 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\hat{g}_{i,t} - \bar{g}_{i,t}\| \\
&\quad + 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\frac{1}{(1-\alpha_1)m} \sum_{\tau\in\mathcal{T}_i^t} \nabla F_{i,\tau}(x_\tau) - \nabla F_{i,\tau}(x_t)\|_\infty \\
&\quad + 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\frac{1}{(1-\alpha_1)m} \sum_{\tau\in\mathcal{T}_i^t} \nabla F_{i,\tau}(x_t) - \nabla F(x_t)\|_\infty
\end{aligned} \tag{A.100}
$$

$$
\begin{aligned}
&\leq 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\hat{g}_{i,t} - \bar{g}_{i,t}\| + 2C_{\alpha_2} L\gamma(m-1) \\
&\quad + 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\frac{1}{(1-\alpha_1)m} \sum_{\tau\in\mathcal{T}_i^t} \nabla F_{i,\tau}(x_t) - \nabla F(x_t)\|_\infty
\end{aligned} \tag{A.101}
$$

$$
\begin{aligned}
&\leq 2(C_{\alpha_1}+1) C_{\alpha_2} L\gamma(m-1) \\
&\quad + 2C_{\alpha_2} C_{\alpha_1} \max_{i\in\mathcal{T}^t,\tau,\tau'\in\mathcal{T}_i^t} \|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F_{i,\tau'}(x_{\tau'})\|_\infty \\
&\quad + 2C_{\alpha_2} \max_{i\in\mathcal{T}^t} \|\frac{1}{(1-\alpha_1)m} \sum_{\tau\in\mathcal{T}_i^t} \nabla F_{i,\tau}(x_t) - \nabla F(x_t)\|_\infty,
\end{aligned} \tag{A.102}
$$

where the last inequality follows from (A.93)-(A.96). Gathering all three terms, we get the desired result.

A.2.3 Proof of Lemma 2.3.1

The goal is to find a uniform bound on $\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]$ independent of the system state at $t-m+1-m_0$. Note that

$$\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}] = \mathbb{P}(Z_t = 1|\mathcal{S}_{t-m+1-m_0}) = \mathbb{P}\left(\tfrac{1}{N}\textstyle\sum_{i\in[N]} Y_{i,t} > \alpha_2|\mathcal{S}_{t-m+1-m_0}\right) \quad \text{(A.103)}$$

$$\leq \sum_{k=\alpha_2 N+1}^{N} \binom{N}{k} (P_Y^m(m_0))^k (1-P_Y^m(m_0))^{(N-k)}, \quad \text{(A.104)}$$

where

$$P_Y^m(m_0) = P_Y^m(m_0, m, \alpha_1, M) = \max_{i\in[N],t} \mathbb{E}[Y_{i,t}|\mathcal{S}_{t-m+1-m_0}] \quad \text{(A.105)}$$

$$= \max_{i\in[N],t} \mathbb{P}[Y_{i,t} = 1|\mathcal{S}_{t-m+1-m_0}] \quad \text{(A.106)}$$

$$= \max_{i\in[N],t} \mathbb{P}\left(\frac{1}{m}\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1|\mathcal{S}_{t-m+1-m_0}\right) \quad \text{(A.107)}$$

We can rewrite (A.107) using total law of probability:

$$P_Y^m(m_0) = \max_{i\in[N]} \sum_{s=0}^{1} \mathbb{P}\left(\frac{1}{m}\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1 | W_{i,t-m+1} = s\right)$$
$$\times \mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s), \quad \text{(A.108)}$$

where $\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s)$ is the probability that $W_{i,t-m+1} = s$ given the distribution at time $t-m+1-m_0$. Let

$$P_s(m, \alpha_1 m) = \mathbb{P}\left(\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1 m | W_{i,t-m+1} = s\right) \quad \text{(A.109)}$$

Accordingly, we know that:

$$P_0(m, \alpha_1 m) = p_b P_1(m-1, \alpha_1 m) + (1-p_b) P_0(m-1, \alpha_1 m) \tag{A.110}$$

Similarly,

$$\begin{aligned} P_1(m, \alpha_1 m) &= (1-p_t) P_1(m-1, \alpha_1 m - 1) + p_t P_0(m-1, \alpha_1 m - 1) && \text{(A.111)} \\ &\geq (1-p_t) P_1(m-1, \alpha_1 m) + p_t P_0(m-1, \alpha_1 m) && \text{(A.112)} \\ &= P_0(m, \alpha_1 m) + (1-p_t-p_b)(P_1(m-1, \alpha_1 m) - P_0(m-1, \alpha_1 m)) && \text{(A.113)} \end{aligned}$$

Since $(1-p_t-p_b) \geq 0$, $P_1(m, \alpha_1 m) \geq P_0(m, \alpha_1 m)$ if $P_1(m-1, \alpha_1 m) \geq P_0(m-1, \alpha_1 m)$. We know that

$$0 = P_0(\alpha_1 m + 1, \alpha_1 m) \leq P_1(\alpha_1 m + 1, \alpha_1 m) \tag{A.114}$$

because when $W_{i,t-m+1} = 0$, then there can be at most $\alpha_1 m$ instances where $W_{i,\tau} = 1$ for $\tau \in [t-m+1, t-m+1+\alpha_1 m]$. Therefore, the probability of having the sum strictly larger than $\alpha_1 m$ is zero. This establishes that $P_1(m, \alpha_1 m) \geq P_0(m, \alpha_1 m)$. We also have a closed form for

$$\begin{aligned} \mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = 1) =& \pi^i_{t-m+1-m_0}(0) \frac{p_b - p_b(1-p_b-p_t)^{m_0}}{p_b + p_t} \\ &+ \pi^i_{t-m+1-m_0}(1) \frac{p_b + p_t(1-p_b-p_t)^{m_0}}{p_b + p_t} && \text{(A.115)} \\ \leq& \frac{p_b + p_t(1-p_b-p_t)^{m_0}}{p_b + p_t}, && \text{(A.116)} \end{aligned}$$

where the inequality holds with equality if $\pi^i_{t-m+1-m_0}(1) = 1$, i.e., at time $t-m+1-m_0$ the agent was corrupted, which is the worst-case intuition. At this point, we have:

$$P_Y^m(m_0) = \max_{i\in[N]} \sum_{s=0}^{1} P_s(m, \alpha_1 m) \mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s) \tag{A.117}$$

$$\leq \sum_{s=0}^{1} P_s(m, \alpha_1 m) \Pi_{m_0}(i), \tag{A.118}$$

where

$$\Pi_{m_0}(0) = \frac{p_t - p_t(1-p_b-p_t)^{m_0}}{p_b + p_t} \tag{A.119}$$

$$\Pi_{m_0}(1) = \frac{p_b + p_t(1-p_b-p_t)^{m_0}}{p_b + p_t}. \tag{A.120}$$

We have established that the initial distribution given above maximizes $P_Y^m(m_0)$. Next, we argue that $P_Y^m(m_0)$ is larger over the Markov chain governed by M^{m_0} with

$$M_{m_0} = \begin{bmatrix} 1 - p_b'(m_0) & p_b'(m_0) \\ p_t'(m_0) & 1 - p_t'(m_0) \end{bmatrix}, \tag{A.121}$$

where $p_b'(m_0) = p_b + p_t(1-p_b-p_t)^{m_0}$ and $p_t'(m_0) = p_t - p_t(1-p_b-p_t)^{m_0}$. With some abuse of notation, let

$$P_s^{M^m}(m, \alpha_1 m) = P_s^{M^m}\left(\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1 m | W_{i,t-m+1} = s \right), \tag{A.122}$$

where the superscript M^m denotes that the random variable follows the Markov chain governed by M for m time steps. Using $P_1(\cdot) \geq P_0(\cdot)$, we have that

$$P_1^{M^m}(m, \alpha_1 m) = P_1^{M^{m-1}}(m-1, \alpha_1 m - 1)(1-p_t) + P_0^{M^{m-1}}(m-1, \alpha_1 m - 1) p_t \tag{A.123}$$

$$\leq P_1^{M^{m-1}}(m-1, \alpha_1 m - 1)(1 - p_t'(m_0)) + P_0^{M^{m-1}}(m-1, \alpha_1 m - 1)p_t'(m_0) \tag{A.124}$$

$$= P_1^{M_{m_0} M^{m-1}}(m, \alpha_1 m), \tag{A.125}$$

where the superscript $M_{m_0} M^{m-1}$ indicates that the random variable follows the Markov chain governed by M_{m_0} initially and M for the next $m-1$ time steps. Similarly,

$$P_0^{M^m}(m, \alpha_1 m) = P_1^{M^{m-1}}(m-1, \alpha_1 m)p_b + P_0^{M^{m-1}}(m-1, \alpha_1 m)(1 - p_b) \tag{A.126}$$

$$\leq P_1^{M^{m-1}}(m-1, \alpha_1 m - 1)p_b'(m_0) + P_0^{M^{m-1}}(m-1, \alpha_1 m - 1)(1 - p_b'(m_0)) \tag{A.127}$$

$$= P_0^{M_{m_0} M^{m-1}}(m, \alpha_1 m). \tag{A.128}$$

Recursively applying this with the boundary conditions

$$P_s^{M^{\alpha_1 m}}(\alpha_1 m, \alpha_1 m) = P_s^{M_{m_0}^{\alpha_1 m}}(\alpha_1 m, \alpha_1 m) = 0$$

for $s = 1, 2$, we get:

$$P_s^{M^m}(m, \alpha_1 m) \leq P_s^{M_{m_0}^m}(m, \alpha_1 m). \tag{A.129}$$

Therefore, we can upper bound $P_Y^m(m_0)$ as

$$P_Y^m(m_0) \leq \sum_{s=0}^{1} P_s^{M_{m_0}^m}(m, \alpha_1 m)\Pi_{m_0}(s). \tag{A.130}$$

The above is essentially the equal to the following:

$$P_Y^m(m_0) \leq \mathbb{P}_{\Pi_{m_0}}(\frac{1}{m} \sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1), \tag{A.131}$$

where the subscript Π_{m_0} denotes the initial distribution of $W_{i,\tau}$. Since the initial distribution of $W_{i,\tau}$ is equal to the stationary distribution of M^{m_0}, we can use the following theorem to bound

$P_Y^m(m_0)$:

Theorem A.2.2 *[90, Theorem 1] For all pairs $((X_n), f)$, such that (X_n) is a finite, ergodic and reversible Markov chain in stationary state with second largest eigenvalue λ and f is a function taking values in $[0,1]$ such that $E[f(X_i)] = \mu$, the following bounds, with $\lambda_0 = max(0, \lambda)$, hold for all $\epsilon > 0$ such that $\mu + \epsilon < 1$ and all time n*

$$\mathbb{P}(\sum_{i=1}^{n} f(X_i) \geq n(\mu + \epsilon)) \leq \exp\left(-2\frac{1-\lambda_0}{1+\lambda_0} n\epsilon^2\right). \tag{A.132}$$

Applying the above theorem with $\lambda = (1 - p_b - p_t)$, $n = m$, $\mu + \epsilon = \alpha_1 m$, $\mu = \Pi_{m_0}(1)$, and $1 + \lambda_0 = 2 - p_b - p_t \leq 2$:

$$P_Y^m(m_0) \leq \exp\left(-m(\alpha_1 - \Pi_{m_0}(1))^2(p_b + p_t)\right), \tag{A.133}$$

gives the desired bound on $P_Y^m(m_0)$. Note that $P_Z^m(m_0)$ is equal to $1 - F_B(\alpha_2 N, N, P_Y^m(m_0))$, where $F_B(\alpha_2 N, N, P_Y^m(m_0)$ is the cumulative distribution function of the binomial distribution with parameters $(N, P_Y^m(m_0))$ evaluated at $\alpha_2 N$. $F_B(\alpha_2 N, N, P_Y^m(m_0)$ is given by [91]:

$$(N - \alpha_2 N)\binom{N}{\alpha_2 N} \int_0^{1-P_Y^m(m_0)} t^{N-\alpha_2 N-1}(1-t)^{\alpha_2 N} dt, \tag{A.134}$$

which decreases with $P_Y^m(m_0)$. Therefore, $P_Z^m(m_0) = 1 - F_B(\alpha_2 N, N, P_Y^m(m_0))$ is maximized when $P_Y^m(m_0)$ is maximized, which gives the desired bound on $P_Z^m(m_0)$.

A.2.4 Sensitivity Analysis of $P_Y^m(m_0)$ with respect to p_b and p_t

Let $u = (\alpha_1 - \Pi_{m_0}^1)^2(p_b + p_t)$. By chain rule, we have

$$\frac{dP_Y^m(m_0)}{dp_i} = \frac{dP_Y^m(m_0)}{du} \times \frac{du}{dp_i} = -me^{-mu}\frac{du}{dp_i}, \quad i = b, t. \tag{A.135}$$

The derivative of u with respect to p_b is given by:

$$\begin{aligned}\frac{du}{dp_b} = \frac{\alpha_1 - \Pi^1_{m_0}}{p_b + p_t} \times (p_b(\alpha_1 + 2m_0 p_t(1 - p_b - p_t)^{m_0 - 1} - 1) \\ + p_t(\alpha_1 + 2m_0 p_t(1 - p_b - p_t)^{m_0 - 1} + (1 - p_b - p_t)^{m_0} - 2))\end{aligned} \tag{A.136}$$

Noting that $\alpha_1 > \Pi^m_{m_0}$, $p_t \leq p_t + p_b$, and $(1 - x)^{1/x} \leq e^{-1}$ for $x \in [0, 1]$:

$$\begin{aligned}\frac{du}{dp_b} \leq \frac{\alpha_1 - \Pi^1_{m_0}}{p_b + p_t} \times (p_b(\alpha_1 + 2m_0(p_b + p_t)e^{-(m_0 - 1)(p_b + p_t)} - 1) \\ + p_t(\alpha_1 + 2m_0(p_b + p_t)e^{-(m_0 - 1)(p_b + p_t)} + e^{-m_0(p_b + p_t)} - 2))\end{aligned} \tag{A.137}$$

Now, we set $m_0 = k_0/(p_b + p_t)$ for some $k_0 \geq 1$:

$$\begin{aligned}\frac{du}{dp_b} \leq & \frac{\alpha_1 - \Pi^m_{m_0}}{p_b + p_t} \times (p_b(\alpha_1 + 2k_0 e^{-k_0 + p_b + p_t} - 1) \\ & + p_t(\alpha_1 + 2k_0 e^{-k_0 + p_b + p_t} + e^{-k_0} - 2)) && \text{(A.138)} \\ \leq & \frac{\alpha_1 - \Pi^m_{m_0}}{p_b + p_t} \times (p_b(\alpha_1 + 2k_0 e^{-k_0 + 1} - 1) \\ & + p_t(\alpha_1 + 2k_0 e^{-k_0 + 1} + e^{-k_0} - 2)) && \text{(A.139)} \\ < & \, 0, && \text{(A.140)}\end{aligned}$$

where the last inequality holds when $k_0 \geq 4$. Therefore, when $m_0 \geq 4/(p_b + p_t)$, we have that $du/dp_b < 0$, which implies that $dP^m_Y(m_0)/dp_b > 0$. Similarly,

$$\begin{aligned}\frac{du}{dp_t} = & \frac{\alpha_1 - \Pi^1_{m_0}}{p_b + p_t} \times \Big(p_t(\alpha_1 + (2m_0 + p_b + p_t - 1)(1 - p_b - p_t)^{m_0 - 1}) \\ & + p_b(\alpha_1 + 2m_0 p_t(1 - p_b - p_t)^{m_0 - 1} - 2(1 - p_b - p_t)^{m_0} + 1)\Big)\end{aligned} \tag{A.141}$$

Noting that $2(1 - p_b - p_t)^{m_0} \leq 2e^{-m_0(p_b + p_t)} = 2e^{-k_0} < 1$ for all $k_0 \geq 1$, all the summands above are positive. Therefore, we conclude that $du/dp_t > 0$, which implies $dP^m_Y(m_0)/dp_t < 0$.

A.2.5 Proof of Theorem 2.3.2

Due to Assumption 2.3.4, the iterates generated by the algorithm stay in $\mathcal{X}$ without projection. Hence, we proceed to analyze the convergence of the algorithm without projection.

Since $f(\cdot, z)$ is L-smooth, $F(\cdot)$ is L-smooth:

$$
\begin{aligned}
&F(x_{t+1}) - F(x_t) \leq \langle \nabla F(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|^2 \\
\leq& -\gamma \langle \nabla F(x_t), \frac{\hat{\hat{g}}_t}{\|\hat{\hat{g}}_t\|} \rangle + \frac{L\gamma^2}{2} && \text{(A.142)} \\
=& -\gamma(1 - Z_t)\langle \nabla F(x_t), \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} \rangle - \gamma Z_t \langle \nabla F(x_t), \frac{\hat{\hat{g}}_t^{Z_t=1}}{\|\hat{\hat{g}}_t^{Z_t=1}\|} \rangle + \frac{L\gamma^2}{2} && \text{(A.143)} \\
=& -\gamma(1 - Z_t)\langle \nabla F(x_t), \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} \rangle \\
&+ \gamma(1 - Z_t)\langle \nabla F(x_t), \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} - \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|} \rangle && \text{(A.144)} \\
&- \gamma Z_t \langle \nabla F(x_t), \frac{\hat{\hat{g}}_t^{Z_t=1}}{\|\hat{\hat{g}}_t^{Z_t=1}\|} \rangle + \frac{L\gamma^2}{2} \\
\leq& -\gamma(1 - 2Z_t)\|\nabla F(x_t)\| + \gamma(1 - Z_t)\|\nabla F(x_t)\|\|e_t\| + \frac{L\gamma^2}{2}, && \text{(A.145)}
\end{aligned}
$$

where

$$
e_t = \frac{\nabla F(x_t)}{\|\nabla F(x_t)\|} - \frac{\hat{\hat{g}}_t^{Z_t=0}}{\|\hat{\hat{g}}_t^{Z_t=0}\|}. \tag{A.146}
$$

Similar to Proof of Theorem 2.3.1 we define the following sets:

• Define $\mathcal{T}^t$ as the set of $(1 - \alpha_2)N$ agents with smallest indices $i \in [N]$ for which $Y_{i,t} = 0$, i.e.,

$$
\mathcal{T}^t = \{i | i \in [N], Y_{i,t} = 0, \sum_i 1 = (1 - \alpha_2)N\} \tag{A.147}
$$

such that $\sum_{i \in \mathcal{T}^t} i$ is minimized.

• For all agents $i \in \mathcal{T}^t$, define $\mathcal{T}_i^t$ as the set of $(1-\alpha_1)m$ smallest time indices $\tau \in [t-m+1, t]$

for which $W_{i,\tau} = 0$, i.e.,

$$\mathcal{T}_i^t = \{\tau | \tau \in [t-m+1, t], W_{i,\tau} = 0, \sum_{\tau} 1 = (1-\alpha_1)m\} \tag{A.148}$$

such that $\sum_{\tau \in \mathcal{T}_i^t} \tau$ is minimized.

We now use Lemma A.2.1 to bound $\|e_t\|$ and take expectation of both sides with respect to $z \sim \mathcal{D}$ noting that $\nabla F_{i,\tau}(x_t) = \nabla F_{i,\tau'}(x_t), \forall t, \tau, \tau'$:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[F(x_{t+1}) - F(x_t)] \leq -\gamma(1-2Z_t) \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|\nabla F(x_t)\|] + \frac{L\gamma^2}{2} \\
&+ 2L\gamma^2(1-Z_t)(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1)) \\
&+ 2\gamma(1-Z_t) \mathop{\mathbb{E}}_{z\sim\mathcal{D}} \left[\left\| \nabla F(x_t) - \frac{1}{(1-\alpha_2)N} \sum_{i \in \mathcal{T}^t} \nabla F_{i,t}(x_t) \right\| \right] \\
&+ 4\gamma(1-Z_t) C_{\alpha_2} \mathop{\mathbb{E}}_{z\sim\mathcal{D}} [\max_{i \in \mathcal{T}^t} \|\nabla F_{i,t}(x_t) - \nabla F(x_t)\|_\infty] \qquad \text{(A.149)} \\
\leq &- \gamma(1-2Z_t) \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|\nabla F(x_t)\|] + \frac{L\gamma^2}{2} \\
&+ 2L\gamma^2(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1)) \\
&+ 2\gamma \frac{\sigma}{\sqrt{(1-\alpha_2)Nb}} \qquad \text{(A.150)} \\
&+ 4\gamma C_{\alpha_2} \inf_{\lambda \in (0, b/a)} \left[\frac{\log\left(2(1-\alpha_2)Nd\right) + b\phi(\lambda/b)}{\lambda} \right] \\
\leq &- \gamma(1-2Z_t) \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|\nabla F(x_t)\|] + \frac{L\gamma^2}{2} \\
&+ 2L\gamma^2(m-1)(1 + C_{\alpha_1} + 2C_{\alpha_2}(C_{\alpha_1}+1)) \\
&+ 2\gamma \frac{\sigma}{\sqrt{(1-\alpha_2)Nb}} + 4\gamma C_{\alpha_2} \frac{a}{b} \log\left(2(1-\alpha_2)Nd\right) \qquad \text{(A.151)} \\
&+ 4\gamma C_{\alpha_2} \frac{\sigma\sqrt{2\log\left((2(1-\alpha_2)Nd\right)}}{\sqrt{b}},
\end{aligned}$$

where the last two inequalities follow from (A.64)-(A.67) in proof of Theorem 2.3.1.

Next step is to take expectation with respect to all randomness, where the challenge is to

compute $\mathbb{E}[Z_t\|\nabla F(x_t)\|]$. Due to the same reasoning in proof of Theorem 2.3.1, x_t and Z_t are dependent. random variables. Therefore, we use a similar trick and use total law of expectation by conditioning on the state at time $t-m+1-m_0$ for some $m_0 \geq 0$, i.e.,

$$\mathbb{E}[Z_t\|\nabla F(x_t)\|] = \mathbb{E}[\mathbb{E}[Z_t\|\nabla F(x_t)\||\mathcal{S}_{t-m+1-m_0}]], \tag{A.152}$$

where $\mathcal{S}_{t-m+1-m_0} = \{x_{t-m+1-m_0}, \{\pi^i_{t-m+1-m_0}\}_{i\in[N]}\}$ and $\pi^i_{t-m+1-m_0}$ is the distribution of the state of agent i at time $t-m+1-m_0$. Note that due to smoothness of $F(\cdot)$ and normalized updates:

$$\|\nabla F(x_t)\| \leq \|\nabla F(x_{t-m+1-m_0})\| + L\gamma(m-1+m_0), \tag{A.153}$$

and therefore (A.152) can be rewritten as

$$\begin{aligned}\mathbb{E}[\mathbb{E}[Z_t\|\nabla F(x_t)\||\mathcal{S}_{t-m+1-m_0}]] \leq &\mathbb{E}[\|\nabla F(x_{t-m+m_0-1})\|\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]] \\ &+ \mathbb{E}[L\gamma(m-1+m_0)\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]]\end{aligned} \tag{A.154}$$

We now use Lemma 2.3.1 (or Lemma A.2.2 in Appendix A.2.8) to establish uniform bounds on $\mathbb{E}[Z_t|\mathcal{S}_{t-m+1-m_0}]$:

$$\mathbb{E}[\|\nabla F(x_t)\|Z_t] \leq \mathbb{E}[\|\nabla F(x_{t-m+m_0-1})\|]P^m_Z(m_0) + L\gamma(m-1+m_0)P^m_Z(m_0) \tag{A.155}$$

$$\leq P^m_Z(m_0)\left(\mathbb{E}[\|\nabla F(x_t)\|] + 2L\gamma(m-1+m_0)\right), \tag{A.156}$$

where the last inequality follows from smoothness of $F(\cdot)$ and normalized updates. Now we

take expectation of (A.150) with respect to all randomness and use (A.156):

$$\begin{aligned}\mathbb{E}[F(x_{t+1}) - F(x_t)] \leq & -\gamma(1-2P_Z^m(m_0))\mathbb{E}[\|\nabla F(x_t)\|] + \frac{L\gamma^2}{2} \\ & + 4L\gamma^2 P_Z^m(m_0)(m-1+m_0) \\ & + 2L\gamma^2(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1)) \\ & + 2\gamma\frac{\sigma}{\sqrt{(1-\alpha_2)Nb}} + 4\gamma C_{\alpha_2}\frac{a}{b}\log\left(2(1-\alpha_2)Nd\right) \\ & + 4\gamma C_{\alpha_2}\frac{\sigma\sqrt{2\log\left((2(1-\alpha_2)Nd\right)}}{\sqrt{b}}\end{aligned} \tag{A.157}$$

Noting that $F(x_t) \geq F(x^\star)$ for all t, we rearrange the terms, sum from $t = m + m_0$ to $t = T + m + m_0 - 1$ to get the following result for all $m_0 \in \mathbb{N}_0$ for which $P_Z^m(m_0) < 1/2$:

$$\begin{aligned}\frac{1}{T}\sum_{t=m+m_0}^{T+m+m_0-1}\mathbb{E}[\|\nabla F(x_t)\|] \leq & \frac{\mathbb{E}[F(x_{m+m_0})] - F(x^\star)}{\gamma T(1-2P_Z^m(m_0))} + \overline{C}(m_0)\gamma \\ & + \frac{2\sigma}{(1-2P_Z^m(m_0))\sqrt{(1-\alpha_2)Nb}} \\ & + \frac{4C_{\alpha_2}a\log\left(2(1-\alpha_2)Nd\right)}{(1-2P_Z^m(m_0))b} \\ & + \frac{4C_{\alpha_2}\sigma\sqrt{2\log\left((2(1-\alpha_2)Nd\right)}}{(1-2P_Z^m(m_0))\sqrt{b}},\end{aligned} \tag{A.158}$$

where

$$\overline{C}(m_0) = L\Big(0.5 + 4P_Z^m(m_0)(m-1+m_0) + 2(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1))\Big). \tag{A.159}$$

Next, we upper bound $\mathbb{E}[F(x_{m+m_0})]$ using smoothness of F:

$$F(x_{m+m_0}) \leq F(x_1) + \langle \nabla F(x_1), x_{m+m_0} - x_1\rangle + \frac{L}{2}\|x_{m+m_0} - x_1\|^2 \tag{A.160}$$

$$\leq F(x_1) + \|\nabla F(x_1)\|\gamma(m+m_0-1) + \frac{L}{2}\gamma^2(m+m_0-1)^2. \tag{A.161}$$

Finally, we set $\gamma = \gamma_0/\sqrt{T}$ and plug the above inequality into (A.158) to get the final result:

$$\begin{aligned}\frac{1}{T}\sum_{t=m+m_0}^{T+m-1+m_0}\mathbb{E}[\|\nabla F(x_t)\|] \leq & \frac{F(x_1)-F(x^\star)}{\sqrt{T}\gamma_0(1-2P_Z^m(m_0))} + \frac{\overline{C}(m_0)\gamma_0}{\sqrt{T}} \\ & + \frac{\|\nabla F(x_1)\|(m-1+m_0)}{T(1-2P_Z^m(m_0))} + \frac{L\gamma_0(m-1+m_0)^2}{2T^{3/2}(1-2P_Z^m(m_0))} \\ & + \frac{2\sigma}{(1-2P_Z^m(m_0))\sqrt{(1-\alpha_2)Nb}} \\ & + \frac{4C_{\alpha_2}a\log\left(2(1-\alpha_2)Nd\right)}{(1-2P_Z^m(m_0))b} \\ & + \frac{4C_{\alpha_2}\sigma\sqrt{2\log\left((2(1-\alpha_2)Nd\right)}}{(1-2P_Z^m(m_0))\sqrt{b}},\end{aligned} \tag{A.162}$$

A.2.6 Proof of Theorem 2.3.3

The proof is identical to that of Theorem 2.3.1 until (A.57). Next, we plug (A.61) into (A.57) and take expectation of both sides with respect to $z \sim \mathcal{D}$, this time noting that $\nabla F_{i,\tau}(x_t)$ and $\nabla F_{i,\tau'}(x_t)$ are iid random variables $\forall t, \tau, \tau'$ such that $\tau \neq \tau'$ in the SA setting. This results in:

$$\begin{aligned}\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|x_{t+1}-x^\star\|^2] \leq & \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|x_t-x^\star\|^2] + \gamma^2 - \frac{2\gamma}{\kappa}(1-Z_t(1+\kappa))\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|x_t-x^\star\|] \\ & + \frac{4\gamma^2L(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1))}{\mu} \\ & + \frac{4\gamma\sigma}{\mu\sqrt{(1-\alpha_2)N(1-\alpha_1)mb}} \\ & + \frac{8\gamma C_{\alpha_2}}{\mu}\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t}\|\nabla F_{i,t}(x_t)-\nabla F(x_t)\|_\infty] \\ & + \frac{4\gamma C_{\alpha_1}}{\mu(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t}2\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{\tau,\tau'\in\mathcal{T}_i^t}\|\nabla F_{i,\tau}(x_{\tau'})-\nabla F(x_{\tau'})\|_\infty] \\ & + \frac{8\gamma C_{\alpha_1}C_{\alpha_2}}{\mu}2\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t,\tau,\tau'\in\mathcal{T}_i^t}\|\nabla F_{i,\tau}(x_{\tau'})-\nabla F(x_{\tau'})\|_\infty].\end{aligned} \tag{A.163}$$

Next, we use Theorem A.2.1 on the last three terms above, noting that the first maximization is over $2(1-\alpha_2)Nd$ sub-gamma random variables, the second maximization is over $4(1-\alpha_1)md$

sub-gamma random variables, and the last maximization is over $4(1-\alpha_1)m(1-\alpha_2)Nd$ sub-gamma random variables. The rest of the proof is identical to that of Theorem 2.3.1, resulting in:

$$\begin{aligned}\mathbb{E}[\|x_{T+m+m_0}-x^\star\|^2] \leq{}& \left(\|x_1-x^\star\|+\gamma(m+m_0-1)\right)^2\left(1-c_0(m_0)\gamma\right)^T \\ &+\frac{4\sigma}{\mu\sqrt{(1-\alpha_2)N(1-\alpha_1)mb}c_0(m_0)}+\frac{\overline{C}(m_0)\gamma}{c_0(m_0)} \\ &+\frac{8aC_{\alpha_2}\log\left(2(1-\alpha_2)Nd\right)}{\mu c_0(m_0)b} \\ &+\frac{8C_{\alpha_2}\sigma\sqrt{2\log\left(2(1-\alpha_2)Nd\right)}}{\mu c_0(m_0)\sqrt{b}} \\ &+\frac{8aC_{\alpha_1}\log\left(4(1-\alpha_1)md\right)}{\mu c_0(m_0)b} \\ &+\frac{8C_{\alpha_1}\sigma\sqrt{2\log\left(4(1-\alpha_1)md\right)}}{\mu c_0(m_0)\sqrt{b}} \\ &+\frac{16aC_{\alpha_1}C_{\alpha_2}\log\left(4(1-\alpha_2)N(1-\alpha_1)md\right)}{\mu c_0(m_0)b} \\ &+\frac{16C_{\alpha_1}C_{\alpha_2}\sigma\sqrt{2\log\left(4(1-\alpha_2)N(1-\alpha_1)md\right)}}{\mu c_0(m_0)\sqrt{b}},\end{aligned} \tag{A.164}$$

A.2.7 Proof of Theorem 2.3.4

The proof is identical to that of Theorem 2.3.2 until (A.145). Next, we plug (A.61) into (A.145) and take expectation of both sides with respect to $z \sim \mathcal{D}$, this time noting that $\nabla F_{i,\tau}(x_t)$ and $\nabla F_{i,\tau'}(x_t)$ are iid random variables $\forall t, \tau, \tau'$ such that $\tau \neq \tau'$ in the SA setting:

$$\begin{aligned}\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[F(x_{t+1}) - F(x_t)] \leq & -\gamma(1-2Z_t)\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\|\nabla F(x_t)\|] + \frac{L\gamma^2}{2} \\ & + 2L\gamma^2(m-1)(1+C_{\alpha_1}+2C_{\alpha_2}(C_{\alpha_1}+1)) \\ & + 2\gamma\frac{\sigma}{\sqrt{(1-\alpha_2)N(1-\alpha_1)mb}} \\ & + 4\gamma C_{\alpha_2}\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t}\|\nabla F_{i,t}(x_t) - \nabla F(x_t)\|_\infty] \\ & + \frac{2\gamma C_{\alpha_1}}{(1-\alpha_2)N}\sum_{i\in\mathcal{T}^t} 2\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{\tau,\tau'\in\mathcal{T}_i^t}\|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F(x_{\tau'})\|_\infty] \\ & + 4\gamma C_{\alpha_1}C_{\alpha_2} 2\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\max_{i\in\mathcal{T}^t,\tau,\tau'\in\mathcal{T}_i^t}\|\nabla F_{i,\tau}(x_{\tau'}) - \nabla F(x_{\tau'})\|_\infty].\end{aligned} \tag{A.165}$$

Next, we use Theorem A.2.1 on the last three terms above, noting that the first maximization is over $2(1-\alpha_2)Nd$ sub-gamma random variables, the second maximization is over $4(1-\alpha_1)md$ sub-gamma random variables, and the last maximization is over $4(1-\alpha_1)m(1-\alpha_2)Nd$ sub-gamma random variables. The rest of the proof is identical to that of Theorem 2.3.2, resulting

in:

$$\begin{aligned}\frac{1}{T}\sum_{t=m+m_0}^{T+m-1+m_0}\mathbb{E}[\|\nabla F(x_t)\|] \leq & \frac{F(x_1)-F(x^\star)}{\sqrt{T}\gamma_0(1-2P_Z^m(m_0))}+\frac{\overline{C}(m_0)\gamma_0}{\sqrt{T}} \\ & + \frac{\|\nabla F(x_1)\|(m-1+m_0)}{T(1-2P_Z^m(m_0))}+\frac{L\gamma_0(m-1+m_0)^2}{2T^{3/2}(1-2P_Z^m(m_0))} \\ & + \frac{2\sigma}{(1-2P_Z^m(m_0))\sqrt{(1-\alpha_2)N(1-\alpha_1)mb}} \\ & + \frac{4C_{\alpha_2}a\log\left(2(1-\alpha_2)Nd\right)}{(1-2P_Z^m(m_0))b} \\ & + \frac{4C_{\alpha_2}\sigma\sqrt{2\log\left((2(1-\alpha_2)Nd\right)}}{(1-2P_Z^m(m_0))\sqrt{b}} \\ & + \frac{4C_{\alpha_1}a\log\left(4(1-\alpha_1)md\right)}{(1-2P_Z^m(m_0))b} \\ & + \frac{4C_{\alpha_1}\sigma\sqrt{2\log\left(4(1-\alpha_1)md\right)}}{(1-2P_Z^m(m_0))\sqrt{b}} \\ & + \frac{8C_{\alpha_1}C_{\alpha_2}a\log\left(4(1-\alpha_2)N(1-\alpha_1)md\right)}{(1-2P_Z^m(m_0))b} \\ & + \frac{8C_{\alpha_1}C_{\alpha_2}\sigma\sqrt{2\log\left(4(1-\alpha_2)N(1-\alpha_1)md\right)}}{(1-2P_Z^m(m_0))\sqrt{b}}\end{aligned} \tag{A.166}$$

A.2.8 Tighter Bound on $P_Z(m_0)$

Lemma A.2.2 *Given the network and algorithm parameters $(m, N, \alpha_1, \alpha_2, M)$, the following holds for all $m_0 \in \mathbb{Z}^+$:*

$$P_Z(m_0) \leq \sum_{k=\alpha_2 N+1}^{N}\binom{N}{k}(P_Y^m(m_0))^k(1-P_Y^m(m_0))^{(1-k)}, \tag{A.167}$$

where

$$P_Y^m(m_0) = \sum_{s=0}^{1}\sum_{k=\alpha_1 m+1}^{m} r_s(k; m, 1-p_t, p_b)\Pi_{m_0}(s) \tag{A.168}$$

with

$$\begin{aligned} & r_s(k; m, 1-p_t, p_b) \\ &= \sum_{i=0}^{\min(k,n-k)} \binom{n-i}{k}\binom{k}{i}(1-p_t)^{k-i}(1-p_b)^{n-k-i}(1-p_t-p_b)^i \\ &\quad + \sum_{i=0}^{\min(k-1+s,n-k-s)} \binom{n-i-1}{k-1+s}\binom{k-1+s}{i}(1-p_t)^{n-k-i-1+s} \\ &\qquad \times (1-p_b)^{n-k-i-s}(1-p_t-p_b)^{i+1} \end{aligned} \tag{A.169}$$

and

$$\Pi_{m_0}(0) = \frac{p_t - p_t(1-p_b-p_t)^{m_0}}{p_b+p_t} \tag{A.170}$$

$$\Pi_{m_0}(1) = \frac{p_b + p_t(1-p_b-p_t)^{m_0}}{p_b+p_t} \tag{A.171}$$

Proof: The goal here is to find a tighter upper bound on $P_Z(m_0)$ than the bound provided in Lemma 2.3.1. The proof is identical to that of Lemma 2.3.1 until (A.108). A tighter bound is established by deriving the exact expression on $P_Y^m(m_0)$ rather than using a Chernoff's bound. We continue from (A.108):

$$\begin{aligned} P_Y^m(m_0) = \max_{i\in[N]} \sum_{s=0}^{1} & \mathbb{P}\left(\frac{1}{m}\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1 | W_{i,t-m+1} = s\right) \\ & \times \mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s), \end{aligned} \tag{A.172}$$

where $\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s)$ is the probability that $W_{i,t-m+1} = s$ given the distribution at time $t-m+1-m_0$. The first multiplicand in the above equation has a closed form as follows

[198]:

$$\mathbb{P}\left(\frac{1}{m}\sum_{\tau=t-m+1}^{t} W_{i,\tau} > \alpha_1 | W_{i,t-m+1} = s\right)$$
$$= \sum_{k=\alpha_1 m+1}^{m} \mathbb{P}\left(\sum_{\tau=t-m+1}^{t} W_{i,\tau} = k | W_{i,t-m+1} = s\right) = \sum_{k=\alpha_1 m+1}^{m} r_s(k; m, 1-p_t, p_b), \quad \text{(A.173)}$$

where

$$r_s(k; m, 1-p_t, p_b) = \sum_{i=0}^{\min(k,m-k)} \binom{m-i}{k}\binom{k}{i}(1-p_t)^{k-i}(1-p_b)^{m-k-i}(1-p_t-p_b)^i$$
$$+ \sum_{i=0}^{\min(k-1+s,m-k-s)} \binom{m-i-1}{k-1+s}\binom{k-1+s}{i}(1-p_t)^{m-k-i-1+s}$$
$$\times (1-p_b)^{m-k-i-s}(1-p_t-p_b)^{i+1}. \quad \text{(A.174)}$$

Next, we determine $\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = s)$ for $s = \{0, 1\}$ as follows:

$$\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = 0) = \frac{p_t}{p_b + p_t}$$
$$+ \frac{(1-p_b-p_t)^{m_0}}{p_b+p_t}(\pi^i_{t-m+1-m_0}(0)p_b - \pi^i_{t-m+1-m_0}(1)p_t), \quad \text{(A.175)}$$
$$\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = 1) = 1 - P_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = 0) \quad \text{(A.176)}$$

The above probabilities depend on the distribution at time $t - m + 1 - m_0$. Noting that $\sum_{k=1}^{m} r_1(k; \cdot) > \sum_{k=1}^{m} r_0(k; \cdot)$ (as shown in Lemma 1), we upper bound (A.108) by upper bounding $\mathbb{P}_{\pi^i_{t-m+1-m_0}}(W_{i,t-m+1} = 1)$. To do so, we lower bound (A.175) by setting $\pi_{t-m+1-m_0}(1) = 1$, i.e., by assuming that at time $t - m + 1 - m_0$ the agent was Byzantine,

which is the worst-case intuition. All in all we have:

$$P_Y^m(m_0) = \sum_{s=0}^{1} \sum_{k=\alpha_1 m+1}^{m} r_s(k; m, 1-p_t, p_b)\Pi_{m_0}(s), \tag{A.177}$$

where

$$\Pi_{m_0}(0) = \frac{p_t - p_t(1-p_b-p_t)^{m_0}}{p_b+p_t} \tag{A.178}$$

$$\Pi_{m_0}(1) = \frac{p_b + p_t(1-p_b-p_t)^{m_0}}{p_b+p_t} \tag{A.179}$$

■

Appendix B

Supplements to Chapter 3

B.1 Proof of Lemma 3.4.1

By definition, $f_i(x_i)$ is strongly concave over $\mathcal{X}_i$, therefore the optimization problem $\max_{x \in \text{dom} f_i} f_i(x_i) - \langle x_i, p_i \rangle$ is strongly concave and has a unique solution for $p_i \in \mathcal{P}_i$. Since $\mathcal{X}_i \subseteq \text{dom} f_i$ by Assumption 3.2.1, the optimal solution is in the interior of the feasible set. Therefore the first-order optimality condition implies that the optimal solution $g_i(p_i)$ satisfies

$$p_i = \nabla f_i(g_i(p_i)), \tag{B.1}$$

which implies that ∇f_i is surjective for $p_i \in \mathcal{P}_i$. We also know that the gradient of a strongly concave function is injective[1], therefore, ∇f_i is bijective and invertible and $g_i(p_i) = \nabla f_i^{-1}(p_i)$, which also proves that $g_i(p_i)$ is bijective. By the inverse function theorem, we get that:

$$\nabla g_i(p_i) = [\nabla^2 f_i(g_i(p_i))]^{-1}. \tag{B.2}$$

[1]To see this, suppose that $x_1 \neq x_2$ and therefore $\|x_1 - x_2\| > 0$. If $\nabla f(x_1) = \nabla f(x_2)$, (3.1) results in $0 \geq \mu \|x_1 - x_2\|^2$, which is a contradiction and $x_1 = x_2$ must hold.

Since f_i is L-smooth and μ-strongly concave, inverse of it's Hessian has eigenvalues in $[-1/\mu, -1/L]$, which results in

$$\|\nabla g_i(p_i)\| = \|[\nabla^2 f_i(g_i(p_i))]^{-1}\| \leq 1/\mu, \tag{B.3}$$

proving the Lipschitz property of $g_i(p_i)$. To show smoothness, we let $x_i^1 = g_i(p_i^1)$ and $x_i^2 = g_i(p_i^2)$ and write:

$$\|\nabla g_i(p_i^1) - \nabla g_i(p_i^2)\| = \|[\nabla^2 f_i(x_i^1)]^{-1} - [\nabla^2 f_i(x_i^2)]^{-1}\| \tag{B.4}$$

$$= \|[\nabla^2 f_i(x_i^1)]^{-1}(\nabla^2 f_i(x_i^2) - \nabla^2 f_i(x_i^1))[\nabla^2 f_i(x_i^2)]^{-1}\| \tag{B.5}$$

$$\leq \beta\|x_i^1 - x_i^2\|/\mu^2 \leq \beta\|p_i^1 - p_i^2\|/\mu^3, \tag{B.6}$$

where the last inequality uses $1/\mu$-Lipschitz continuity of $g_i(p_i)$, which proves β/μ^3-smoothness of $g_i(p_i)$.

B.2 Proof of Lemma 3.4.2

Firstly we note that by the choice of $\tau \geq \sqrt{\Delta/H_{\mathcal{X}}}$, we can ensure that $\Delta^t \leq H_{\mathcal{X}}$ and that $\mathcal{X}_{\Delta^t}$ is non-empty. Next, we show that $e_i^t \leq 1/(2L),\ \forall t \geq 0$. Note that e_i^t is decreasing with t, and therefore is maximized for $t = 0$:

$$e_i^t \leq e_i^0 = 2\beta\sqrt{d_i}\left(\eta^0 + 2L(d_i - 1)(M\sqrt{n}\gamma^0 + 2\Delta^0\Gamma_{\mathcal{X}})\right)/\mu^3 \tag{B.7}$$

For $\tau \geq 2\mu\Delta\Gamma_{\mathcal{X}}/(M\sqrt{n})$ and $d_i \leq \bar{d}$, we get:

$$e_i^0 \leq \frac{2\beta\sqrt{\bar{d}}}{\mu^3\tau}\left(\frac{M}{8\Gamma_{\mathcal{X}}} + \frac{4L(\bar{d} - 1)M\sqrt{n}}{\mu}\right) \tag{B.8}$$

$$= \beta M \sqrt{\bar{d}} \left(\mu + 32L\Gamma_{\mathcal{X}} \sqrt{n}(\bar{d}-1)\right)/(4\mu^4 \Gamma_{\mathcal{X}} \tau). \tag{B.9}$$

Next, using $\tau \geq L\beta M \sqrt{\bar{d}} \left(\mu + 32L\Gamma_{\mathcal{X}} \sqrt{n}(\bar{d}-1)\right)/(2\mu^4 \Gamma_{\mathcal{X}})$:

$$e_i^t \leq e_i^0 \leq 1/(2L). \tag{B.10}$$

We will prove the lemma by induction that if $\|\hat{\nabla} g_i^k - \nabla g_i(p_i^k)\| \leq e_i^k$ holds for $k \in [\max\{0, t-d_i+1\}, t-1]$, then it holds for $k = t$. Using Cauchy-Schwarz inequality:

$$\|\hat{\nabla} g_i^t - \nabla g_i(p_i^t)\| \leq \sqrt{d_i} \max_{j \in [d_i]} \|[\hat{\nabla} g_i^t]_{:,j} - [\nabla g_i(p_i^t)]_{:,j}\|. \tag{B.11}$$

For a given $j \in [d_i]$, by construction of $\hat{\nabla} g_i^t$ we have

$$[\hat{\nabla} g_i^t]_{:,j} = (g_i(p_i^{\ell_j} + \eta^{\ell_j} e_j) - g_i(p_i^{\ell_j}))/\eta^{\ell_j}, \tag{B.12}$$

for some $\ell_j \in [\max\{0, t-d_i+1\}, t]$. Using the Taylor series expansion, we can rewrite the above as:

$$[\hat{\nabla} g_i^t]_{:,j} = [\nabla g_i(p_i^{\ell_j})]_{:,j} + R_1/\eta^{\ell_j}, \tag{B.13}$$

where $\|R_1\| \leq \beta(\eta^{\ell_j})^2/(2\mu^3)$ follows from [199, Lemma 1] using β/μ^3-smoothness of g_i. Accordingly,

$$\begin{aligned} \|\hat{\nabla} g_i^t - \nabla g_i(p_i^t)\| &\leq \sqrt{d_i} \max_{j \in [d_i]} \|[\nabla g_i(p_i^{\ell_j})]_{:,j} - [\nabla g_i(p_i^t)]_{:,j}\| + \sqrt{d_i}\beta\eta^{\ell_j}/(2\mu^3) \\ &\leq \max_{\ell_j \in [\max\{0, t-d_i+1\}, t]} \frac{\beta\sqrt{d_i}}{\mu^3} \|p_i^{\ell_j} - p_i^t\| + \frac{\sqrt{d_i}\beta\eta^{\ell_j}}{2\mu^3}, \end{aligned} \tag{B.14}$$

where we used

$$\|[\nabla g_i(p_i^{\ell_j})]_{:,j} - [\nabla g_i(p_i^t)]_{:,j}\| \leq \|\nabla g_i(p_i^{\ell_j}) - \nabla g_i(p_i^t)\|, \tag{B.15}$$

for all $j \in [d_i]$, and smoothness of g_i. Furthermore, note that for $\tau \geq 2\bar{d} - 1$, $\eta^{t-d_i+1} \leq 4\eta^t$ and therefore for $t = 0$ we have

$$\|\hat{\nabla} g_i^0 - \nabla g_i(p^0)\| \leq 2\sqrt{d_i}\beta\eta^0/\mu^3 \leq e_i^0. \tag{B.16}$$

Accordingly, the statement holds for $t = 0$, which covers the base case. For $t > 0$, we continue from (B.14) and bound $\|p_i^{\ell_j} - p_i^t\|$ as

$$\|p_i^{\ell_j} - p_i^t\| \leq \sum_{k=\ell_j}^{t-1} \|p_i^k - p_i^{k+1}\| \tag{B.17}$$

$$= \sum_{k=\ell_j}^{t-1} \|[\hat{\nabla} g_i^k]^{-1}(\hat{x}_i^{k+1} - x_i^k)\| \tag{B.18}$$

$$\leq \sum_{k=\ell_j}^{t-1} \|[\hat{\nabla} g_i^k]^{-1}\| \|\hat{x}_i^{k+1} - x_i^k\|. \tag{B.19}$$

The following two lemmas, whose proofs can be found in Appendices B.4 and B.5 bound each of the terms in the above summation:

Lemma B.2.1 *Suppose that* $\|\hat{\nabla} g_i^t - \nabla g_i(p_i^t)\| \leq 1/(2L)$ *for some* t. *Then* $\sigma_{\min}(\hat{\nabla} g_i^t) \geq 1/(2L)$ *and* $\|[\hat{\nabla} g_i^t]^{-1}\| \leq 2L$.

Lemma B.2.2 *For all* $t \geq 0$, *if* $x^t \in \mathcal{X}^{\text{int}}$, *then for a user* $i \in [n]$ *the following holds:*

$$\|\hat{x}_i^{t+1} - x_i^t\| \leq M\sqrt{n}\gamma^t + \Delta^t \Gamma_{\mathcal{X}}. \tag{B.20}$$

Using Lemmas B.2.1 and B.2.2, we get

$$\max_{\ell_j \in [\max\{0, t-d_i+1\}, t]} \|p_i^{\ell_j} - p_i^t\| \leq \max_{\ell_j \in [\max\{0, t-d_i+1\}, t]} 2L \sum_{k=\ell_j}^{t-1} M\sqrt{n}\gamma^k + \Delta^k \Gamma_{\mathcal{X}} \tag{B.21}$$

$$\leq 2L(t - \ell_{\min})(M\sqrt{n}\gamma^{\ell_{\min}} + \Delta^{\ell_{\min}}\Gamma_{\mathcal{X}}), \tag{B.22}$$

where $\ell_{\min} = \max\{0, t - d_i + 1\}$. Lastly, note that $t - \ell_{\min} \leq d_i - 1$, $\gamma^{\ell_{\min}}/\gamma^t \leq 2$, and $\Delta^{\ell_{\min}}/\Delta^t \leq 4$ for $\tau \geq 2\bar{d} - 1$, which gives the final result.

B.3 Proof of Proposition 3.4.2

We will prove by induction that if at iteration t, $\forall k \in [\max\{t - \bar{d} + 1, 0\}, t]$, $x^k \in \mathcal{X}^{\text{int}}_{\frac{\sqrt{n}\eta^k}{\mu}}$, then $x^{t+1} \in \mathcal{X}^{\text{int}}_{\frac{\sqrt{n}\eta^{t+1}}{\mu}}$ and use Assumption 3.3.1 that $x^0 \in \mathcal{X}^{\text{int}}_{\frac{\sqrt{n}\eta^0}{\mu}}$. This will ensure that $x^{t+1,s} \in \mathcal{X}^{\text{int}}$ as well by choice of Δ^t and η^t. Therefore, we assume that $x^k \in \mathcal{X}^{\text{int}}_{\frac{\sqrt{n}\eta^k}{\mu}}$. Note that $\hat{x}^{t+1} \in \mathcal{X}^{\text{int}}$ by definition.

For all $i \in [n]$, we consider a modified utility function $\tilde{f}_i(x_i)$, which is equal to $f_i(x_i)$ if $x_i \in \mathcal{X}_i$, and an L-smooth, μ-strongly concave extension with β-smooth gradient beyond the set $\mathcal{X}_i$. Accordingly, $\text{dom}\tilde{f}_i = \mathbb{R}^{d_i}$, and $\tilde{f}_i$ is L-smooth and μ-strongly concave over $\mathbb{R}^{d_i}$ with β-smooth gradient.

Using the modified utility function, we define the modified price response function

$$\tilde{g}_i(p_i) = \arg\max_{x_i \in \mathbb{R}^{d_i}} \tilde{f}_i(x_i) - \langle x_i, p_i \rangle. \tag{B.23}$$

The following Lemma, whose proof can be found in Appendix B.7, characterizes the regularity properties of $\tilde{g}_i(p_i)$, $\forall i \in [n]$, under Assumption 3.2.2:

Lemma B.3.1 *For all $i \in [n]$, let $\tilde{g}_i(p_i)$ be the modified price response function in* (B.23)*. Then, $\tilde{g}_i(p_i)$ is bijective, $1/\mu$-Lipschitz continuous and β/μ^3-smooth over $\mathbb{R}^{d_i}$. Furthermore,*

let $\mathcal{P}_i = \{p_i \in \mathbb{R}^{d_i} : g_i(p_i) \in \mathcal{X}_i^{\text{int}}\}$. *The following hold true:*

1. *If* $\tilde{g}_i(p_i) \in \mathcal{X}_i^{\text{int}}$, *then* $p_i \in \mathcal{P}_i$.

2. *If* $p_i \in \mathcal{P}_i$, *then* $\tilde{g}_i(p_i) = g_i(p_i)$.

For each user $i \in [n]$, we let $\tilde{x}_i^{t+1} = \tilde{g}_i(p_i^{t+1})$ and we rearrange the price update rule:

$$\tilde{x}_i^{t+1} - \hat{x}_i^{t+1} = \tilde{x}_i^{t+1} - x_i^t - \hat{\nabla} g_i^t (p^{t+1} - p^t). \tag{B.24}$$

We can also write the Taylor expansion of the modified price response function $\tilde{g}_i(p)$ around p_i^t:

$$\tilde{g}_i(p_i^{t+1}) - \tilde{g}_i(p_i^t) = \nabla \tilde{g}_i(p_i^t)(p_i^{t+1} - p_i^t) + R_1. \tag{B.25}$$

We replace $\tilde{g}_i(p_i^t) = g_i(p_i^t) = x_i^t$ and $\nabla \tilde{g}_i(p_i^t) = \nabla g_i(p_i^t)$ (since $p_i^t \in \mathcal{P}_i$) and plug the above equation into (B.24):

$$\tilde{x}_i^{t+1} - \hat{x}_i^{t+1} = (\nabla g_i(p_i^t) - \hat{\nabla} g_i^t)(p_i^{t+1} - p_i^t) + R_1. \tag{B.26}$$

To bound the norm of the above equation, we use Lemma 3.4.2 to bound the norm of the first term and [199, Lemma 1] to bound the second term:

$$\|\tilde{x}_i^{t+1} - \hat{x}_i^{t+1}\| \leq e_i^t \|p_i^{t+1} - p_i^t\| + \frac{\beta}{2\mu^3} \|p_i^{t+1} - p_i^t\|^2. \tag{B.27}$$

Rearranging the price update rule and using Lemmas 3.4.2 and B.2.2 we can bound the norm of the price change:

$$\|p_i^{t+1} - p_i^t\| \leq \|[\hat{\nabla} g_i^t]^{-1}\| \|\hat{x}_i^{t+1} - x_i^t\| \leq 2L(M\sqrt{n}\gamma^t + \Delta^t \Gamma_{\mathcal{X}}). \tag{B.28}$$

Note that both upper bounds for e_i^t and $\|p_i^{t+1} - p_i^t\|$ are decreasing with t. We can bound e_i^t

using $\tau > \frac{2\mu\Delta\Gamma_{\mathcal{X}}}{M\sqrt{n}}$ and $1 \leq \Gamma_{\mathcal{X}}$ as:

$$e_i^t < \beta M\sqrt{d_i n}(\mu/\sqrt{n} + 32L(\bar{d}-1))/(4\mu^4(t+\tau)) \tag{B.29}$$

$$= \beta M\sqrt{d_i n}\gamma^t(\mu/\sqrt{n} + 32L(\bar{d}-1))/(4\mu^3), \tag{B.30}$$

and further upper bound $\|p_i^{t+1} - p_i^t\|$ as

$$\|p_i^{t+1} - p_i^t\| \leq 3LM\sqrt{n}\gamma^t. \tag{B.31}$$

Plugging the above bounds and γ^t into (B.27):

$$\|\tilde{x}_i^{t+1} - \hat{x}_i^{t+1}\| < \frac{3\beta LM^2 n}{4\mu^5(t+\tau)^2}\Big(6L + \sqrt{d_i}\left(\mu/\sqrt{n} + 32L(\bar{d}-1)\right)\Big). \tag{B.32}$$

Next, using Cauchy-Schwarz inequality, we bound $\|\tilde{x}^{t+1} - x^{t+1}\|$ as

$$\|\tilde{x}^{t+1} - \hat{x}^{t+1}\| < \frac{3\beta LM^2 n^{3/2}}{4\mu^5(t+\tau)^2}\Big(6L + \sqrt{d}\left(\mu/\sqrt{n} + 32L(\bar{d}-1)\right)\Big) \tag{B.33}$$

$$= 3\Delta^t/4, \tag{B.34}$$

where we used the definition of Δ^t and $\sum_{i\in[n]}\sqrt{d_i} \leq \sqrt{dn}$. This establishes that by definition of a shrunk set and $\Delta^t/4 = \frac{\sqrt{n}\eta^{t+1}}{\mu}$, $\tilde{x}^{t+1} \in \mathcal{X}^{\text{int}}_{\frac{\sqrt{n}\eta^{t+1}}{\mu}}$. Furthermore, let $\tilde{x}_i^{t+1,s} = \tilde{g}_i(p_i^{t+1,s})$. Using $1/\mu$-Lipschitz continuity of $\tilde{g}_i(p_i)$:

$$\|\tilde{x}_i^{t+1,s} - \tilde{x}_i^{t+1}\| \leq \Delta^t/(4\sqrt{n}), \tag{B.35}$$

and $\|\tilde{x}^{t+1,s} - \tilde{x}^{t+1}\| \leq \Delta^t/4$. Accordingly, we have

$$\|\tilde{x}^{t+1,s} - \hat{x}^{t+1}\| < \Delta^t, \tag{B.36}$$

which establishes that $\tilde{x}^{t+1,s} \in \mathcal{X}^{\text{int}}$.

Lastly, note that if $\tilde{x}^{t+1}, \tilde{x}^{t+1,s} \in \mathcal{X}^{\text{int}}$, then for all $i \in [n]$, $\tilde{x}_i^{t+1}, \tilde{x}_i^{t+1,s} \in \mathcal{X}_i^{\text{int}}$, or equivalently, $\tilde{g}_i(p_i^{t+1}), \tilde{g}_i(p_i^{t+1,s}) \in \mathcal{X}_i^{\text{int}}$. Using Lemma B.3.1 we have that $p_i^{t+1}, p_i^{t+1,s} \in \mathcal{P}_i$, $\forall i \in [n]$. Hence, $\tilde{g}_i(p^{t+1}) = g_i(p^{t+1})$ and $\tilde{x}_i^{t+1} = x_i^{t+1}$ as well as $\tilde{g}_i(p^{t+1,s}) = g_i(p^{t+1,s})$ and $\tilde{x}_i^{t+1,s} = x_i^{t+1,s}$ for all $i \in [n]$, which proves the proposition.

B.4 Proof of Lemma B.2.1

Note that for $p_i^t \in \mathcal{P}_i$, $\nabla g_i(p^t) = [\nabla^2 f_i(g_i(p^t))]^{-1}$ is symmetric by Schwarz's theorem, since $\nabla^2 f_i(g_i(p_i))$ is β-Lipschitz continuous for $p_i \in \mathcal{P}_i$. Accordingly, the minimum singular value of $\nabla g_i(p_i^t)$ is equal to smallest absolute eigenvalue of $[\nabla^2 f_i(g_i(p^t))]^{-1}$, i.e., $\sigma_{\min}(\nabla g_i(p_i^t)) = 1/L$. This implies that if $\|\hat{\nabla} g_i^t - \nabla g_i(p_i^t)\| \leq 1/(2L)$ holds, then

$$\sigma_{\min}(\hat{\nabla} g_i^t) = \min_{\|x\|=1} \|\hat{\nabla} g_i^t x\| = \min_{\|x\|=1} \|\nabla g_i(p_i^t)x + (\hat{\nabla} g_i^t - \nabla g_i(p_i^t))x\| \tag{B.37}$$

$$\geq \min_{\|x\|=1} \|\nabla g_i(p_i^t)x\| - \max_{\|x\|=1} \|(\hat{\nabla} g_i^t - \nabla g_i(p_i^t))x\| \tag{B.38}$$

$$= 1/L - 1/(2L) \geq 1/(2L), \tag{B.39}$$

which implies that $\|[\hat{\nabla} g_i^t]^{-1}\| = 1/\sigma_{\min}(\hat{\nabla} g_i^t) \leq 2L$.

B.5 Proof of Lemma B.2.2

To bound $\|\hat{x}_i^{t+1} - x_i^t\|$, we will use the following as an auxiliary result:

Theorem B.5.1 *[118, Theorem 1.2.1] Let $\mathcal{X}$ be a convex and compact set in $\mathbb{R}^d$. Then, the metric projection onto $\mathcal{X}$ is contracting, that is,*

$$\|\Pi_{\mathcal{X}}(x) - \Pi_{\mathcal{X}}(y)\| \leq \|x - y\|, \ \forall x, y, \in \mathbb{R}^d.$$

Using the above result, we bound $\|\hat{x}_i^{t+1} - x_i^t\|$ as:

$$\|\hat{x}_i^{t+1} - x_i^t\| \leq \|\hat{x}^{t+1} - x^t\| \tag{B.40}$$

$$= \|\Pi_{\mathcal{X}_{\Delta^t}}(x^t + p^t\gamma^t) - \Pi_{\mathcal{X}_{\Delta^t}}(x^t) + \Pi_{\mathcal{X}_{\Delta^t}}(x^t) - x^t\| \tag{B.41}$$

$$\leq \|\Pi_{\mathcal{X}_{\Delta^t}}(x^t + p^t\gamma^t) - \Pi_{\mathcal{X}_{\Delta^t}}(x^t)\| + \|\Pi_{\mathcal{X}_{\Delta^t}}(x^t) - x^t\| \tag{B.42}$$

$$\leq \|p^t\gamma^t\| + \Delta^t\Gamma_{\mathcal{X}} \leq M\sqrt{n}\gamma^t + \Delta^t\Gamma_{\mathcal{X}}, \tag{B.43}$$

where we used $\|p_i^t\| = \|\nabla f_i(x_i^t)\| \leq M$ since $x_i^t \in \mathcal{X}_i^{\text{int}}$, and Proposition 3.4.1.

B.6 Proof of Theorem 3.4.1

We denote the regret incurred by the update stage as $R_u(T) = \sum_{t=1}^{T/2} f(x^\star) - f(x^t)$ and the regret incurred by the sampling stage as $R_s(T) = \sum_{t=1}^{T/2} f(x^\star) - f(x^{t,s})$. Let $y^{t+1} = x^t + \gamma^t p_t$. By Lemma B.3.1, we know that $p^t = \nabla f(x^t)$, $\forall t \geq 0$, since $x^t \in \mathcal{X}^{\text{int}}$ by Proposition 3.4.2. For $t \geq 1$, we write using strong concavity:

$$f(x^\star) - f(x^t) \leq \langle -\nabla f(x^t), x^t - x^\star\rangle - \frac{\mu}{2}\|x^t - x^\star\|^2 \tag{B.44}$$

$$= \frac{1}{\gamma^t}\langle x^t - y^{t+1}, x^t - x^\star\rangle - \frac{\mu}{2}\|x^t - x^\star\|^2 \tag{B.45}$$

$$= \frac{1}{2\gamma^t}\left(\|x^t - y^{t+1}\|^2 + \|x^t - x^\star\|^2 - \|y^{t+1} - x^\star\|^2\right) - \frac{\mu}{2}\|x^t - x^\star\|^2. \tag{B.46}$$

Next, we bound the $\|y^{t+1} - x^\star\|^2$ term using Theorem B.5.1 as follows:

$$\|y^{t+1} - x^\star\|^2 \geq \|\Pi_{\mathcal{X}_{\Delta^t}}(y^{t+1}) - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\|^2 = \|\hat{x}^{t+1} - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\|^2 \tag{B.47}$$

$$= \|\hat{x}^{t+1} - x^{t+1} + x^{t+1} - x^\star + x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\|^2 \tag{B.48}$$

$$= \|\hat{x}^{t+1} - x^{t+1}\|^2 + \|x^{t+1} - x^\star\|^2 + \|x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\|^2$$

$$
\begin{aligned}
&+ 2\langle \hat{x}^{t+1} - x^{t+1}, x^{t+1} - x^\star \rangle + 2\langle x^{t+1} - x^\star, x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star) \rangle \\
&+ 2\langle x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star), \hat{x}^{t+1} - x^{t+1} \rangle && \text{(B.49)} \\
\geq &\|x^{t+1} - x^\star\|^2 - 2\|\hat{x}^{t+1} - x^{t+1}\| \|x^{t+1} - x^\star\| \\
&- 2\|x^{t+1} - x^\star\| \|x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\| && \text{(B.50)} \\
&- 2\|x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\| \|\hat{x}^{t+1} - x^{t+1}\| \\
\geq &\|x^{t+1} - x^\star\|^2 - 2\Delta^t R(\Gamma_{\mathcal{X}} + 3/4) - 3/2(\Delta^t)^2 \Gamma_{\mathcal{X}} && \text{(B.51)} \\
:= &\|x^{t+1} - x^\star\|^2 - C_t, && \text{(B.52)}
\end{aligned}
$$

where the last inequality uses $\|x^{t+1} - \hat{x}^{t+1}\| < 3\Delta^t/4$ given by Proposition 3.4.2 and Proposition 3.4.1 to bound $\|x^\star - \Pi_{\mathcal{X}_{\Delta^t}}(x^\star)\|$. Plugging this in (B.46):

$$
f(x^\star) - f(x^t) \leq \frac{M^2 n \gamma^t}{2} - \frac{\mu}{2}\|x^t - x^\star\|^2 + \frac{C^t}{2\gamma^t} + \frac{1}{2\gamma^t}(\|x^t - x^\star\|^2 - \|x^{t+1} - x^\star\|^2). \quad \text{(B.53)}
$$

Summing from $t = 1$ to $T/2$ telescopes the $\|x^t - x^\star\|^2$ terms:

$$
\begin{aligned}
nR_u(T) \leq &\frac{M^2 n \log(T/2)}{2\mu} + \frac{\mu\tau}{2}\|x^1 - x^\star\|^2 + \sum_{t=2}^{T/2} \left(\frac{1}{2\gamma^t} - \frac{1}{2\gamma^{t-1}} - \frac{\mu}{2} \right) \|x^t - x^\star\|^2 \\
&- \frac{1}{2\gamma^{T/2}}\|x^{T/2+1} - x^\star\|^2 + \sum_{t=1}^{T/2} \frac{C^t}{2\gamma^t} && \text{(B.54)} \\
\leq &\frac{M^2 n \log(T/2)}{2\mu} + \frac{\mu\tau}{2}\|x^1 - x^\star\|^2 + \sum_{t=1}^{T/2} \frac{C^t}{2\gamma^t}. && \text{(B.55)}
\end{aligned}
$$

Finally, note that $C^t = \mathcal{O}(1/t^2)$ because the it consists of terms Δ^t and $(\Delta^t)^2$. Therefore, we can use the bounds $\sum_{t=1}^{T/2} \frac{1}{t+\tau} \leq \sum_{t=1}^{T/2} \frac{1}{t+2} \leq \log(T/2)$ and for $k \geq 2$, $\sum_{t=1}^{T/2} \frac{1}{(t+2)^k} \leq 1$ to show

that:

$$\sum_{t=1}^{T/2} \frac{C^t}{2\gamma^t} \leq \mu\Delta R(3/4 + \Gamma_{\mathcal{X}}) \log(T/2) + 3\mu\Delta^2\Gamma_{\mathcal{X}}/4. \tag{B.56}$$

Plugging (B.56) into (B.55) and dividing by both sides by n, we get the regret incurred by the update stages. For the sampling stages, we note that due to the strong concavity of f

$$f(x^t) - f(x^{t,s}) \leq \langle \nabla f(x^{t,s}), x^t - x^{t,s} \rangle \leq M\sqrt{n}\frac{\Delta^{t-1}}{4}. \tag{B.57}$$

Accordingly $f(x^\star) - f(x^{t,s}) \leq f(x^\star) - f(x^t) + M\sqrt{n}\Delta^{t-1}/4$. Summing from $t = 1$ to $T/2$, we get

$$nR_s(T) = nR_u(T) + \frac{M}{4}\sum_{t=1}^{T} \Delta^{t-1} \leq nR_u(T) + \frac{\Delta M\sqrt{n}}{4}, \tag{B.58}$$

which gives the final result as

$$R(T) \leq 2R_u(T) + \Delta M/(4\sqrt{n}). \tag{B.59}$$

To get the convergence result, we rearrange (B.53):

$$\|x^{t+1} - x^\star\|^2 \leq \|x^t - x^\star\|^2(1 - \mu\gamma^t) + M^2 n(\gamma^t)^2 + C^t + 2\gamma^t(f(x^t) - f(x^\star)) \tag{B.60}$$

$$\leq \|x^t - x^\star\|^2(1 - \mu\gamma^t) + M^2 n(\gamma^t)^2 + C^t. \tag{B.61}$$

We get an equation like the above for all $t \geq 0$. We multiply each by $(1-\mu\gamma^{t+1})$ for $t < T/2-1$ and sum them from $t = 0$ to $t = T/2 - 1$ to get:

$$\begin{aligned}\|x^{T/2} - x^\star\|^2 \leq & \|x^0 - x^\star\|^2 \prod_{t=0}^{T/2-1}(1-\mu\gamma^t) + M^2 n \sum_{t=0}^{T/2-1}(\gamma^t)^2 \prod_{i=t+1}^{T/2-1}(1-\mu\gamma^i) \\ & + \sum_{t=0}^{T/2-1} C^t \prod_{i=t+1}^{T/2-1}(1-\mu\gamma^i) \end{aligned} \tag{B.62}$$

$$\begin{aligned} \leq & \|x^0 - x^\star\|^2 \frac{\tau - 1}{\tau - 1 + T/2} + \frac{M^2 n \log(T/2)}{\mu^2(T/2 + \tau - 1)} \\ & + \frac{2R(3/4 + \Gamma_{\mathcal{X}})\Delta \log(T/2)}{(T/2 + \tau - 1)} + \frac{3\Delta^2\Gamma_{\mathcal{X}}}{2(T/2 + \tau - 1)}. \end{aligned} \tag{B.63}$$

which completes the proof.

B.7 Proof of Lemma B.3.1

. The first part of the lemma follows from the same steps as in Lemma 3.4.1 for $p_i \in \mathbb{R}^{d_i}$ instead of $p_i \in \mathcal{P}_i$, and using $\tilde{f}_i$ and $\tilde{g}_i$ instead of f_i and g_i.

Next, we prove the second part of the lemma. For the first statement, given a $p_i \in \mathbb{R}^{d_i}$, suppose that $\tilde{g}_i(p_i) \in \mathcal{X}_i^{\text{int}}$. This implies that there exists $x_i \in \mathcal{X}_i^{\text{int}}$ that satisfies $\nabla \tilde{f}_i(x_i) = p_i$. Since $\tilde{f}_i(x_i) = f_i(x_i)$ for $x_i \in \mathcal{X}_i^{\text{int}}$, the same x_i solves the optimization problem in (3.5), which implies $gx_i(p_i) = \tilde{g}_i(p_i)$. Therefore, $g_i(p_i) \in \mathcal{X}_i^{\text{int}}$, which proves $p_i \in \mathcal{P}_i$ by definition.

To prove the second statement, note that if $p_i \in \mathcal{P}_i$, then $g_i(p_i) \in \mathcal{X}_i^{\text{int}}$. Since $\mathcal{X}_i \subseteq \text{dom} f_i$ by Assumption 3.2.1, the first order optimality condition of (3.5) implies that there exists $x_i = g_i(p_i) \in \mathcal{X}_i^{\text{int}}$ such that $\nabla f_i(x_i) = p_i$. The same x_i solves the optimization problem (B.23), since $f_i(x_i) = \tilde{f}_i(x_i)$ for $x_i \in \mathcal{X}_i^{\text{int}}$. The optimal solution to (B.23) has to be unique due to strong concavity, therefore it must hold true that $\tilde{g}_i(p_i) = g_i(p_i)$.

B.8 Proof of Remark 3.3.2

For a user $i \in [n]$, using the modified price response function $\tilde{g}_i(p_i)$ introduced in the proof of Proposition 3.4.2, we have that

$$\|\tilde{x}_i^{-t} - x_i^0\| \leq \eta^0/\mu, \ \forall t \in [-d_i, -1], \tag{B.64}$$

which implies that $\tilde{x}_i^{-t} \in \mathcal{X}_i^{\text{int}}$ because $x^0 \in \mathcal{X}^{\text{int}}_{\frac{\eta^0\sqrt{n}}{\mu}}$. As such, $\tilde{x}_i^{-t} = x_i^{-t}$ and $p_i^{-t} = \nabla f_i(x_i^{-t})$.

Appendix C

Supplements to Chapter 4

C.1 Proofs for Results in Section 4.2

C.1.1 Proof of Proposition 4.2.1

To prove Proposition 4.2.1, we first formulate the static optimization problem via a network flow model that characterizes the *capacity region* of the network for a given set of prices $\ell_{ij}(t) = \ell_{ij}\ \forall t$ (Hence, $\Lambda_{ij}(t) = \Lambda_{ij}\ \forall t$). The capacity region is defined as the set of all arrival rates $[\Lambda_{ij}]_{i,j\in\mathcal{M}}$, where there exists a charging and routing policy under which the queueing network of the system is stable. Let x_i^v be the number of vehicles available at node i, α_{ij}^v be the fraction of vehicles at node i with energy level v being routed to node j, and α_{ic}^v be the fraction of vehicles charging at node i starting with energy level v. We say the static vehicle allocation for node i and energy level v is feasible if $\alpha_{ic}^v + \sum_{\substack{j\in\mathcal{M}\\ j\neq i}} \alpha_{ij}^v \leq 1$.

The optimization problem that characterizes the capacity region of the network ensures that the total number of vehicles routed from i to j is at least as large as the nominal arrival rate to

the queue (i, j). Namely, the vehicle allocation problem can be formulated as follows:

$$\min_{x_i^v, \alpha_{ij}^v, \alpha_{ic}^v} \quad \rho \tag{C.1a}$$

$$\text{subject to} \quad \Lambda_{ij} \leq \sum_{v=v_{ij}}^{v_{\max}} x_i^v \alpha_{ij}^v \quad \forall i, j \in \mathcal{M}, \tag{C.1b}$$

$$\rho \geq \alpha_{ic}^v + \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} \alpha_{ij}^v \quad \forall i \in \mathcal{M},\ \forall v \in \mathcal{V}, \tag{C.1c}$$

$$x_i^v = x_i^{v-1} \alpha_{ic}^{v-1} + \sum_{j \in \mathcal{M}} x_i^{v+v_{ji}} \alpha_{ji}^{v+v_{ji}} \quad \forall i \in \mathcal{M},\ \forall v \in \mathcal{V}, \tag{C.1d}$$

$$\alpha_{ic}^{v_{\max}} = 0 \quad \forall i \in \mathcal{M}, \tag{C.1e}$$

$$\alpha_{ij}^v = 0 \quad \forall v < v_{ij},\ \forall i, j \in \mathcal{M} \tag{C.1f}$$

$$x_i^v \geq 0,\ \alpha_{ij}^v \geq 0\ \alpha_{ic}^v \geq 0,\ \forall i, j \in \mathcal{M},\ \forall v \in \mathcal{V}, \tag{C.1g}$$

$$x_i^v = \alpha_{ic}^v = \alpha_{ij}^v = 0 \quad \forall v \notin \mathcal{V},\ \forall i, j \in \mathcal{M}. \tag{C.1h}$$

The constraint (C.1c) upper bounds the allocation of vehicles for each node i and energy level v. The constraints (C.1d)-(C.1f) are similar to those of optimization problem (4.1) with $x_i^v = x_{ic}^v + \sum_{j \in \mathcal{M}} x_{ij}^v$, $\alpha_{ic}^v = x_{ic}^v / x_i^v$, and $\alpha_{ij}^v = x_{ij}^v / x_i^v$.

Lemma C.1.1 *Let the optimal value of* (C.1) *be* ρ^**. Then,* $\rho^* \leq 1$ *is a necessary and sufficient condition of rate stability of the system under some routing and charging policy.*

Proof: Consider the fluid scaling of the queueing network, $Q_{ij}^{rt} = \frac{q_{ij}(\lfloor rt \rfloor)}{r}$ (see [200] for more discussion on the stability of fluid models), and let Q_{ij}^t be the corresponding fluid limit. The fluid model dynamics is as follows:

$$Q_{ij}^t = Q_{ij}^0 + A_{ij}^t - X_{ij}^t,$$

where A_{ij}^t is the total number of riders from node i to node j that have arrived to the network

until time t and X_{ij}^t is the total number of vehicles routed from node i to j up to time t. Suppose that $\rho^* > 1$ and there exists a policy under which for all $t \geq 0$ and for all origin-destination pairs (i,j), $Q_{ij}^t = 0$. Pick a point t_1, where $Q_{ij}^{t_1}$ is differentiable for all (i,j). Then, for all (i,j), $\dot{Q}_{ij}^{t_1} = 0$. Since $\dot{A}_{ij}^{t_1} = \Lambda_{ij}$, this implies $\dot{X}_{ij}^{t_1} = \Lambda_{ij}$. On the other hand, $\dot{X}_{ij}^{t_1}$ is the total number of vehicles routed from i to j at t_1. This implies $\Lambda_{ij} = \sum_{v=v_{ij}}^{v_{\max}} x_i^v \alpha_{ij}^v$ for all (i,j) and there exists α_{ij}^v and α_{ic}^v at time t_1 such that the flow balance constraints hold and the allocation vector $[\alpha_{ij}^v \ \alpha_{ic}^v]$ is feasible, i.e. $\alpha_{ic}^v + \sum_{\substack{j=1 \\ j\neq i}}^{m} \alpha_{ij}^v \leq 1$. This contradicts $\rho^* > 1$.

Now suppose $\rho^* \leq 1$ and $\alpha^* = [\alpha_{ij}^{v*} \ \alpha_{ic}^{v*}]$ is an allocation vector that solves the static problem. The cumulative number of vehicles routed from node i to j up to time t is

$$S_{ij}^t = \sum_{v=v_{ij}}^{v_{\max}} x_i^v \alpha_{ij}^v t = \sum_{v=0}^{v_{\max}} x_i^v \alpha_{ij}^v t \geq \Lambda_{ij} t$$

. Suppose that for some origin-destination pair (i,j), the queue $Q_{ij}^{t_1} \geq \epsilon > 0$ for some positive t_1 and ϵ. By continuity of the fluid limit, there exists $t_0 \in (0, t_1)$ such that $Q_{ij}^{t_0} = \epsilon/2$ and $Q_{ij}^t > 0$ for $t \in [t_0, t_1]$. Then, $\dot{Q}_{ij}^t > 0$ implies $\Lambda_{ij} > \sum_{v=v_{ij}}^{v_{\max}} x_i^v \alpha_{ij}^v$, which is a contradiction. ■

By Lemma C.1.1, the *capacity region* C_Λ of the network is the set of all $\Lambda_{ij} \in \mathbb{R}^+$ for which the corresponding optimal solution to the optimization problem (C.1) satisfies $\rho^* \leq 1$. As long as $\rho^* \leq 1$, there exists a routing and charging policy such that the queues will be bounded away from infinity.

The platform operator's goal is to maximize its profits by setting prices and making routing and charging decisions such that the system remains stable. In its most general form, the problem can be formulated as follows:

$$\begin{aligned} \max_{\ell_{ij}, x_i^v, \alpha_{ij}^v, \alpha_{ic}^v} \quad & U(\Lambda_{ij}(\ell_{ij}), x_i^v, \alpha_{ij}^v, \alpha_{ic}^v) \\ \text{subject to} \quad & [\Lambda_{ij}(\ell_{ij})]_{i,j\in\mathcal{M}} \in C_\Lambda, \end{aligned} \tag{C.2}$$

where $U(\cdot)$ is the utility function that depends on the prices, demand for rides and the vehicle decisions.

Recall that $x_{ic}^v = x_i^v \alpha_{ic}^v$ and $x_{ij}^v = x_i^v \alpha_{ij}^v$. Using these variables and noting that $\alpha_{ic}^v + \sum_{j \in \mathcal{M}} \alpha_{ij}^v = 1$ when $\rho^* \leq 1$, the platform operator's profit maximization problem can be stated as (4.1). A feasible solution of (4.1) guarantees rate stability of the system, since the corresponding vehicle allocation problem (C.1) has solution $\rho^* \leq 1$.

C.1.2 Proof of Proposition 4.2.2

For brevity of notation, let $\beta + p_i = P_i$. Let ν_{ij} be the dual variables corresponding to the demand satisfaction constraints and μ_i^v be the dual variables corresponding to the flow balance constraints. Since the optimization problem (4.1) is a convex quadratic maximization problem (given a with uniform $F(\cdot)$) and Slater's condition is satisfied, strong duality holds. We can write the dual problem as:

$$\min_{\nu_{ij}, \mu_i^v} \max_{\ell_{ij}} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\lambda_{ij} (1 - \frac{\ell_{ij}}{\ell_{\max}}) (\ell_i - \nu_{ij}) \right) \tag{C.3a}$$

$$\text{subject to} \quad \nu_{ij} \geq 0, \tag{C.3b}$$

$$\nu_{ij} + \mu_i^v - \mu^{v - v_{ij}} - \beta \tau_{ij} \leq 0, \tag{C.3c}$$

$$\mu_i^v - \mu_i^{v+1} - P_i \leq 0 \quad \forall i, j, v. \tag{C.3d}$$

For fixed ν_{ij} and μ_i^v, the inner maximization results in the optimal prices:

$$\ell_{ij}^* = \frac{\ell_{\max} + \nu_{ij}}{2}. \tag{C.4}$$

By strong duality, the optimal primal solution satisfies the dual solution with optimal dual variables ν_{ij}^* and μ_i^{v*}, which completes the first part of the proposition. The dual problem with

optimal prices in (C.4) can be written as:

$$\min_{\nu_{ij},\mu_i^v} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{\lambda_{ij}}{\ell_{\max}} \left(\frac{\ell_{\max} - \nu_{ij}}{2}\right)^2 \tag{C.5a}$$

$$\text{subject to} \quad \nu_{ij} \geq 0, \tag{C.5b}$$

$$\nu_{ij} + \mu_i^v - \mu_j^{v-v_{ij}} - \beta\tau_{ij} \leq 0, \tag{C.5c}$$

$$\mu_i^v - \mu_i^{v+1} - P_i \leq 0 \quad \forall i, j, v. \tag{C.5d}$$

The objective function in (C.5a) with optimal dual variables, along with (C.4) suggests:

$$P = \sum_{i=1}^{m}\sum_{j=1}^{m} \frac{\lambda_{ij}}{\ell_{\max}} (\ell_{\max} - \ell_{ij}^*)^2,$$

where profits P is the value of the objective function of both optimal and dual problems. To get the upper bound on prices, we go through the following algebraic calculations using the constraints. The inequality (C.5d) gives:

$$\mu_i^{v-v_{ji}} \leq v_{ji}P_i + \mu_i^v, \tag{C.6}$$

and equivalently:

$$\mu_j^{v-v_{ij}} \leq v_{ij}P_j + \mu_j^v. \tag{C.7}$$

The inequalities (C.5c) and (C.5b) yield:

$$\mu_i^v - \mu_j^{v-v_{ij}} - \beta\tau_{ij} \leq 0,$$

and equivalently:

$$\mu_j^v - \mu_i^{v-v_{ji}} - \beta\tau_{ji} \leq 0, \tag{C.8}$$

Inequalities (C.6) and (C.8):

$$\mu_j^v \leq \mu_i^v + \beta\tau_{ji} + v_{ji}P_i. \tag{C.9}$$

And finally, the constraint (C.5c):

$$\begin{aligned}\nu_{ij} &\leq \beta\tau_{ij} + \mu_j^{v-v_{ij}} - \mu_i^v \\ &\overset{\text{(C.7)}}{\leq} \beta\tau_{ij} + v_{ij}P_j + \mu_j^v - \mu_i^v \\ &\overset{\text{(C.9)}}{\leq} \beta\tau_{ij} + v_{ij}P_j + \beta\tau_{ji} + v_{ji}P_i.\end{aligned}$$

Replacing $P_i = p_i + \beta$ and rearranging the terms:

$$\nu_{ij} \leq \beta(\tau_{ij} + \tau_{ji} + v_{ij} + v_{ji}) + v_{ij}p_j + v_{ji}p_i. \tag{C.10}$$

Using the upper bound on the dual variables ν_{ij} and (C.4), we can upper bound the optimal prices.

C.2 Proofs for Results in Section 4.3

C.2.1 Proof of Proposition 4.3.1

For brevity of notation, let $\beta_c + p_i = P_i$. Let λ_{ij} be the dual variables corresponding to the demand satisfaction constraints and μ_i^e be the dual variables corresponding to the flow balance constraints. We can state the dual problem as:

$$\min_{\lambda_{ij},\mu_i^e} \max_{\ell_{ij}^1} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij} D(\ell_{ij}^1, \infty)\left(\ell_{ij}^1 - \lambda_{ij}\right) \tag{C.11a}$$

$$\text{subject to} \quad \lambda_{ij} \geq 0, \tag{C.11b}$$

$$\lambda_{ij} + \mu_i^e - \mu_j^{e-e_{ij}} - \beta_t\tau_{ij} \leq 0, \tag{C.11c}$$

$$\mu_i^e - \mu_i^{e+1} - P_i \leq 0 \quad \forall i, j, e. \tag{C.11d}$$

For fixed λ_{ij} and μ_i^e, the first order optimality condition is:

$$\frac{\partial D(\ell_{ij}^1, \infty)}{\partial \ell_{ij}^1}(\ell_{ij}^1 - \lambda_{ij}) + D(\ell_{ij}^1, \infty) = 0 \tag{C.12}$$

Depending on the region ℓ_{ij}^1 is in, the demand function $D(\ell_{ij}^1, \infty)$ has different forms:

$$D(\ell_{ij}^1, \infty) = \begin{cases} 1 - \frac{(\ell_{ij}^1)^2}{2\ell_{\max}^2 \sigma(1-\sigma)} & , \frac{\ell_{ij}^1}{\ell_{\max}} < (1-\sigma) \\ \frac{1+\sigma-\frac{2\ell_{ij}^1}{\ell_{\max}}}{2\sigma} & , (1-\sigma) \leq \frac{\ell_{ij}^1}{\ell_{\max}} < \sigma \\ \frac{(1-\frac{\ell_{ij}^1}{\ell_{\max}})^2}{2\sigma(1-\sigma)} & , \sigma \leq \frac{\ell_{ij}^1}{\ell_{\max}} \leq 1 \end{cases} \tag{C.13}$$

First, suppose that $\frac{\ell_{ij}^1}{\ell_{\max}} < (1-\sigma)$. Solving for ℓ_{ij}^1 in (C.12) using (C.13), we get:

$$\ell_{ij}^m = \left(\lambda_{ij} + \sqrt{\lambda_{ij}^2 + 6\ell_{\max}^2 \sigma(1-\sigma)}\right)/3. \tag{C.14}$$

Furthermore, the second order optimality condition satisfies:

$$\frac{\partial^2 D(\ell_{ij}^1, \infty)}{\partial (\ell_{ij}^1)^2}(\ell_{ij}^m - \lambda_{ij}) + 2\frac{\partial D(\ell_{ij}^1, \infty)}{\partial \ell_{ij}^1}\bigg|_{\ell_{ij}^1 = \ell_{ij}^m} < 0. \tag{C.15}$$

Hence, KKT conditions are satisfied and the optimal primal solution satisfies the dual solution with optimal dual variables λ_{ij}^m. By checking the condition $\ell_{ij}^m \leq (1-\sigma)\ell_{\max}$ using (C.14), we get the condition that $\lambda_{ij}^m \leq \frac{3-5\sigma}{2}$. The optimal prices for the regions where $\frac{\ell_{ij}^m}{\ell_{\max}} \in [1-\sigma, \sigma)$ and $\frac{\ell_{ij}^m}{\ell_{\max}} \in [\sigma, 1]$ are derived in a similar fashion using the demand functions in those regions given in (C.13).

To get the upper bound on prices, we go through the following algebraic calculations using

the constraints. The inequality (C.11d) gives:

$$\mu_i^{e-e_{ji}} \leq e_{ji}P_i + \mu_i^e, \tag{C.16}$$

and equivalently:

$$\mu_j^{e-e_{ij}} \leq e_{ij}P_j + \mu_j^e. \tag{C.17}$$

The inequalities (C.11c) and (C.11b) yield:

$$\mu_i^e - \mu_j^{e-e_{ij}} - \beta_t\tau_{ij} \leq 0,$$

and equivalently:

$$\mu_j^e - \mu_i^{e-e_{ji}} - \beta_t\tau_{ji} \leq 0, \tag{C.18}$$

Inequalities (C.16) and (C.18):

$$\mu_j^e \leq \mu_i^e + \beta_t\tau_{ji} + e_{ji}P_i. \tag{C.19}$$

And finally, the constraint (C.11c):

$$\begin{aligned}\lambda_{ij} \leq \beta\tau_{ij} + \mu_j^{e-e_{ij}} - \mu_i^e &\overset{\text{(C.17)}}{\leq} \beta_t\tau_{ij} + e_{ij}P_j + \mu_j^e - \mu_i^e \\ &\overset{\text{(C.19)}}{\leq} \beta_t\tau_{ij} + e_{ij}P_j + \beta\tau_{ji} + e_{ji}P_i.\end{aligned}$$

Replacing $P_i = p_i + \beta_c$ and rearranging the terms:

$$\lambda_{ij} \leq \beta_t(\tau_{ij}+\tau_{ji}) + e_{ij}(p_j+\beta_c) + e_{ji}(p_i+\beta_c) = \overline{\lambda}_{ij}, \tag{C.20}$$

where the last equality follows from the definition provided in the proposition. Hence, we get

the desired upper bound on the prices using the upper bound on the dual variables.

C.2.2 Proof of Proposition 4.3.2

Using Assumption 4.3.2, we see that $\frac{(3-5\sigma)}{2} \leq 0$ and $\frac{(3-5\sigma)}{2}\ell_{\max} \leq \lambda_{ij}^m \leq \max_{i,j} \overline{\lambda}_{ij} \leq \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max} \leq \frac{3\sigma-1}{2}\ell_{\max}$. Hence, the optimal prices fall in the region $[(1-\sigma)\ell_{\max}, \sigma\ell_{\max})$, and are given by:

$$\ell_{ij}^m = ((1+\sigma)\ell_{\max} + \lambda_{ij}^m)/4. \tag{C.21}$$

The dual problem with optimal prices in (C.21) can be stated as:

$$\min_{\lambda_{ij},\mu_i^e} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\theta_{ij}}{4\sigma\ell_{\max}} \left(\frac{(1+\sigma)\ell_{\max} - 2\lambda_{ij}}{2}\right)^2 \tag{C.22a}$$

$$\text{subject to} \quad \text{(C.11b)} - \text{(C.11d)}. \tag{C.22b}$$

The objective function in (C.22a) with optimal dual variables, along with (C.21) suggests:

$$P^m = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\theta_{ij}}{4\sigma\ell_{\max}}((1+\sigma)\ell_{\max} - 2\ell_{ij}^m)^2,$$

where profits P^m is the optimal value of the objective function of both primal and dual problems (Since the demand function is linear in the specified region, the problem is convex and KKT conditions are satisfied. Hence, strong duality holds).

Consumer surplus is given by the difference between the price that customers pay and the price that they are willing to pay. For OD pair (i,j) the customers with $v_1 > \ell_{ij}^m$ have a positive surplus of $v_1 - \ell_{ij}^m$ and the customers with $v_1 \leq \ell_{ij}^m$ have a zero surplus since they either do not take the ride or have exactly a valuation of ℓ_{ij}^m. Since $v_1 = \sigma x + (1-\sigma)y$ and x and y are iid uniform random variables in $[0, \ell_{\max}]$, the consumer surplus for a single unit of potential riders

between OD pairs (i, j) is computed as:

$$\int_0^{\ell_{\max}} \int_{\frac{\ell_{ij}^m-(1-\sigma)y}{\sigma}}^{\ell_{\max}} \frac{1}{\ell_{\max}^2}(\sigma x + (1-\sigma)y - \ell_{ij}^m)dxdy = \frac{\ell_{\max}(\sigma^2+\sigma+1) - 3\ell_{ij}^m(1+\sigma-\frac{\ell_{ij}^m}{\ell_{\max}})}{6\sigma}. \tag{C.23}$$

The total consumer surplus is then:

$$\text{CS}^m \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij} \frac{\ell_{\max}(\sigma^2+\sigma+1) - 3\ell_{ij}^m(1+\sigma-\frac{\ell_{ij}^m}{\ell_{\max}})}{6\sigma}. \tag{C.24}$$

C.2.3 Proof of Proposition 4.3.3

In order to prove that the firms are in an equilibrium, we first follow similar steps as the proof of Proposition 4.3.1 and determine the optimal prices. Suppose that $\ell_{ij}^1,\ \ell_{ij}^2 \in \left[\frac{1-\sigma}{2}\ell_{\max}, (1-\sigma)\ell_{\max}\right]$. In that region:

$$D(\ell_{ij}^1, \ell_{ij}^2) = \frac{4\sigma(1-\sigma-\frac{\ell_{ij}^1-\ell_{ij}^2}{\ell_{\max}}) - (\frac{\ell_{ij}^1+\ell_{ij}^2}{\ell_{\max}}+\sigma-1)^2}{8\sigma(1-\sigma)} \tag{C.25}$$

$$\frac{\partial D(\ell_{ij}^1, \ell_{ij}^2)}{\partial \ell_{ij}^1} = \frac{1}{\ell_{\max}} \frac{-6\sigma - \frac{2\ell_{ij}^1}{\ell_{\max}} - \frac{2\ell_{ij}^2}{\ell_{\max}} + 2}{8\sigma(1-\sigma)} \tag{C.26}$$

$$\frac{\partial^2 D(\ell_{ij}^1, \ell_{ij}^2)}{\partial (\ell_{ij}^1)^2} = \frac{1}{\ell_{\max}^2} \frac{-2}{8\sigma(1-\sigma)} \tag{C.27}$$

Evaluated at $\ell_{ij}^1 = \ell_{ij}^2$, the above expressions become:

$$D(\ell_{ij}^1, \ell_{ij}^2)\Big|_{\ell_{ij}^1=\ell_{ij}^2} = \frac{1}{2} - \frac{(\frac{2\ell_{ij}^1}{\ell_{\max}}+\sigma-1)^2}{8\sigma(1-\sigma)} \tag{C.28}$$

$$\frac{\partial D(\ell_{ij}^1, \ell_{ij}^2)}{\partial \ell_{ij}^1}\Bigg|_{\ell_{ij}^1=\ell_{ij}^2} = \frac{1}{\ell_{\max}} \frac{-6\sigma - \frac{4\ell_{ij}^1}{\ell_{\max}} + 2}{8\sigma(1-\sigma)} \tag{C.29}$$

For a given λ_{ij}, the first order optimality condition is:

$$\frac{\partial D(\ell_{ij}^1, \ell_{ij}^2)}{\partial \ell_{ij}^1}\bigg|_{\ell_{ij}^1=\ell_{ij}^2} (\ell_{ij}^1 - \lambda_{ij}) + D(\ell_{ij}^1, \ell_{ij}^2)\Big|_{\ell_{ij}^1=\ell_{ij}^2} = 0. \tag{C.30}$$

We plug equations (C.28) and (C.29) into the above expression to get a quadratic equation in ℓ_{ij}^1, which has two solutions. One of the solutions is infeasible with $\ell_{ij}^1 < 0$. Hence, we get a unique solution at:

$$\ell_{ij}^d = \frac{(3-5\sigma)\ell_{\max} + 2\lambda_{ij} + \sqrt{\Delta_1}}{8}, \tag{C.31}$$

where $\Delta_1 = 4\ell_{\max}^2 + (2\lambda_{ij} + (15\sigma - 3)\ell_{\max})(2\lambda_{ij} + (1-\sigma)\ell_{\max})$. Note that in the region where $\ell_{ij}^1 = \ell_{ij}^2 \leq (1-\sigma)\ell_{\max}$, $D(\ell_{ij}^1, \ell_{ij}^2)$ is concave and thus we need to check the second order optimality condition:

$$\frac{\partial^2 D(\ell_{ij}^1, \ell_{ij}^2)}{\partial (\ell_{ij}^1)^2}(\ell_{ij}^d - \lambda_{ij}) + 2\frac{\partial D(\ell_{ij}^1, \ell_{ij}^2)}{\partial \ell_{ij}^1}\bigg|_{\ell_{ij}^1=\ell_{ij}^2=\ell_{ij}^d} < 0. \tag{C.32}$$

By plugging Equations (C.27), (C.29), and (C.31) into the above expression, one verifies that it holds true. Hence, KKT conditions are satisfied and the optimal primal solution satisfies the dual solution with optimal dual variables λ_{ij}^d:

$$\ell_{ij}^d = \frac{(3-5\sigma)\ell_{\max} + 2\lambda_{ij}^d + \sqrt{\Delta_1^*}}{8}, \tag{C.33}$$

where $\Delta_1^* = 4\ell_{\max}^2 + (2\lambda_{ij}^d + (15\sigma - 3)\ell_{\max})(2\lambda_{ij}^d + (1-\sigma)\ell_{\max})$. Since the conjecture was that $\ell_{ij}^d \in \left[\frac{1-\sigma}{2}\ell_{\max}, (1-\sigma)\ell_{\max}\right]$, we check:

$$\frac{1-\sigma}{2}\ell_{\max} \leq \frac{(3-5\sigma)\ell_{\max} + 2\lambda_{ij}^d + \sqrt{\Delta_1^*}}{8} \leq (1-\sigma)\ell_{\max},$$

to get $\lambda_{ij}^d \leq \frac{3(1-\sigma)^2}{2(1+\sigma)}$. For $\lambda_{ij}^d = 0$, (C.33) evaluates to $\frac{(3-5\sigma)+\sqrt{-15\sigma^2+18\sigma+1}}{8}\ell_{\max} \geq \frac{1-\sigma}{2}\ell_{\max}$, hence the prices fall in the specified region.

Now suppose that $\ell_{ij}^1, \ell_{ij}^2 \in ((1-\sigma)\ell_{\max}, \frac{\sigma+1}{2}\ell_{\max}]$. In that region:

$$D(\ell_{ij}^1, \ell_{ij}^2) = \frac{(1-\sigma+\frac{\ell_{ij}^2-\ell_{ij}^1}{\ell_{\max}})(3+\sigma-\frac{3\ell_{ij}^1+\ell_{ij}^2}{\ell_{\max}})}{8\sigma(1-\sigma)} \tag{C.34}$$

By following the same steps as before, we get optimal prices uniquely as:

$$\ell_{ij}^d = \frac{(5-3\sigma)\ell_{\max} + 2\lambda_{ij}^d - \sqrt{\Delta_2^*}}{4}, \tag{C.35}$$

where $\Delta_2^* = 2(\sigma\ell_{\max} - \lambda_{ij}^d)^2 + 2(\ell_{\max} - \lambda_{ij}^d)^2 + 11(\sigma-1)^2\ell_{\max}^2$. By imposing the condition that $\ell_{ij}^d \in ((1-\sigma)\ell_{\max}, \frac{\sigma+1}{2}\ell_{\max}]$, one identifies:

$$\frac{3(1-\sigma)^2}{2(1+\sigma)}\ell_{\max} < \lambda_{ij}^d \leq \frac{3\sigma+1}{4}\ell_{\max}. \tag{C.36}$$

The upper bound on the dual variables is derived identically to the Proposition 4.3.1. Hence according to Assumption 4.3.2

$$\lambda_{ij}^d \leq \overline{\lambda}_{ij} \leq \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max} < \frac{3\sigma+1}{2}\ell_{\max},$$

is satisfied.

All in all, we get the optimal prices as:

$$\ell_{ij}^d = \begin{cases} \frac{(3-5\sigma)\ell_{\max}+2\lambda_{ij}^d+\sqrt{\Delta_1^*}}{8} & \frac{\lambda_{ij}^d}{\ell_{\max}} \leq \frac{3(\sigma-1)^2}{2(\sigma+1)} \\ \frac{(5-3\sigma)\ell_{\max}+2\lambda_{ij}^d-\sqrt{\Delta_2^*}}{4} & o.w. \end{cases} \tag{C.37}$$

We now show that when both firms set prices equal to $\{\ell_{ij}^d\}_{i,j\in\mathcal{N}}$, they are in an equilibrium.

Given firm $-f$'s prices equal to $\{\ell_{ij}^d\}_{i,j\in\mathcal{N}}$, firm f solves the following best response problem to determine its optimal prices:

$$\max_{x_{ic}^e, x_{ij}^e, \ell_{ij}^f} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij}\ell_{ij}^f D(\ell_{ij}^f, \ell_{ij}^d) - \sum_{i=1}^{n}\sum_{e=0}^{e_{\max}-1} (\beta_c + p_i)x_{ic}^e - \beta_t \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e \tau_{ij} \tag{C.38a}$$

$$\text{subject to} \quad \theta_{ij} D(\ell_{ij}^f, \ell_{ij}^d) \leq \sum_{e=e_{ij}}^{e_{\max}} x_{ij}^e \quad \forall i, j \in \mathcal{N}, \tag{C.38b}$$

$$(4.16c) - (4.16g).$$

The first order optimality condition states:

$$\frac{\partial D(\ell_{ij}^f, \ell_{ij}^d)}{\partial \ell_{ij}^f}(\ell_{ij}^f - \lambda_{ij}) + D(\ell_{ij}^f, \ell_{ij}^d) = 0. \tag{C.39}$$

Setting $\ell_{ij}^f = \ell_{ij}^d$ satisfies the above equation with the optimal dual variable λ_{ij}^d because $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d$ is a solution to (C.30) with $\lambda_{ij} = \lambda_{ij}^d$. Since both firms have the identical best response problem (C.38), the first order condition is satisfied for both when $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d, \forall i, j \in \mathcal{N}$, and hence the firms are in an equilibrium.

C.2.4 Proof Of Proposition 4.3.4

We show that when $\ell_{ij}^1 \neq \ell_{ij}^2$ and both firms serve greater than zero demand for an OD pair (i, j), the firms cannot be in equilibrium. We do it by showing that the first order condition can not hold for both firms simultaneously.

We let $\ell_{ij}^1 = \ell_{ij}^2 + \delta\ell_{\max}$, and add the following constraints:

- We constrain $\delta < (1 - \sigma)$ (If $\delta \geq (1 - \sigma)$, then firm 1 does not serve any demand for

that OD pair since the lines depicted on Figure 4.10a intersect at $y \geq \ell_{\max}$).

- We let $\ell_{ij}^1 + \ell_{ij}^2 \geq (1-\sigma)\ell_{\max}$ (Otherwise if $\ell_{ij}^1 + \ell_{ij}^2 = (1-\sigma)\ell_{\max} - 2\epsilon$ lines depicted in Figure 4.10a intersect at $x = \frac{-2\epsilon}{2\sigma}$. Then both firms can increase their profits by increasing their prices by ϵ, while keeping the demand same).

- We let $\ell_{ij}^1 + \ell_{ij}^2 \leq (1+\sigma)\ell_{\max}$ (Otherwise, the lines depicted in Figure 4.10a intersect at $x \geq \ell_{\max}$, and hence their prices don't affect each others' demand. In that case, the prices are determined according to the monopoly prices, which are bounded by $\sigma\ell_{\max}$ according to Assumption 4.3.2 and hence their sum is always bounded by $(1+\sigma)\ell_{\max}$).

Depending on whether ℓ_{ij}^1 and ℓ_{ij}^2 are greater than $(1-\sigma)\ell_{\max}$, we have different demand functions and hence we study the following three cases:

Case 1: Let $\ell_{ij}^1, \ell_{ij}^2 \leq (1-\sigma)\ell_{\max}$. For ease of notation, we define $\ell_f := \frac{\ell_{ij}^f}{\ell_{\max}}$ for firm f. When $\ell_1, \ell_2 \leq 1-\sigma$, the demand function for firm f is given by:

$$D(\ell_f, \ell_{-f}) = \frac{4\sigma(1-\sigma-(\ell_f - \ell_{-f})) - (\ell_f + \ell_{-f} + \sigma - 1)^2}{8\sigma(1-\sigma)} \tag{C.40}$$

$$\frac{\partial D(\ell_f, \ell_{-f})}{\partial \ell_f} = \frac{-6\sigma - 2\ell_f - 2\ell_{-f} + 2}{8\sigma(1-\sigma)} \tag{C.41}$$

Using (C.40) and $\ell_1 - \ell_2 = \delta$, the demand functions are determined as:

$$D(\ell_1, \ell_2) = \frac{4\sigma(1-\sigma-\delta) - (\sigma + 2\ell_1 - 1 - \delta)^2}{8\sigma(1-\sigma)}, \tag{C.42}$$

$$D(\ell_2, \ell_1) = \frac{4\sigma(1-\sigma+\delta) - (\sigma + 2\ell_1 - 1 - \delta)^2}{8\sigma(1-\sigma)}. \tag{C.43}$$

Furthermore, using (C.41), the derivatives of the demand functions are determined as:

$$\frac{\partial D(\ell_1, \ell_2)}{\partial \ell_1} = \frac{\partial D(\ell_2, \ell_1)}{\partial \ell_2} = \frac{-6\sigma - 4\ell_1 + 2 + 2\delta}{8\sigma(1-\sigma)}. \tag{C.44}$$

In an equilibrium, both firms should satisfy the first order condition (FOC). We show that the FOC can not hold for both of the firms. Define $\lambda_f = \frac{\lambda_{ij}^f}{\ell_{\max}}$ for firm f and let FOC for firm 2 hold:

$$\frac{\partial D(\ell_2, \ell_1)}{\partial \ell_2}(\ell_2 - \lambda_2) + D(\ell_2, \ell_1) = 0. \tag{C.45}$$

Using (C.42), (C.43), and (C.44), we can rewrite the above equation as:

$$\begin{aligned}&\frac{\partial D(\ell_1, \ell_2)}{\partial \ell_1}(\ell_2 - \ell_1 + \ell_1 - \lambda_1 + \lambda_1 - \lambda_2) + D(\ell_1, \ell_2) - \frac{4\sigma(1-\sigma-\delta)}{8\sigma(1-\sigma)} + \frac{4\sigma(1-\sigma+\delta)}{8\sigma(1-\sigma)} \\ &= \frac{\partial D(\ell_1, \ell_2)}{\partial \ell_1}(\ell_1 - \lambda_1) + D(\ell_1, \ell_2) + \frac{\delta}{1-\sigma} + \frac{\partial D(\ell_1, \ell_2)}{\partial \ell_1}(-\delta - (\lambda_2 - \lambda_1)) = 0.\end{aligned} \tag{C.46}$$

To proceed, we use the following lemma:

Lemma C.2.1 *Let $|\delta| \leq 1 - \sigma$, $\ell_1 - \ell_2 = \delta$, and $\ell_1, \ell_2 \leq 1 - \sigma$. If the prices satisfy the FOC, then the following inequality holds:*

$$|\lambda_1 - \lambda_2| \leq (2 - \sigma(3\sigma - 2))|\delta| \tag{C.47}$$

Proof: Using (C.43) and (C.44), one can state the FOC for a given price $\ell_{-f} = \ell_f + \delta$ to get a quadratic equation in ℓ_f. This equation has two solutions, one of which is infeasible. Hence, we get the optimal price ℓ_f as:

$$\ell_f = \frac{3 - 5\sigma + 2\lambda_f - 3\delta + \sqrt{\Delta}}{8}, \tag{C.48}$$

where

$$\Delta = (\delta + 2\lambda_f + 7\sigma - 1)^2 + 32\sigma(\delta - 2\sigma + 1).$$

We compute the change in the optimal price with respect to the dual variable as:

$$\frac{\partial \ell_f}{\partial \lambda_f} = \frac{1}{4}\left(1 + \frac{\delta + 2\lambda_f + 7\sigma - 1}{\sqrt{\Delta}}\right) \tag{C.49}$$

The goal is to lower bound $\frac{\partial \ell_f}{\partial \lambda_f}$. In order to do so, we study how $\frac{\partial \ell_f}{\partial \lambda_f}$ changes with λ_f and δ. We first observe that

$$\frac{\partial^2 \ell_f}{\partial \lambda_f^2} < 0,$$

hence the we need to maximize λ_f in order to minimize $\frac{\partial \ell_f}{\partial \lambda_f}$. Since ℓ_f is constrained to be less than $1 - \sigma$, by upper bounding the expression in (C.48) we get a bound on λ_f as:

$$\lambda_f \leq \frac{\delta^2 + \delta(4 - 8\sigma) + 3(\sigma - 1)^2}{2(\delta + \sigma + 1)} \tag{C.50}$$

Next, we plug the upper bound on λ_f to (C.49) to get an expression that is only dependent on σ and δ and compute the partial derivative with respect to δ to get:

$$\frac{\partial}{\partial \delta}\frac{\partial \ell_f}{\partial \lambda_f} < 0,$$

hence we maximize δ in order to minimize $\frac{\partial \ell_f}{\partial \lambda_f}$. We set $\delta = 1 - \sigma$ to get:

$$\frac{\partial \ell_f}{\partial \lambda_f} \geq \frac{1}{2 - \sigma(3\sigma - 2)}. \tag{C.51}$$

Above inequality hods for all $\ell_f \leq 1 - \sigma$. Since $\ell_1, \ell_2 \leq 1 - \sigma$, this means:

$$\left|\frac{\ell_1 - \ell_2}{\lambda_1 - \lambda_2}\right| \geq \frac{\ell_1 - \ell_2}{\lambda_1 - \lambda_2} \geq \frac{1}{2 - \sigma(3\sigma - 2)}. \tag{C.52}$$

Plugging $\ell_1 - \ell_2 = \delta$ concludes the proof. ■

Going back to (C.46), we rearrange:

$$\begin{aligned}
&\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(\ell_1-\lambda_1)+D(\ell_1,\ell_2) = -\frac{\delta}{1-\sigma}-\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(-\delta-(\lambda_2-\lambda_1))\\
&\overset{\text{Lemma C.2.1}}{\leq} -\frac{\delta}{1-\sigma}-\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}\delta(-1-(\sigma(3\sigma-2)-2))\\
&= -\frac{\delta}{1-\sigma}-\delta(1-\sigma(3\sigma-2))\frac{-6\sigma-4\ell_1+2+2\delta}{8\sigma(1-\sigma)}\\
&= \frac{\delta}{8\sigma(1-\sigma)}(-8\sigma+(1-\sigma(3\sigma-2))(6\sigma+4\ell_1-2-2\delta))\\
&\overset{\ell_1\leq 1-\sigma}{\leq} \frac{\delta}{8\sigma(1-\sigma)}(-8\sigma+(1-\sigma(3\sigma-2))(2+2\sigma-2\delta))\\
&\leq \frac{\delta}{8\sigma(1-\sigma)}(-8\sigma+(1-\sigma(3\sigma-2))(2+2\sigma))\\
&= -\frac{\delta}{4\sigma(1-\sigma)}(3\sigma^3+\sigma^2+\sigma-1)<0, \quad \forall\sigma\in[1/2,1].
\end{aligned} \tag{C.53}$$

We conclude that FOC for firm 1 does not hold, hence they can not be in an equilibrium.

Case 2: Let $\ell_1,\ell_2 \geq (1-\sigma)$. In this region, the demand function and its derivative for firm f can be stated as:

$$D(\ell_f,\ell_{-f}) = \frac{(1-\sigma+(\ell_{-f}-\ell_f))(3+\sigma-(3\ell_f+\ell_{-f}))}{8\sigma(1-\sigma)} \tag{C.54}$$

$$\frac{\partial D(\ell_f,\ell_{-f})}{\partial \ell_f} = \frac{-6+2\sigma+6\ell_f-2\ell_{-f}}{8\sigma(1-\sigma)} \tag{C.55}$$

Using the above equations and $\ell_1-\ell_2=\delta$, we write:

$$D(\ell_1,\ell_2) = \frac{(1-\sigma-\delta)(3+\sigma-4\ell_2-3\delta)}{8\sigma(1-\sigma)}, \tag{C.56}$$

$$D(\ell_2,\ell_1) = \frac{(1-\sigma+\delta)(3+\sigma-4\ell_2-\delta)}{8\sigma(1-\sigma)}, \tag{C.57}$$

$$\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1} = \frac{-6+2\sigma+4\ell_2+6\delta}{8\sigma(1-\sigma)}, \tag{C.58}$$

$$\frac{\partial D(\ell_2,\ell_1)}{\partial \ell_2} = \frac{-6+2\sigma+4\ell_2-2\delta}{8\sigma(1-\sigma)}. \tag{C.59}$$

We follow similar steps as in Case 1 to show that FOC for both firms can not hold. We state the FOC for firm 2:

$$\begin{aligned}
&\frac{\partial D(\ell_2,\ell_1)}{\partial \ell_2}(\ell_2-\lambda_2) + D(\ell_2,\ell_1) \overset{\text{(C.58),(C.59)}}{=} \left(\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1} - \frac{8\delta}{8\sigma(1-\sigma)}\right)(\ell_2-\lambda_2) + D(\ell_2,\ell_1) \\
&\overset{\text{(C.56),(C.57)}}{=} \left(\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1} - \frac{8\delta}{8\sigma(1-\sigma)}\right)(\ell_2-\lambda_2) + D(\ell_1,\ell_2) + \frac{\delta(8-4\delta-8\ell_2)}{8\sigma(1-\sigma)} \\
&= \frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(\ell_2-\ell_1+\ell_1-\lambda_1+\lambda_1-\lambda_2) \\
&\quad - \frac{8\delta}{8\sigma(1-\sigma)}(\ell_2-\lambda_2) + D(\ell_1,\ell_2) + \frac{\delta(8-4\delta-8\ell_2)}{8\sigma(1-\sigma)} \\
&= \frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(\ell_1-\lambda_1) + D(\ell_1,\ell_2) - \frac{8\delta}{8\sigma(1-\sigma)}(\ell_2-\lambda_2) \\
&\quad + \frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(-\delta+\lambda_1-\lambda_2) + \frac{\delta(8-4\delta-8\ell_2)}{8\sigma(1-\sigma)} \\
&= 0
\end{aligned} \tag{C.60}$$

For the FOC of firm 1 to hold, the following expression has to be equal to 0:

$$\frac{8\delta}{8\sigma(1-\sigma)}(\ell_2-\lambda_2) - \frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(-\delta+\lambda_1-\lambda_2) - \frac{\delta(8-4\delta-8\ell_2)}{8\sigma(1-\sigma)} \tag{C.61}$$

We show that the above expression is always less than zero by upper bounding it. To proceed, we use the following lemma:

Lemma C.2.2 *Let $|\delta| \le 1-\sigma$, $\ell_1-\ell_2=\delta$, and $\ell_1,\ell_2 \ge 1-\sigma$. If the prices satisfy the FOC, then the following inequality holds:*

$$|\lambda_1-\lambda_2| \le \frac{2|\delta|}{1-\frac{2\bar{\lambda}-2\sigma}{\sqrt{48(1-\sigma)^2+(2\bar{\lambda}-2\sigma)^2}}}, \tag{C.62}$$

where

$$\overline{\lambda} = \frac{(3-\sigma)(3\sigma-1)}{4(5-3\sigma)}.$$

Proof: Using (C.57) and (C.59), one can state the FOC for a given price $\ell_{-f} = \ell_f + \delta$ to get a quadratic equation in ℓ_f. This equation has two solutions, one of which is infeasible. Hence, we get the optimal price ℓ_f as:

$$\ell_f = \frac{5 - 3\sigma + 3\delta + 2\lambda_f - \sqrt{\Delta}}{4} \tag{C.63}$$

where

$$\Delta = (\delta - \sigma - 1 + 2\lambda_f)^2 + 12(\sigma-1)^2 + 12\delta(\delta + 2 - 2\sigma).$$

Similar to Lemma C.2.1, the goal is to lower bound $\frac{\partial \ell_f}{\partial \lambda_f}$. It is computed as:

$$\frac{\partial \ell_f}{\partial \lambda_f} = \frac{1}{4}\left(2 - \frac{2(\delta + \lambda_f - \sigma - 1)}{\sqrt{\Delta}}\right) \tag{C.64}$$

In order to minimize the RHS of the above expression, we study how it depends on the variables it is a function of. We first identify that

$$\frac{\partial^2 \ell_f}{\partial \lambda_f^2} < 0, \quad \frac{\partial}{\partial \delta}\frac{\partial \ell_f}{\partial \lambda_f} < 0, \tag{C.65}$$

and hence $\frac{\partial \ell_f}{\partial \lambda_f}$ is minimized when $\delta = 1 - \sigma$ and $\lambda_f = \frac{(3-\sigma)(3\sigma-1)}{4(5-3\sigma)}$. We plug these expressions to $\frac{\partial \ell_f}{\partial \lambda_f}$ to get:

$$\frac{\partial \ell_f}{\partial \lambda_f} \geq \frac{1}{2}\left(1 - \frac{2\overline{\lambda} - 2\sigma}{\sqrt{48(1-\sigma)^2 + (2\overline{\lambda} - 2\sigma)^2}}\right), \tag{C.66}$$

where $\overline{\lambda} := \frac{(3-\sigma)(3\sigma-1)}{4(5-3\sigma)}$.

The above inequality holds for all $\ell_f \geq 1-\sigma$ as long as the FOC holds, hence this means:

$$\Big|\frac{\ell_1-\ell_2}{\lambda_1-\lambda_2}\Big| \geq \frac{\ell_1-\ell_2}{\lambda_1-\lambda_2} \geq \frac{1-\frac{2\bar{\lambda}-2\sigma}{\sqrt{48(1-\sigma)^2+(2\bar{\lambda}-2\sigma)^2}}}{2}. \tag{C.67}$$

Plugging $\ell_1-\ell_2=\delta$ concludes the proof. ■

The upper bound (C.62) in Lemma C.2.2 is concave in σ and is decreasing with σ in the interval $[0,1]$. For brevity of exposition, we therefore use a linear upper bound in σ. It can be shown that

$$\frac{2}{1-\frac{2\bar{\lambda}-2\sigma}{\sqrt{48(1-\sigma)^2+(2\bar{\lambda}-2\sigma)^2}}} \leq \frac{9-5\sigma}{4}, \quad \forall \sigma \in [0,1]. \tag{C.68}$$

Using (C.68), we upper bound (C.61):

$$\begin{aligned}&\frac{8\delta}{8\sigma(1-\sigma)}(\ell_2-\lambda_2)-\frac{\partial D(\ell_1,\ell_2)}{\partial \ell_1}(-\delta+\lambda_1-\lambda_2)-\frac{\delta(8-4\delta-8\ell_2)}{8\sigma(1-\sigma)}\\ &<\frac{\delta}{8\sigma(1-\sigma)}\Big(8(\ell_2-\lambda_2)-(8-4\delta-8\ell_2)-\frac{5-5\sigma}{4}(-6+2\sigma+4\ell_2+6\delta)\Big)\end{aligned} \tag{C.69}$$

Given λ_2 and δ, ℓ_2 is uniquely determined as in (C.63). We plug ℓ_2 to (C.69), and conjecture that (C.69) <0. That gives:

$$\delta(45\sigma+19)+(1-\sigma)(5\sigma-10\lambda_2+53)<(5\sigma+11)\sqrt{\Delta} \tag{C.70}$$

We take the square of both sides, collect terms on LHS, and re-state the conjecture as:

$$f(\sigma,\lambda_2,\delta)<0, \tag{C.71}$$

where

$$\begin{aligned} f(\sigma, \lambda_2, \delta) =& \delta^2(1700\sigma^2 + 280\sigma - 1212) \\ &+ 8\delta(25\sigma^2 - 275\sigma^2 + 459\sigma - 81) \\ &- 300\sigma^4 - 400\sigma^3 + 2296\sigma^2 - 5856\sigma \\ &- 128(5\sigma + 3)\lambda_2^2 + 32\lambda_2\delta(25\sigma^2 - 30\sigma - 27) \\ &- 64\lambda_2(5\sigma^2 - 46\sigma + 9) + 1236 \end{aligned} \tag{C.72}$$

Our goal is to maximize $f(\sigma, \lambda_2, \delta)$ and show that it is less than 0. We first identify that $\frac{\partial f(\sigma,\lambda_2,\delta)}{\partial \lambda_2} > 0$, hence $f(\sigma, \lambda_2, \delta)$ is maximized when λ_2 is maximized. We evaluate $f(\sigma, \lambda_2, \delta)$ at $\lambda_2 = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}$. We next identify that $\frac{\partial f(\sigma,,\delta)}{\partial \delta} > 0$, hence set $\delta = 1 - \sigma$ to get:

$$\begin{aligned} &f(\sigma, \lambda_2, \delta) \leq f(\sigma, \frac{(3\sigma - 1)(3 - \sigma)}{4(5 - 3\sigma)}, 1 - \sigma) \\ =& \frac{8(3\sigma-1)(5\sigma-7)(-186 + 507\sigma - 209\sigma^2 - 155\sigma^2 + 75\sigma^4)}{(5 - 3\sigma)^2}. \end{aligned} \tag{C.73}$$

The above equation has roots at $\sigma \approx -1.8406$, $\sigma = 1/3$, $\sigma \approx 0.49744$, $\sigma \approx 1.2599$, and $\sigma = 7/5$ and therefore is less than 0 for $\sigma \in [1/2, 1]$. Hence, we conclude that $f(\sigma, \lambda_2, \delta) < 0$ and the conjecture was true. Going back, this means that the final expression in (C.69) is less than zero, which means the expression in (C.61) is less than zero, meaning that the FOC of firm 1:

$$\frac{\partial D(\ell_1, \ell_2)}{\partial \ell_1}(\ell_1 - \lambda_1) + D(\ell_1, \ell_2) < 0. \tag{C.74}$$

Hence, the FOC for firm 1 can not hold, meaning the firms can not be in an equilibrium.

Case 3: Let $\ell_2 \leq 1 - \sigma$, $\ell_1 \geq 1 - \sigma$. We show by contradiction that the FOC-satisfying prices can not be δ apart. We know that if the prices are in equilibrium, FOC holds for both.

The optimal prices are given by (C.48) and (C.63) (replacing δ by $-\delta$) as:

$$\ell_1 = \frac{5 - 3\sigma - 3\delta + 2\lambda_1 - \sqrt{\Delta_1}}{4}, \tag{C.75}$$

$$\ell_2 = \frac{3 - 5\sigma + 2\lambda_2 - 3\delta + \sqrt{\Delta_2}}{8}, \tag{C.76}$$

where

$$\Delta_1 = (-\delta - \sigma - 1 + 2\lambda_1)^2 + 12(\sigma - 1)^2 + 12\delta(\delta - 2 + 2\sigma),$$

and

$$\Delta_2 = (\delta + 2\lambda_2 + 7\sigma - 1)^2 + 32\sigma(\delta - 2\sigma + 1).$$

But since $\ell_1 = \ell_2 + \delta$, the following must be true:

$$\frac{3 - 5\sigma + 2\lambda_2 + 5\delta + \sqrt{\Delta_2}}{8} = \frac{5 - 3\sigma - 3\delta + 2\lambda_1 - \sqrt{\Delta_1}}{4}. \tag{C.77}$$

We show that the above equality can not hold by upper bounding the following function, which is the difference between the LHS and the RHS of the above equality:

$$g(\lambda_1, \lambda_2, \delta, \sigma) = \frac{-7 + \sigma + 11\delta + 2\lambda_2 - 4\lambda_1 + \sqrt{\Delta_2} - 2\sqrt{\Delta_1}}{8}. \tag{C.78}$$

In order to upper bound the above function, we use the following lemma:

Lemma C.2.3 *Let $|\delta| \leq 1-\sigma$ and $\ell_1 - \ell_2 = \delta$. If the prices satisfy the FOC, then the following inequality holds:*

$$|\lambda_1 - \lambda_2| \geq |\delta| \tag{C.79}$$

Proof: Given $\ell_{-f} = \ell_f + \delta$, the optimal prices for firm f are given by equations (C.48) and (C.63), for $\ell_f \leq 1-\sigma$ and $\ell_f > 1-\sigma$, respectively. Our goal is to upper bound $\frac{\partial \ell_f}{\partial \lambda_f}$, so that we can lower bound the difference between the dual variables, given that the price difference

is δ. In proofs of Lemmas C.2.1 and C.2.2, we have shown that

$$\frac{\partial^2 \ell_f}{\partial \lambda_f^2} < 0, \quad \frac{\partial}{\partial \delta}\frac{\partial \ell_f}{\partial \lambda_f} < 0,$$

and hence in order to upper bound $\frac{\partial \ell_f}{\partial \lambda_f}$, we set $\lambda_f = 0$ and $\delta = 0$ in equations (C.49) and (C.64). That gives:

$$\frac{\partial \ell_f}{\partial \lambda_f} \leq \frac{1}{4}\left(1 + \frac{7\sigma - 1}{\sqrt{-18\sigma^2 + 15\sigma + 1}}\right), \quad \ell_f \leq 1 - \sigma, \tag{C.80}$$

$$\frac{\partial \ell_f}{\partial \lambda_f} \leq \frac{1}{4}\left(2 + \frac{2(\sigma + 1)}{\sqrt{13\sigma^2 - 22\sigma + 13}}\right), \quad \ell_f > 1 - \sigma \tag{C.81}$$

Both of the above equations are increasing with σ, and are equal to 1 when $\sigma = 1$. Hence:

$$\frac{\partial \ell_f}{\partial \lambda_f} \leq 1, \tag{C.82}$$

or equivalently

$$\frac{\partial \lambda_f}{\partial \ell_f} \geq 1. \tag{C.83}$$

Since this holds for all ℓ_f:

$$\left|\frac{\lambda_1 - \lambda_2}{\ell_1 - \ell_2}\right| \geq \frac{\lambda_1 - \lambda_2}{\ell_1 - \ell_2} \geq 1. \tag{C.84}$$

Setting $\ell_1 - \ell_2 = \delta$ concludes the proof. ■

Noting that $\frac{\partial g(\lambda_1, \lambda_2, \delta, \sigma)}{\partial \lambda_2} \geq 0$ and using Lemma C.2.3 with $\lambda_2 \leq \lambda_1 - \delta$, we upper bound (C.78) as:

$$g(\lambda_1, \lambda_2, \sigma, \delta) \leq \hat{g}(\lambda_1, \sigma, \delta) = \frac{-7 + \sigma + 9\delta - 2\lambda_1 + \sqrt{\hat{\Delta}_1} - 2\sqrt{\Delta_2}}{8}, \tag{C.85}$$

where

$$\hat{\Delta}_1 = (2\lambda_1 - \delta + 7\sigma - 1)^2 + 32\sigma(\delta - 2\sigma + 1).$$

Our goal is to maximize $\hat{g}$ over its variables, and show that it is always less than 0. That would mean that when the prices are determined by the FOC, the difference between $\ell_2 + \delta$ and ℓ_1 is always less than zero, which contradicts with $\ell_1 - \ell_2 = \delta$. We compute the partial derivatives of $\hat{g}$ with respect to δ and λ_1 to get:

$$\frac{\partial \hat{g}(\lambda_1, \sigma, \delta)}{\partial \lambda_1} > 0, \quad \frac{\partial \hat{g}(\lambda_1, \sigma, \delta)}{\partial \delta} > 0, \tag{C.86}$$

and hence $\hat{g}(\lambda_1, \sigma, \delta)$ is maximized when $\delta = 1 - \sigma$ and $\lambda_1 = \frac{1}{2}$ (Note that $\lambda_1 \leq \frac{(3-\sigma)(3\sigma-1)}{4(5-3\sigma)} \leq \frac{1}{2}$):

$$\hat{g}(\lambda_1, \sigma, \delta) \leq \hat{g}(1/2, \sigma, 1-\sigma) = \frac{-8\sigma - 1 + \sqrt{-32\sigma^2 + 48\sigma + 1}}{8}. \tag{C.87}$$

Finally, we observe that $\frac{\partial \hat{g}(1/2, \sigma, 1-\sigma)}{\partial \sigma} < 0$ for $\sigma \in [1/2, 1]$, and hence $\hat{g}$ is maximized when $\sigma = 1/2$. Evaluated at $\sigma = 1/2$:

$$\hat{g}(1/2, 1/2, 1/2) = \frac{1}{8}(\sqrt{17} - 5) < 0. \tag{C.88}$$

Hence, with the FOC satisfying prices, $\ell_2 + \delta$ is always less than ℓ_1, which is a contradiction. This means that FOC can not hold for both firms and thus they can not be in an equilibrium.

We have shown that given $\delta > 0$, the FOC can not hold for both of the firms in none of the cases. Hence, the only case when FOC holds for both of the firms is when $\delta = 0$, i.e., $\ell_1 = \ell_2$. Therefore asymmetric equilibria can not exist.

C.2.5 Proof of Proposition 4.3.5

The symmetric duopoly equilibrium prices are determined in the proof of Proposition 4.3.3 (in Appendix C.2.3) as:

$$\ell_{ij}^d = \begin{cases} \frac{(3-5\sigma)\ell_{\max}+2\lambda_{ij}^d+\sqrt{\Delta_1^*}}{8} & \frac{\lambda_{ij}^d}{\ell_{\max}} \leq \frac{3(\sigma-1)^2}{2(\sigma+1)} \\ \frac{(5-3\sigma)\ell_{\max}+2\lambda_{ij}^d-\sqrt{\Delta_2^*}}{4} & o.w. \end{cases}, \tag{C.89}$$

where

$$\Delta_1^* = 4\ell_{\max}^2 + (2\lambda_{ij}^d + (15\sigma-3)\ell_{\max})(2\lambda_{ij}^d + (1-\sigma)\ell_{\max})$$

and

$$\Delta_2^* = 2(\sigma\ell_{\max} - \lambda_{ij}^d)^2 + 2(\ell_{\max} - \lambda_{ij}^d)^2 + 11(\sigma-1)^2\ell_{\max}^2.$$

Both equations in (C.89) are decreasing functions of λ_{ij}^d. In order to lower bound the difference between the monopoly prices and the duopoly prices, we lower bound the monopoly prices by setting $\lambda_{ij}^m = 0$ and upper bound the duopoly prices by setting $\lambda_{ij}^d = \overline{\lambda}_{ij}$. In order to upper bound the difference, we upper bound the monopoly prices by setting $\lambda_{ij}^m = \overline{\lambda}_{ij}$ and lower bound the duopoly prices by setting $\lambda_{ij}^d = 0$.

C.2.6 Proof of Proposition 4.3.6

Using $\lambda_{ij}^m \leq \overline{\lambda}_{ij} \leq \max_{i,j} \overline{\lambda}_{ij} \leq \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ and $\lambda_{ij}^m \geq 0$, (4.17) and (4.18) give:

$$\frac{(1+\sigma)\ell_{\max}}{4} \leq \ell_{ij}^m \leq \frac{7+14\sigma-9\sigma^2}{40-24\sigma}\ell_{\max}. \tag{C.90}$$

Furthermore since $\sigma \in [3/5, 1]$; $\frac{2}{5} \leq \frac{1+\sigma}{4}$ and $\frac{7+14\sigma-9\sigma^2}{40-24\sigma} \leq \frac{3}{4}$, which completes the part for bounds on monopoly prices. The bounds on duopoly prices is identical using equations in (C.37).

According to the definitions of $\underline{\ell}^m, \overline{\ell}^m, \underline{\ell}^d$, and $\overline{\ell}^d$, the bounds on the ratio of prices is:

$$\frac{\underline{\ell}^d}{\overline{\ell}^m} \leq \frac{\ell_{ij}^d}{\ell_{ij}^m} \leq \frac{\overline{\ell}^d}{\underline{\ell}^m}. \tag{C.91}$$

By plugging in the expressions of $\underline{\ell}^m, \overline{\ell}^m, \underline{\ell}^d$, and $\overline{\ell}^d$ we get the desired inequality.

C.2.7 Proof of Proposition 4.3.7

From (C.13), the demand function of the monopoly with the optimal prices is:

$$D(\ell_{ij}^m, \infty) = \frac{1 + \sigma - \frac{2\ell_{ij}^m}{\ell_{\max}}}{2\sigma}, \tag{C.92}$$

since $\ell_{ij}^m \in [(1-\sigma)\ell_{\max}, \sigma\ell_{\max}]$ under Assumption 4.3.2. Plugging in the expressions for $\underline{\ell}^m$ and $\overline{\ell}^m$ and imposing the condition $\sigma \in [3/5, 1]$, we get the desired bounds on (4.28).

The duopoly demand function for $\ell_{ij}^d = \underline{\ell}^d$ is given by (C.28) as:

$$\overline{D}^d = D(\underline{\ell}^d, \underline{\ell}^d) = \frac{1}{2} - \frac{(\frac{2\underline{\ell}^d}{\ell_{\max}} + \sigma - 1)^2}{8\sigma(1-\sigma)}, \tag{C.93}$$

since $\underline{\ell}^d \leq (1-\sigma)\ell_{\max}$. The duopoly demand function for $\ell_{ij}^d = \overline{\ell}^d$ is given by (C.34) as:

$$\underline{D}^d = D(\overline{\ell}^d, \overline{\ell}^d) = \frac{3 + \sigma - 4\frac{\overline{\ell}^d}{\ell_{\max}}}{8\sigma}, \tag{C.94}$$

since $\overline{\ell}^d \geq (1-\sigma)\ell_{\max}$. By plugging in the expressions for $\underline{\ell}^d$ and $\overline{\ell}^d$ and imposing the condition $\sigma \in [3/5, 1]$, we get the desired inequalities in (4.29).

According to the definitions of $\underline{D}^m, \overline{D}^m, \underline{D}^d$, and $\overline{D}^d$, the bounds on the ratio of demand functions is:

$$\frac{\underline{D}^d}{\overline{D}^m} \leq \frac{D^d}{D^m} \leq \frac{\overline{D}^d}{\underline{D}^m}. \tag{C.95}$$

By plugging in the expressions of $\underline{D}^m, \overline{D}^m, \underline{D}^d$, and $\overline{D}^d$ and using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.30).

C.2.8 Proof of Proposition 4.3.8

The total profits generated in monopoly is given by (4.19). Accordingly, the profits earned by serving the induced demand between OD pair (i, j) is:

$$P_{ij}^m = \frac{\theta_{ij}}{4\sigma\ell_{\max}}(\ell_{\max}(1+\sigma) - 2\ell_{ij}^m)^2 \tag{C.96}$$

Furthermore, lower optimal monopoly prices generate higher profits according to (4.19). Hence, the upper bound on P_{ij}^m is given by:

$$\frac{\theta_{ij}}{4\sigma\ell_{\max}}(\ell_{\max}(1+\sigma) - 2\underline{\ell}^m)^2 = \theta_{ij}\frac{(1+\sigma)^2}{16\sigma} = \theta_{ij}\overline{P}^m, \tag{C.97}$$

To get the lower bound, we evaluate (C.96) at $\ell_{ij}^m = \overline{\ell}^m$. By using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.31).

For the profits generated in duopoly, we first show that lower duopoly equilibrium prices generate higher profits. Since (4.21) bears a similar form to (4.16), the dual objective with the optimal prices and dual variables can be stated as (similar to (C.11a)):

$$P^d = \sum_{i=1}^{n}\sum_{j=1}^{n}\theta_{ij}D(\ell_{ij}^d, \ell_{ij}^d)\left(\ell_{ij}^d - \lambda_{ij}^d\right) = \sum_{i=1}^{n}\sum_{j=1}^{n}P_{ij}^d, \tag{C.98}$$

where we define

$$P_{ij}^d = \theta_{ij}D(\ell_{ij}^d, \ell_{ij}^d)\left(\ell_{ij}^d - \lambda_{ij}^d\right) \tag{C.99}$$

to be profits earned by serving the induced demand between OD pair (i, j). By taking the

derivative of P_{ij}^d with respect to ℓ_{ij}^d:

$$\frac{dP_{ij}^d}{d\ell_{ij}^d} = \theta_{ij}\left(\frac{dD(\ell_{ij}^d, \ell_{ij}^d)}{d\ell_{ij}^d}\left(\ell_{ij}^d - \lambda_{ij}^d\right) + D(\ell_{ij}^d, \ell_{ij}^d)(1 - \frac{d\lambda_{ij}^d}{d\ell_{ij}^d})\right) \tag{C.100}$$

From (C.37), $\frac{d\ell_{ij}^d}{d\lambda_{ij}^d} \leq 1$. Hence, $\frac{d\lambda_{ij}^d}{d\ell_{ij}^d} \geq 1$. Furthermore, $\ell_{ij}^d \geq \lambda_{ij}^d$ according to (C.37). Finally from (C.28) and (C.34) (evaluated at $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d$), $\frac{dD(\ell_{ij}^d, \ell_{ij}^d)}{d\ell_{ij}^d} \leq 0$. All in all that gives:

$$\frac{dP_{ij}^d}{d\ell_{ij}^d} \leq 0,$$

which means lower duopoly equilibrium prices generate higher profits. In order to get the upper bound, we evaluate (C.99) at $\ell_{ij}^d = \underline{\ell}^d$ (and $\lambda_{ij}^d = 0$ in this case)[1]. To get the lower bound, we evaluate (C.99) at $\ell_{ij}^d = \overline{\ell}^d$ (and $\lambda_{ij}^d = \frac{(3\sigma-1)(3-\sigma)}{4(5-3\sigma)}\ell_{\max}$ in this case)[2]. To get the desired inequality at (4.32), we impose $\sigma \in [3/5, 1]$.

According to the definitions of $\underline{P}^m, \overline{P}^m, \underline{P}^d$, and $\overline{P}^d$, the bounds on the ratio of profits earned by serving the induced demand for OD pair (i, j) is:

$$\frac{\underline{P}^d}{\overline{P}^m} \leq \frac{P_{ij}^d}{P_{ij}^m} \leq \frac{\overline{P}^d}{\underline{P}^m}. \tag{C.101}$$

By plugging in the expressions of $\underline{P}^m, \overline{P}^m, \underline{P}^d$, and $\overline{P}^d$ and using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.33).

[1]When $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d = \underline{\ell}^d \leq (1-\sigma)\ell_{\max}$, (4.21) is not a convex optimization problem (since the demand function is concave in that region). Hence, strong duality might not hold. However, since we are computing an upper bound on the objective function, the objective value of (4.21) is always less than or equal to the objective value of the dual problem given by (C.98), due to weak duality. Hence, the upper bound holds and is tight when strong duality holds.

[2]When $\ell_{ij}^1 = \ell_{ij}^2 = \ell_{ij}^d = \overline{\ell}^d \geq (1-\sigma)\ell_{\max}$, (4.21) is a convex optimization problem since the demand function is linear in that region. Hence, strong duality holds and the value of (C.98) is equal to the objective value of (4.21).

C.2.9 Proof of Proposition 4.3.9

The consumer surplus in monopoly is given by (4.20). Accordingly, the consumer surplus of customers requesting a ride between OD pair (i,j) is:

$$\text{CS}_{ij}^m = \theta_{ij}\frac{\ell_{\max}(\sigma^2+\sigma+1) - 3\ell_{ij}^m(1+\sigma-\frac{\ell_{ij}^m}{\ell_{\max}})}{6\sigma}, \tag{C.102}$$

Observe that $\frac{\partial \text{CS}_{ij}^m}{\partial \ell_{ij}^m} = \theta_{ij}\frac{6\frac{\ell_{ij}^m}{\ell_{\max}}-3\sigma-3}{6\sigma} \leq 0$ for $\ell_{ij}^m \in [(1-\sigma)\ell_{\max}, \sigma\ell_{\max}]$. Hence, lower optimal monopoly prices generate higher consumer surplus. Since $\ell_{ij}^m \geq \underline{\ell}^m = \frac{1+\sigma}{4}\ell_{\max}$, the upper bound on CS_{ij}^m is given by:

$$\theta_{ij}\frac{\ell_{\max}(\sigma^2+\sigma+1) - 3\ell_{\max}\frac{1+\sigma}{4}(1+\sigma-\frac{1+\sigma}{4})}{6\sigma} = \theta_{ij}\frac{7\sigma^2-2\sigma+7}{96\sigma}\ell_{\max} = \theta_{ij}\overline{\text{CS}}^m. \tag{C.103}$$

Similarly, the lower bound on the consumer surplus is given by evaluating (C.102) at $\ell_{ij}^m = \overline{\ell}^m = \frac{7+14\sigma-9\sigma^2}{40-24\sigma}\ell_{\max}$. By using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.34).

Similar to the monopoly, lower duopoly prices generate higher consumer surplus by inducing more customers and generating higher surplus per customer. For $\ell_{ij}^d = \underline{\ell}^d \leq (1-\sigma)\ell_{\max}$, the upper bound on the consumer surplus of customers requesting a ride between OD pair (i,j) is computed as:

$$\begin{aligned}
&2\theta_{ij}\Big(\int_{\frac{\ell_{\max}}{2}}^{\frac{\underline{\ell}^d}{1-\sigma}}\int_{\frac{\underline{\ell}^d-(1-\sigma)y}{\sigma}}^{\ell_{\max}}(\sigma x+(1-\sigma)y-\underline{\ell}^d)dxdy + \int_{\frac{\underline{\ell}^d}{1-\sigma}}^{\ell_{\max}}\int_{0}^{\ell_{\max}}(\sigma x+(1-\sigma)y-\underline{\ell}^d)dxdy\Big)\\
&\quad= \frac{\theta_{ij}\ell_{\max}}{24\sigma(1-\sigma)}\Big((2\sigma)^3 - (\sigma+1-2\frac{\underline{\ell}^d}{\ell_{\max}})^3 - 24\sigma(1-\frac{\underline{\ell}^d}{\ell_{\max}})(\sigma-1+\frac{\underline{\ell}^d}{\ell_{\max}})\Big)\\
&\quad= \theta_{ij}\overline{\text{CS}}^d.
\end{aligned} \tag{C.104}$$

where the factor 2 is due to the symmetry of two firms.

For $\ell_{ij}^d = \overline{\ell}^d = \frac{1+\sigma}{4}\ell_{\max} \geq (1-\sigma)\ell_{\max}$, the lower bound is computed as:

$$2\theta_{ij}\int_{\frac{\ell_{\max}}{2}}^{\ell_{\max}}\int_{\frac{\overline{\ell}^d-(1-\sigma)y}{\sigma}}^{\ell_{\max}}(\sigma x+(1-\sigma)y-\overline{\ell}^d)dxdy = \theta_{ij}\frac{\sigma^2-2\sigma+13}{96\sigma}\ell_{\max} = \theta_{ij}\underline{\mathrm{CS}}^d \quad \text{(C.105)}$$

By using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.35).

According to the definitions of $\underline{\mathrm{CS}}^m, \overline{\mathrm{CS}}^m, \underline{\mathrm{CS}}^d$, and $\overline{\mathrm{CS}}^d$, the bounds on the ratio of consumer surplus of customers requesting ride between OD pair (i, j) is:

$$\frac{\underline{\mathrm{CS}}^d}{\overline{\mathrm{CS}}^m} \leq \frac{\mathrm{CS}_{ij}^d}{\mathrm{CS}_{ij}^m} \leq \frac{\overline{\mathrm{CS}}^d}{\underline{\mathrm{CS}}^m}. \quad \text{(C.106)}$$

By plugging in the expressions of $\underline{\mathrm{CS}}^m, \overline{\mathrm{CS}}^m, \underline{\mathrm{CS}}^d$, and $\overline{\mathrm{CS}}^d$ and using the condition $\sigma \in [3/5, 1]$, we get the desired inequality in (4.36).

C.2.10 MPC with Dynamic Prices in Duopoly

One possible way to model the real-time duopoly pricing is an alternating-move duopoly game. Specifically, every even t_0, firm 1 sets new prices and executes fleet decisions, whereas firm 2 only executes fleet decisions while keeping prices same as the previous time period. Every odd t_0, firm 2 sets new prices and executes fleet decisions, whereas firm 1 only executes fleet decisions while keeping prices same as the previous time period. Furthermore, every even t_0, firm 1 is able to observe firm 2's prices at the planning time, however, the future prices of firm 2 depend on firm 2's future states, which is unavailable to firm 1. Every odd t_0 however, since firm 2 will set the prices, firm 1 is oblivious to what firm 2's prices will be, including the planning time. One possible way of planning for these uncertainties would be to assume that firm 2's unknown prices would be the symmetric duopoly equilibrium prices and determine the best strategy accordingly. In respect to these modeling specifications, we can formulate MPC optimization problem with dynamic prices in the duopoly with slight modifications to (4.37).

In particular, every even t_0 firm 1 solves (4.37) with

$$D(\ell^1_{ijt_0}, \infty) \leftarrow D(\ell^1_{ijt_0}, \ell^2_{ijt_0}), \tag{C.107}$$

$$D(\ell^1_{ijt}, \infty) \leftarrow D(\ell^1_{ijt}, \ell^d_{ij}), \quad \forall t > t_0, \tag{C.108}$$

$$\ell^1_{ijt} = \ell^1_{ijt-1}, \quad \forall i, j \in \mathcal{N},\ \forall t = t_0 + 2k - 1, k \in \mathbb{Z}^+, \tag{C.109}$$

where $\ell^2_{ijt_0} = \ell^2_{ijt_0-1}$. Every odd t_0, firm 1 solves (4.37) with

$$D(\ell^1_{ijt}, \infty) \leftarrow D(\ell^1_{ijt}, \ell^d_{ij}), \quad \forall t \geq t_0 \tag{C.110}$$

$$\ell^1_{ijt} = \ell^1_{ijt-1}, \quad \forall i, j \in \mathcal{N},\ \forall t = t_0 + 2k - 2, k \in \mathbb{Z}^+. \tag{C.111}$$

The same method is applied for firm 2 with odd/even t switched.

Bibliography

[1] B. Turan, S. Hutchinson, and M. Alizadeh, *A safe first-order method for pricing-based resource allocation in safety-critical networks*, *arXiv preprint arXiv:2310.03808* (2023).

[2] B. Turan and M. Alizadeh, *Competition in electric autonomous mobility-on-demand systems*, *IEEE Transactions on Control of Network Systems* **9** (2022), no. 1 295–307.

[3] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, *Robust distributed optimization with randomly corrupted gradients*, *IEEE Transactions on Signal Processing* **70** (2022) 3484–3498.

[4] B. Turan, R. Pedarsani, and M. Alizadeh, *Dynamic pricing and fleet management for electric autonomous mobility on demand systems*, *Transportation Research Part C: Emerging Technologies* **121** (2020) 102829.

[5] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, *Resilient primal–dual optimization algorithms for distributed resource allocation*, *IEEE Transactions on Control of Network Systems* **8** (2020), no. 1 282–294.

[6] S. Hutchinson, B. Turan, and M. Alizadeh, *The impact of the geometric properties of the constraint set in safe optimization with bandit feedback*, in *Learning for Dynamics and Control Conference*, pp. 497–508, PMLR, 2023.

[7] S. Hutchinson, B. Turan, and M. Alizadeh, *A safe pricing mechanism for distributed resource allocation with bandit feedback*, in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5092–5098, IEEE, 2022.

[8] A. Moradipari, B. Turan, Y. Abbasi-Yadkori, M. Alizadeh, and M. Ghavamzadeh, *Feature and parameter selection in stochastic linear bandits*, in *International Conference on Machine Learning*, pp. 15927–15958, PMLR, 2022.

[9] B. Turan and M. Alizadeh, *Safe dual gradient method for network utility maximization problems*, in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 6953–6959, IEEE, 2022.

[10] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, *On robustness of the normalized random block coordinate method for non-convex optimization*, in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 974–980, IEEE, 2021.

[11] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, *On robustness of the normalized subgradient method with randomly corrupted subgradients*, in *2021 American Control Conference (ACC)*, pp. 965–971, IEEE, 2021.

[12] N. Tucker, B. Turan, and M. Alizadeh, *Online charge scheduling for electric vehicles in autonomous mobility on demand fleets*, in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 226–231, IEEE, 2019.

[13] B. Turan, N. Tucker, and M. Alizadeh, *Smart charging benefits in autonomous mobility on demand systems*, in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 461–466, IEEE, 2019.

[14] S. H. Low and D. E. Lapsley, *Optimization flow control. i. basic algorithm and convergence*, *IEEE/ACM Transactions on networking* **7** (1999), no. 6 861–874.

[15] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. Wong, and J. Jatskevich, *Optimal real-time pricing algorithm based on utility maximization for smart grid*, in *2010 First IEEE international conference on smart grid communications*, pp. 415–420, IEEE, 2010.

[16] N. Li, L. Chen, and S. H. Low, *Optimal demand response based on utility maximization in power networks*, in *2011 IEEE power and energy society general meeting*, pp. 1–8, IEEE, 2011.

[17] M. Chiang and J. Bell, *Balancing supply and demand of bandwidth in wireless cellular networks: utility maximization over powers and rates*, in *IEEE INFOCOM 2004*, vol. 4, pp. 2800–2811, IEEE, 2004.

[18] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, *Rate control for communication networks: shadow prices, proportional fairness and stability*, *Journal of the Operational Research society* **49** (1998) 237–252.

[19] M. Dehghan, L. Massoulie, D. Towsley, D. S. Menasche, and Y. C. Tay, *A utility optimization approach to network cache design*, *IEEE/ACM Transactions on Networking* **27** (2019), no. 3 1013–1027.

[20] M. Zhao, J. Li, and Y. Yang, *Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks*, in *2011 23rd International Teletraffic Congress (ITC)*, pp. 238–245, IEEE, 2011.

[21] R. Deng, Y. Zhang, S. He, J. Chen, and X. Shen, *Maximizing network utility of rechargeable sensor networks with spatiotemporally coupled constraints*, *IEEE Journal on Selected Areas in Communications* **34** (2016), no. 5 1307–1319.

[22] N. Mehr, J. Lioris, R. Horowitz, and R. Pedarsani, *Joint perimeter and signal control of urban traffic via network utility maximization*, in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2017.

[23] D. P. Palomar and M. Chiang, *A tutorial on decomposition methods for network utility maximization*, *IEEE JSAC* **24** (2006), no. 8 1439–1451.

[24] J. Koshal, A. Nedić, and U. V. Shanbhag, *Multiuser optimization: Distributed algorithms and error analysis*, *SIAM Journal on Optimization* **21** (2011), no. 3 1046–1081.

[25] C.-K. Yu, M. Van Der Schaar, and A. H. Sayed, *Distributed learning for stochastic generalized nash equilibrium problems*, *IEEE Transactions on Signal Processing* **65** (2017), no. 15 3893–3908.

[26] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, *A survey of distributed optimization*, *Annual Reviews in Control* **47** (2019) 278–305.

[27] P. Mertikopoulos, E. V. Belmega, R. Negrel, and L. Sanguinetti, *Distributed stochastic optimization via matrix exponential learning*, *IEEE Transactions on Signal Processing* **65** (2017), no. 9 2277–2290.

[28] Y. Chen, S. Kar, and J. M. Moura, *The internet of things: Secure distributed inference*, *IEEE Signal Processing Magazine* **35** (2018), no. 5 64–75.

[29] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, *Federated learning: Strategies for improving communication efficiency*, *arXiv preprint arXiv:1610.05492* (2016).

[30] A. Vempaty, L. Tong, and P. K. Varshney, *Distributed inference with byzantine data: State-of-the-art review on data falsification attacks*, *IEEE Signal Processing Magazine* **30** (2013), no. 5 65–75.

[31] S. Sundaram and C. N. Hadjicostis, *Distributed function calculation via linear iterative strategies in the presence of malicious agents*, *IEEE TAC* **56** (2011), no. 7 1495–1508.

[32] F. Pasqualetti, A. Bicchi, and F. Bullo, *Consensus computation in unreliable networks: A system theoretic approach*, *IEEE TAC* **57** (2012), no. 1 90–104.

[33] R. Gentz, S. X. Wu, H.-T. Wai, A. Scaglione, and A. Leshem, *Data injection attacks in randomized gossiping*, *IEEE TSIPN* **2** (2016), no. 4 523–538.

[34] S. Sundaram and B. Gharesifard, *Distributed optimization under adversarial nodes*, *IEEE TAC* (2018).

[35] Y. Chen, S. Kar, and J. Moura, *Resilient distributed estimation: Sensor attacks*, *IEEE Transactions on Automatic Control* (2018).

[36] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, *Robust distributed consensus using total variation*, *IEEE Transactions on Automatic Control* **61** (2016), no. 6 1550–1564.

[37] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, *Resilient asymptotic consensus in robust networks*, *IEEE Journal on Selected Areas in Communications* **31** (April, 2013) 766–781.

[38] J. S. Baras and X. Liu, *Trust is the cure to distributed consensus with adversaries*, in *2019 27th Mediterranean Conference on Control and Automation (MED)*, pp. 195–202, July, 2019.

[39] N. Ravi, A. Scaglione, and A. Nedić, *A case of distributed optimization in adversarial environment*, in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5252–5256, May, 2019.

[40] J. Feng, H. Xu, and S. Mannor, *Distributed Robust Learning*, *arXiv preprint arXiv:1409.5937* (2014).

[41] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, *Byzantine-robust distributed learning: Towards optimal statistical rates*, in *International Conference on Machine Learning*, pp. 5650–5659, PMLR, 2018.

[42] D. Alistarh, Z. Allen-Zhu, and J. Li, *Byzantine stochastic gradient descent*, in *NeurIPS*, pp. 4618–4628, 2018.

[43] Y. Chen, L. Su, and J. Xu, *Distributed statistical machine learning in adversarial settings: Byzantine gradient descent*, *Proc. ACM Meas. Anal. Comput. Syst.* **1** (2017), no. 2 44:1–44:25.

[44] D. Data, L. Song, and S. Diggavi, *Data encoding methods for byzantine-resilient distributed optimization*, in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2719–2723, July, 2019.

[45] D. L. Donoho and P. J. Huber, *The notion of breakdown point*, *A festschrift for Erich L. Lehmann* **157184** (1983).

[46] P. J. Huber, *Robust statistics*. Springer, 2011.

[47] S. Minsker *et. al.*, *Geometric median and robust estimation in banach spaces*, *Bernoulli* **21** (2015), no. 4 2308–2335.

[48] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, *Robust estimators in high dimensions without the computational intractability*, in *IEEE FOCS*, pp. 655–664, 2016.

[49] J. Steinhardt, M. Charikar, and G. Valiant, *Resilience: A criterion for learning in the presence of arbitrary outliers*, *arXiv preprint arXiv:1703.04940* (2017).

[50] Y. Nesterov, *Smooth minimization of non-smooth functions*, *Mathematical programming* **103** (2005), no. 1 127–152.

[51] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, *A dual approach for optimal algorithms in distributed optimization over networks*, *Optimization Methods and Software* (2020) 1–40.

[52] D. Blatt, A. O. Hero, and H. Gauchman, *A convergent incremental gradient method with a constant step size*, *SIAM Journal on Optimization* **18** (2007), no. 1 29–51.

[53] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo, *On the convergence rate of incremental aggregated gradient algorithms*, *SIAM Journal on Optimization* **27** (2017), no. 2 1035–1048.

[54] P. Tseng and S. Yun, *Incrementally updated gradient methods for constrained and regularized optimization*, *Journal of Optimization Theory and Applications* **160** (Mar, 2014) 832–853.

[55] M. Grant and S. Boyd, *Cvx: Matlab software for disciplined convex programming, version 2.1*, 2014.

[56] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, *Matpower: Steady-state operations, planning, and analysis tools for power systems research and education*, *IEEE Transactions on Power Systems* **26** (Feb, 2011) 12–19.

[57] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et. al.*, *Advances and open problems in federated learning*, *arXiv preprint arXiv:1912.04977* (2019).

[58] Z. Yang, A. Gang, and W. U. Bajwa, *Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the byzantine threat model*, *IEEE Signal Processing Magazine* **37** (2020), no. 3 146–159.

[59] V. Mnih and G. E. Hinton, *Learning to label aerial images from noisy data*, in *Proceedings of the 29th International conference on machine learning (ICML-12)*, pp. 567–574, 2012.

[60] F. Tramer and D. Boneh, *Adversarial training and robustness for multiple perturbations*, *arXiv preprint arXiv:1904.13000* (2019).

[61] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, *Qsgd: Communication-efficient sgd via gradient quantization and encoding*, *Advances in neural information processing systems* **30** (2017).

[62] F. S. Stonyakin, D. Dvinskikh, P. Dvurechensky, A. Kroshnin, O. Kuznetsova, A. Agafonov, A. Gasnikov, A. Tyurin, C. A. Uribe, D. Pasechnyuk, and S. Artamonov, *Gradient methods for problems with inexact model of the objective*, in *Mathematical Optimization Theory and Operations Research* (M. Khachay, Y. Kochetov, and P. Pardalos, eds.), (Cham), pp. 97–114, Springer International Publishing, 2019.

[63] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, *Learning with noisy labels*, *Advances in neural information processing systems* **26** (2013).

[64] J. Steinhardt, P. W. Koh, and P. Liang, *Certified defenses for data poisoning attacks*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 3520–3532, Curran Associates Inc., 2017.

[65] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, *Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks*, *IEEE Trans. Signal Process.* **68** (2020) 4583–4596.

[66] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, *How to backdoor federated learning*, in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020.

[67] Y. Chen, L. Su, and J. Xu, *Distributed statistical machine learning in adversarial settings: Byzantine gradient descent*, *Proceedings of the ACM on Measurement and Analysis of Computing Systems* **1** (2017), no. 2 1–25.

[68] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, *Draco: Byzantine-resilient distributed training via redundant gradients*, in *International Conference on Machine Learning*, pp. 903–912, PMLR, 2018.

[69] X. Cao and L. Lai, *Distributed approximate newton's method robust to byzantine attackers*, *IEEE Transactions on Signal Processing* **68** (2020) 6011–6025.

[70] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, *Machine learning with adversaries: Byzantine tolerant gradient descent*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

[71] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, *Defending against saddle point attack in byzantine-robust distributed learning*, in *International Conference on Machine Learning*, pp. 7074–7084, PMLR, 2019.

[72] Z. Yang and W. U. Bajwa, *Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning*, *IEEE Trans. Signal Inf. Process. Netw.* **5** (2019), no. 4 611–627.

[73] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, *Attack of the tails: Yes, you really can backdoor federated learning*, *arXiv preprint arXiv:2007.05084* (2020).

[74] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, *A survey on security control and attack detection for industrial cyber-physical systems*, *Neurocomputing* **275** (2018) 1674–1683.

[75] B. Bhushan, G. Sahoo, and A. K. Rai, *Man-in-the-middle attack in wireless and computer networking—a review*, in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*, pp. 1–6, IEEE, 2017.

[76] Y. Nesterov, *Introductory Lectures on Convex Optimization*, vol. 87. Springer Science & Business Media, 2004.

[77] E. Hazan, K. Levy, and S. Shalev-Shwartz, *Beyond convexity: Stochastic quasi-convex optimization*, in *Advances in Neural Information Processing Systems*, pp. 1594–1602, 2015.

[78] Y. You, I. Gitman, and B. Ginsburg, *Large batch training of convolutional networks*, *arXiv preprint arXiv:1708.03888* (2017).

[79] K. Y. Levy, *The power of normalization: Faster evasion of saddle points*, *arXiv preprint arXiv:1611.04831* (2016).

[80] A. Cutkosky and H. Mehta, *Momentum improves normalized sgd*, in *International Conference on Machine Learning*, pp. 2260–2268, PMLR, 2020.

[81] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, *Deep learning with differential privacy*, in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

[82] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, *Can you really backdoor federated learning?*, *arXiv preprint arXiv:1911.07963* (2019).

[83] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, *Can gradient clipping mitigate label noise?*, in *International Conference on Learning Representations*, 2019.

[84] A. Nedić, D. P. Bertsekas, and V. S. Borkar, *Distributed asynchronous incremental subgradient methods*, *Studies in Computational Mathematics* **8** (2001), no. C 381–407.

[85] M. Gurbuzbalaban, A. Ozdaglar, and P. A. Parrilo, *On the convergence rate of incremental aggregated gradient algorithms*, *SIAM Journal on Optimization* **27** (2017), no. 2 1035–1048.

[86] A. Agarwal and J. C. Duchi, *Distributed delayed stochastic optimization*, in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 5451–5452, IEEE, 2012.

[87] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, *The sample average approximation method for stochastic discrete optimization*, *SIAM Journal on Optimization* **12** (2002), no. 2 479–502.

[88] M. T. Wasan, *Stochastic approximation*. No. 58. Cambridge University Press, 2004.

[89] S. Kim, R. Pasupathy, and S. G. Henderson, *A guide to sample average approximation*, *Handbook of simulation optimization* (2015) 207–243.

[90] C. A. León and F. Perron, *Optimal hoeffding bounds for discrete reversible markov chains*, *The Annals of Applied Probability* **14** (2004), no. 2 958–970.

[91] G. P. Wadsworth, *Introduction to probability and random variables*, p. 52. McGraw-Hill book Company, 1960.

[92] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[93] D. A. Levin and Y. Peres, *Markov chains and mixing times*, vol. 107. American Mathematical Soc., 2017.

[94] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, *Emnist: Extending mnist to handwritten letters*, in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, IEEE, 2017.

[95] I. Necoara, V. Nedelcu, and I. Dumitrache, *Parallel and distributed optimization methods for estimation and control in networks*, *Journal of Process Control* **21** (2011), no. 5 756–766.

[96] D. P. Bertsekas, *Nonlinear programming*, *Journal of the Operational Research Society* **48** (1997), no. 3 334–334.

[97] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[98] N. Z. Shor, *Minimization methods for non-differentiable functions*, vol. 3. Springer Science & Business Media, 2012.

[99] A. Nedić and A. Ozdaglar, *Approximate primal solutions and rate analysis for dual subgradient methods*, *SIAM Journal on Optimization* **19** (2009), no. 4 1757–1780.

[100] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, *An $o(1/k)$ gradient method for network resource allocation problems*, *IEEE Transactions on Control of Network Systems* **1** (2014), no. 1 64–73.

[101] I. Necoara and V. Nedelcu, *Rate analysis of inexact dual first-order methods application to dual decomposition*, *IEEE Transactions on Automatic Control* **59** (2013), no. 5 1232–1243.

[102] I. Necoara and V. Nedelcu, *On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems*, *Automatica* **55** (2015) 209–216.

[103] A. Simonetto and H. Jamali-Rad, *Primal recovery from consensus-based dual decomposition for distributed convex optimization*, *Journal of Optimization Theory and Applications* **168** (2016) 172–197.

[104] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, *Dual decomposition for multi-agent distributed optimization with coupling constraints*, *Automatica* **84** (2017) 149–158.

[105] I. Notarnicola and G. Notarstefano, *Constraint-coupled distributed optimization: a relaxation and duality approach*, *IEEE Transactions on Control of Network Systems* **7** (2019), no. 1 483–492.

[106] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, *A survey on demand response programs in smart grids: Pricing methods and optimization algorithms*, *IEEE Communications Surveys & Tutorials* **17** (2014), no. 1 152–178.

[107] M. Zhu and S. Martinez, *On distributed convex optimization under inequality and equality constraints*, *IEEE Transactions on Automatic Control* **57** (2011), no. 1 151–164.

[108] K. Sakurama and M. Miura, *Distributed constraint optimization on networked multi-agent systems*, *Applied Mathematics and Computation* **292** (2017) 272–281.

[109] Y. Zheng and Q. Liu, *A review of distributed optimization: Problems, models and algorithms*, *Neurocomputing* **483** (2022) 446–459.

[110] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, *Mathematical programming* **39** (1987), no. 1 93–116.

[111] E. S. Levitin and B. T. Polyak, *Constrained minimization methods*, *USSR Computational mathematics and mathematical physics* **6** (1966), no. 5 1–50.

[112] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[113] P. Armand, J. C. Gilbert, and S. Jan-Jégou, *A feasible bfgs interior point algorithm for solving convex minimization problems*, *SIAM Journal on Optimization* **11** (2000), no. 1 199–222.

[114] E. Wei, A. Ozdaglar, and A. Jadbabaie, *A distributed newton method for network utility maximization*, in *49th IEEE Conference on Decision and Control (CDC)*, pp. 1816–1821, IEEE, 2010.

[115] S. Athuraliya and S. H. Low, *Optimization flow control with newton-like algorithm*, *Telecommunication Systems* **15** (2000), no. 3 345–358.

[116] I. Necoara and J. Suykens, *Interior-point lagrangian decomposition method for separable convex optimization*, *Journal of Optimization Theory and Applications* **143** (2009), no. 3 567–588.

[117] J. Mo and J. Walrand, *Fair end-to-end window-based congestion control*, *IEEE/ACM Transactions on networking* **8** (2000), no. 5 556–567.

[118] R. Schneider, *Convex bodies: the Brunn–Minkowski theory*. No. 151. Cambridge university press, 2014.

[119] [Online]. Available: https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/.

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

[121] R. Zhang and M. Pavone, *Control of robotic Mobility-on-Demand systems: A queueing-theoretical perspective*, In *Int. Journal of Robotics Research* **35** (2016), no. 1–3 186–203.

[122] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, *Robotic load balancing for Mobility-on-Demand systems*, *Int. Journal of Robotics Research* **31** (2012), no. 7 839–854.

[123] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, *Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms*, *Autonomous Robots* **42** (2018), no. 7 1427–1442.

[124] M. Volkov, J. Aslam, and D. Rus, *Markov-based redistribution policy model for future urban mobility networks*, *Conference Record - IEEE Conference on Intelligent Transportation Systems* (09, 2012) 1906–1911.

[125] Q. Wei, J. A. Rodriguez, R. Pedarsani, and S. Coogan, *Ride-sharing networks with mixed autonomy*, in *2019 American Control Conference (ACC)*, pp. 3303–3308, IEEE, 2019.

[126] R. Zhang, F. Rossi, and M. Pavone, *Model predictive control of autonomous mobility-on-demand systems*, in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May, 2016.

[127] F. Miao, S. Lin, S. Munir, J. A. Stankovic, H. Huang, D. Zhang, T. He, and G. J. Pappas, *Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach*, in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pp. 100–109, 2015.

[128] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, *Data-driven model predictive control of autonomous mobility-on-demand systems*, in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6019–6025, IEEE, 2018.

[129] F. Miao, S. Han, A. M. Hendawi, M. E. Khalefa, J. A. Stankovic, and G. J. Pappas, *Data-driven distributionally robust vehicle balancing using dynamic region partitions*, in *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 261–272, April, 2017.

[130] M. Tsao, R. Iglesias, and M. Pavone, *Stochastic model predictive control for autonomous mobility on demand*, in *2018 21st International conference on intelligent transportation systems (ITSC)*, pp. 3941–3948, IEEE, 2018.

[131] K. Spieser, S. Samaranayake, and E. Frazzoli, *Vehicle routing for shared-mobility systems with time-varying demand*, in *2016 American Control Conference (ACC)*, pp. 796–802, July, 2016.

[132] R. M. A. Swaszek and C. Cassandras, *Load balancing in mobility-on-demand systems: Reallocation via parametric control using concurrent estimation*, *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019) 2148–2153.

[133] M. Repoux, M. Kaspi, B. Boyacı, and N. Geroliminis, *Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations*, *Transportation Research Part B: Methodological* **130** (2019) 82–104.

[134] B. Boyacı, K. G. Zografos, and N. Geroliminis, *An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations*, *Transportation Research Part B: Methodological* **95** (2017) 214–237.

[135] J. Warrington and D. Ruchti, *Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems*, *Transportation Research Part C: Emerging Technologies* **104** (2019) 110–134.

[136] C. Mao and Z. Shen, *A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network*, *Transportation Research Part C: Emerging Technologies* **93** (2018) 179–197.

[137] F. Zhu and S. V. Ukkusuri, *Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach*, *Transportation Research Part C: Emerging Technologies* **41** (2014) 30 – 47.

[138] E. Walraven, M. T. Spaan, and B. Bakker, *Traffic flow optimization: A reinforcement learning approach*, *Engineering Applications of Artificial Intelligence* **52** (2016) 203 – 212.

[139] F. Zhu, H. A. Aziz, X. Qian, and S. V. Ukkusuri, *A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework*, *Transportation Research Part C: Emerging Technologies* **58** (2015) 487 – 501.

[140] L. Li, Y. Lv, and F. Wang, *Traffic signal timing via deep reinforcement learning*, *IEEE/CAA Journal of Automatica Sinica* **3** (2016), no. 3 247–254.

[141] D. A. Lazar, E. Bıyık, D. Sadigh, and R. Pedarsani, *Learning how to dynamically route autonomous vehicles on shared roads*, *Transportation research part C: emerging technologies* **130** (2021) 103258.

[142] M. Han, P. Senellart, S. Bressan, and H. Wu, *Routing an autonomous taxi with reinforcement learning*, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 2421–2424, 2016.

[143] M. Guériau and I. Dusparic, *Samod: Shared autonomous mobility-on-demand using decentralized reinforcement learning*, in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1558–1563, Nov, 2018.

[144] J. Wen, J. Zhao, and P. Jaillet, *Rebalancing shared mobility-on-demand systems: A reinforcement learning approach*, in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 220–225, Oct, 2017.

[145] K. Lin, R. Zhao, Z. Xu, and J. Zhou, *Efficient large-scale fleet management via multi-agent deep reinforcement learning*, in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1774–1783, 2018.

[146] C. Mao, Y. Liu, and Z.-J. M. Shen, *Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach*, *Transportation Research Part C: Emerging Technologies* **115** (2020) 102626.

[147] E. Veldman and R. A. Verzijlbergh, *Distribution grid impacts of smart electric vehicle charging from different perspectives*, *IEEE Transactions on Smart Grid* **6** (Jan, 2015) 333–342.

[148] W. Su, H. Eichi, W. Zeng, and M. Chow, *A survey on the electrification of transportation in a smart grid environment*, *IEEE Transactions on Industrial Informatics* **8** (Feb, 2012) 1–10.

[149] J. C. Mukherjee and A. Gupta, *A review of charge scheduling of electric vehicles in smart grid*, *IEEE Systems Journal* **9** (2014), no. 4 1541–1553.

[150] T. D. Chen, K. M. Kockelman, and J. P. Hanna, *Operations of a Shared, Autonomous, Electric Vehicle Fleet: Implications of Vehicle & Charging Infrastructure Decisions*, *Transportation Research Part A: Policy and and Practice* **94** (2016) 243–254.

[151] C. Bongiovanni, M. Kaspi, and N. Geroliminis, *The electric autonomous dial-a-ride problem*, *Transportation Research Part B: Methodological* **122** (2019) 436 – 456.

[152] F. Rossi, R. Iglesias, M. Alizadeh, and M. Pavone, *On the interaction between autonomous mobility-on-demand systems and the power network: Models and coordination algorithms*, *IEEE Transactions on Control of Network Systems* **7** (2019), no. 1 384–397.

[153] T. D. Chen and K. M. Kockelman, *Management of a shared autonomous electric vehicle fleet: Implications of pricing schemes*, *Transportation Research Record* **2572** (2016), no. 1 37–46.

[154] Y. Guan, A. M. Annaswamy, and H. E. Tseng, *Cumulative prospect theory based dynamic pricing for shared mobility on demand services*, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 2239–2244, IEEE, 2019.

[155] C. J. R. Sheppard, G. S. Bauer, B. F. Gerke, J. B. Greenblatt, A. T. Jenn, and A. R. Gopal, *Joint optimization scheme for the planning and operations of shared autonomous electric vehicle fleets serving mobility on demand*, *Transportation Research Record* **2673** (2019), no. 6 579–597, [https://doi.org/10.1177/0361198119838270].

[156] K. Bimpikis, O. Candogan, and D. Sabán, *Spatial pricing in ride-sharing networks*, *Operations Research* **67** (2019) 744–769.

[157] R. Pedarsani, J. Walrand, and Y. Zhong, *Robust scheduling for flexible processing networks*, *Advances in Applied Probability* **49** (2017), no. 2 603–628.

[158] L. P. Kaelbling, M. L. Littman, and A. W. Moore, *Reinforcement learning: A survey*, *Journal of artificial intelligence research* **4** (1996) 237–285.

[159] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, *Microscopic traffic simulation using sumo*, in *2018 21st international conference on intelligent transportation systems (ITSC)*, pp. 2575–2582, IEEE, 2018.

[160] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, *A survey of actor-critic reinforcement learning: Standard and natural policy gradients*, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42** (Nov, 2012) 1291–1307.

[161] C. J. Watkins and P. Dayan, *Q-learning*, *Machine learning* **8** (1992) 279–292.

[162] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[163] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, *Machine learning* **8** (1992) 229–256.

[164] A. G. Barto, R. S. Sutton, and C. W. Anderson, *Neuronlike adaptive elements that can solve difficult learning control problems*, *IEEE transactions on systems, man, and cybernetics* (1983), no. 5 834–846.

[165] I. H. Witten, *An adaptive optimal controller for discrete-time markov environments*, *Information and Control* **34** (1977) 286–295.

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

[167] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, *Openai gym*, *arXiv preprint arXiv:1606.01540* (2016).

[168] A. Hill, A. Raffin, M. Ernestus, A. Gleave, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines." `https://github.com/hill-a/stable-baselines`, 2018.

[169] "The average electric car in the US is getting cheaper." [Online]. Available: https://qz.com/1695602/the-average-electric-vehicle-is-getting-cheaper-in-the-us/.

[170] [Online]. Available: http://oasis.caiso.com.

[171] "United States Average Hourly Wages." [Online]. Available: https://tradingeconomics.com/united-states/wages.

[172] "How much does driving your car cost, per minute?." [Online]. Available: https://www.bostonglobe.com/ideas/2014/08/08/how-much-driving-really-costs-per-minute/BqnNd2q7jETedLhxxzY2CI/story.html.

[173] [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[174] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning*, *arXiv preprint arXiv:1509.02971* (2015).

[175] [Online]. Available: http://tncstoday.sfcta.org/.

[176] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, *Crawdad data set epfl/mobility (v. 2009-02-24)*, 2009.

[177] E. Weyl *et. al.*, *Imperfect platform competition: A general framework*, .

[178] J.-C. Rochet and J. Tirole, *Platform competition in two-sided markets*, *J. Eur. Econ. Assoc.* **1** (2003), no. 4 990–1029.

[179] M. Armstrong and J. Wright, *Two-sided markets, competitive bottlenecks and exclusive contracts*, *Economic Theory* **32** (2007), no. 2 353–380.

[180] Y. Dou and D. Wu, *Dynamic platform competition: Optimal pricing and piggybacking under network effects*, *SSRN Electronic Journal* (2016).

[181] T. Kodera *et. al.*, *Spatial competition among multiple platforms*, *Economics Bulletin* **30** (2010), no. 2 1561–1525.

[182] J. Cramer and A. B. Krueger, *Disruptive change in the taxi business: The case of uber*, *Am Econ Rev* **106** (2016), no. 5 177–82.

[183] M. McGregor, B. Brown, and M. Glöss, *Disrupting the cab: Uber, ridesharing and the taxi industry*, *J. Peer Prod.* (2015), no. 6.

[184] J. Mayer, *Uber Is Not A Monopoly*, 2016. [Online]. Available: https://www.forbes.com/sites/jaredmeyer/2016/02/15/uber-guardian-not-monopoly-ridesharing/.

[185] S. Jiang, L. Chen, A. Mislove, and C. Wilson, *On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi*, in *Proceedings of the 2018 World Wide Web Conference*, pp. 863–872, 2018.

[186] K. Bimpikis, O. Candogan, and D. Saban, *Spatial pricing in ride-sharing networks*, *Oper. Res.* **67** (2019), no. 3 744–769.

[187] J. C. Castillo, D. Knoepfle, and G. Weyl, *Surge pricing solves the wild goose chase*, in *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 241–242, 2017.

[188] S. Banerjee, C. Riquelme, and R. Johari, *Pricing in ride-share platforms: A queueing-theoretic approach*, *Avail. at SSRN 2568258* (2015).

[189] A. Nikzad, *Thickness and competition in ride-sharing markets*, *Available at SSRN 3065672* (2017).

[190] S. Dafermos, *The general multimodal network equilibrium problem with elastic demand*, *Networks* **12** (1982), no. 1 57–72.

[191] H. Yang, *Sensitivity analysis for the elastic-demand network equilibrium problem with applications*, *Transportation Research Part B: Methodological* **31** (1997), no. 1 55–70.

[192] L. Gurobi Optimization, *Gurobi optimizer reference manual*, 2020.

[193] Edmunds, *The True Cost of Powering an Electric Car*, 2019. [Online]. Available: https://www.edmunds.com/fuel-economy/the-true-cost-of-powering-an-electric-car.html.

[194] S. Wollenstein-Betech, I. C. Paschalidis, and C. G. Cassandras, *Joint pricing and rebalancing of autonomous mobility-on-demand systems*, in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2573–2578, IEEE, 2020.

[195] T. Chen, Q. Ling, and G. B. Giannakis, *An online convex optimization approach to proactive network resource allocation*, *IEEE Transactions on Signal Processing* **65** (2017), no. 24 6350–6364.

[196] D. Gammelli, K. Yang, J. Harrison, F. Rodrigues, F. C. Pereira, and M. Pavone, *Graph neural network reinforcement learning for autonomous mobility-on-demand systems*, in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2996–3003, IEEE, 2021.

[197] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, *A delayed proximal gradient method with linear convergence rate*, in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Sep., 2014.

[198] H. J. Helgert, *On sums of random variables defined on a two-state markov chain*, *Journal of Applied Probability* **7** (1970), no. 3 761–765.

[199] Y. Nesterov and B. T. Polyak, *Cubic regularization of newton method and its global performance*, *Mathematical Programming* **108** (2006), no. 1 177–205.

[200] J. G. Dai, *On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models*, *The Annals of Applied Probability* **5** (1995), no. 1 49–77.