Extensions and Applications of Fast-Lipschitz Optimization

MARTIN JAKOBSSON

Licentiate Thesis
Stockholm, Sweden, 2014
Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i elektro- och systemteknik fredagen den 5 september 2012, klockan 10.15 i sal L1, Kungliga Tekniska högskolan, Drottning Kristinas väg 30, Stockholm.

© Martin Jakobsson, 2014

Tryck: Universitetsservice US AB
Abstract

Fast-Lipschitz optimization is a framework for optimization problems having a special structure in which the optimal solution is given by a set of equations, easily solvable through fixed point iterations. This solution method is simple to implement and particularly well suited for distributed optimization applications, such as those arising in wireless sensor networks.

This thesis provides several theoretical contributions to the Fast-Lipschitz framework. In particular, it further develops the qualifying conditions under which a problem is guaranteed to be Fast-Lipschitz. Known qualifying conditions are unified and extended by a new set of conditions. This is done through a new approach to the analysis of existing conditions, based on the Karush–Kuhn–Tucker (KKT) conditions. The Fast-Lipschitz framework is further extended by examining problem structures that are not treatable by the existing framework, such as problems with more or fewer constraints than variables. Finally, this thesis provides a comparison between the convergence properties of Fast-Lipschitz optimization and those of a traditional method based on gradient descent.

The thesis then applies the theory of Fast-Lipschitz optimization to the area of radio power control over wireless networks. Conditions are given under which well known distributed power control algorithms, such as those involving standard types of interference functions, fall in the Fast-Lipschitz framework. This gives a direct connection between these algorithms and a family of optimization problems, and the fixed points thereby assume a meaning of optimality.

Finally, the thesis gives illustrative examples of the new theory and examples of applications such as in a general non-convex optimal control problem and a non-monotonic power control problem.
First of all, I would like to thank my supervisor Carlo Fischione. Thank you for giving me the chance to be a PhD student, and for your continuous support in leading my research with constructive advice and feedback. Your energy and enthusiasm is inspiring.

This work has been improved through the feedback and suggestions of Chathuranga Weeraddana, Georgios Athanasiou, Themistoklis Charalambous, and Lazaros Gkatzikis. Thank you all for taking the time.

I also want to thank my past and present colleagues at the Department of Automatic Control. Interacting with you, whether it is through research, course work, traveling, or just lunch, makes the department an exiting place to work. I am particularly indebted to my fantastic room mates, always eager to help on topics ranging from \LaTeX error messages to electrical circuits. Furthermore, I want to thank Annelie, Hanna, Karin, and Kristina. You keep the office afloat and our spirits high.

This work has been made possible through the funding of the European Union’s FP7 projects ‘Hydrobionets’ and ‘NoE Hycon2’, and the Swedish Research Council projects ‘In Network Optimization’ and ‘SRA TNG’.

Last, but certainly not least, I want to thank my family. Your unconditional love and patient support mean the world to me!

Martin Jakobsson
Stockholm, August 2014
# Contents

Abstract iii  
Acknowledgements v  

1 Introduction 1  
1.1 Motivating examples .............................................. 2  
1.2 Problem formulation ............................................. 6  
1.3 Thesis outline and contribution ................................ 9  

2 Background 13  
2.1 Notation ............................................................... 13  
2.2 Distributed optimization ......................................... 14  
2.3 Radio power control .............................................. 15  
2.4 Fast-Lipschitz optimization ..................................... 18  

3 A New Set of Qualifying Conditions 21  
3.1 A general qualifying condition .................................... 21  
3.2 Contraction properties of Fast-Lipschitz problems ............ 23  
3.3 Proof of the general qualifying condition ....................... 24  
3.4 Special cases ....................................................... 29  
3.5 Summary ............................................................. 35  

4 Relaxations of the Fast-Lipschitz form 37  
4.1 Fast-Lipschitz minimization ...................................... 37  
4.2 Additional constraints ............................................. 38  
4.3 Fewer constraints than variables – Constant constraints .... 38  
4.4 Non-strictly monotonic objective function – Missing variables in the objective function ........................................... 41  
4.5 Summary ............................................................. 44
Currently, many engineering systems are interconnected by networks, i.e., they consist of many communicating subsystems (often called nodes or agents). Examples include wireless or wired communication networks, the national power grid, and groups of autonomous robots. In order to get the desired behavior of the network, the nodes must interact in some predetermined way. In many situations of substantial engineering interest, these interactions are governed by an optimization problem, where each node’s control variables are set in order to optimize a global (network-wide) objective.

When these networks are large, or when the subsystems are physically separated, the centralized solution of such optimization problems can be problematic. A centralized solution method would imply that all information relevant to the decision at hand must be sent to the central location. If the system is big, this might entail substantial information to handle and transfer. For example, such information might not fit in the node’s memory. Even if the amount of information is small enough to handle, there is always some cost associated to information gathering (e.g., in terms of data traffic, energy consumption, and delays). The controller must then, based on the gathered information, make appropriate decisions and communicate them back to the different subsystems (again, associated with some cost).

In a distributed solution method, each node calculates its own decision by executing a previously agreed upon iterative algorithm. The node will still need to exchange some information with the rest of the network, otherwise there would be no interconnection between the subsystems and the network-wide optimum would not be reached. However, in many situations of practical interest, only a small amount of information has to be exchanged, and only within some small neighborhood around the node. The required communication will therefore be much smaller than in the centralized solution method.

Finding an algorithm that quickly allows the computation of the network opti-
1. **Introduction**

minimum by distributed and local computations is challenging. This is the active and important research area of parallel and distributed optimization. The topic of this thesis is the framework of Fast-Lipschitz optimization, which is a particular case of distributed optimization introduced by C. Fischione in 2011 [1]. This framework considers particular problem structures, where the optimal solution can be found through simple distributed algorithms.

Before we discuss the problem formulation in greater detail, we present motivating examples where distributed optimization is needed.

### 1.1 Motivating examples

In this section we illustrate the importance of distributed optimization algorithms through general examples from wireless communication, sensor networks and big data analysis.

**Control algorithms in 5G communication networks**

During the last decades, wireless communication networks have become an integral part of our everyday life. There has been an exponential increase in transmission rates, from the order of 10 kbit/s in GSM (Global System for Mobile Communications, also known as second generation) networks to the order of 100 000 kbit/s in today’s fourth generation (4G) cellular networks [2]. Currently, mobile communication networks throughout the world, together handle more than one billion gigabytes of data every month [3].

However, the spectrum of radio frequencies is limited, and each frequency band only supports a limited amount of information transfer. In densely populated areas, interference from other users is the main limitation to performance. If the data rates and traffic volumes are to keep increasing, the upcoming next generation (5G) wireless communication technology will have to incorporate new solutions, such as highly localized base stations (e.g., covering only a building floor) and peer-to-peer communication among mobile devices such as smartphones.

Furthermore, the traffic in the network is also diversifying. Currently, the traffic in mobile networks consists mainly of voice and data applications. Voice call applications require a low but steady data rate, whereas data applications (e.g., streaming video) require as high data rates as possible but are less sensitive to data rate fluctuations. In addition, 5G networks will need to support new machine-to-machine applications. Such devices typically produce data at a low rate, but they require high reliability and low latency. A prime example of this is the project “LTE 4 Smart Energy”[1], where the goal is to optimize the use of 5G networks to monitor and control the smart energy grids of the future. An illustration of these concepts is given in Figure 1.1.

---

[1] The project “LTE 4 Smart Energy” is a collaboration between KTH, Ericsson and Siemens. It is partially funded by EIT ICT Labs and Wireless@kth.
1.1. Motivating examples

The heterogeneity of devices, applications and infrastructure increases both the scale and the complexity of 5G networks, which need to organize a diverse set of devices and fairly share the limited network resources among them. The scale of the network makes central coordination impossible, and hence new distributed methods need to be developed.

Cyber-physical systems and wireless sensor networks

A wireless sensor network consists of many small devices (nodes) spread out in an area, with the purpose of monitoring some environmental variables. For example, one could monitor the air quality in a mine, the humidity or temperature in a greenhouse, pollution levels in a river, or the presence of intruders in a restricted area [4].

The sensors are wireless and battery powered so that they can be placed in remote or hazardous locations, and cheap enough to deploy in large numbers. Because of the limitations on size and cost, individual sensors have a limited computational power, sensing range and accuracy. However, if the sensors cooperate with their nearby neighbors it is possible for the network as a whole to accurately cover the area of interest. For example, the sensors can increase accuracy by sharing measurements and current estimates, and reduce energy cost by aggregating and relaying...
Figure 1.2: Stockholm Royal Seaport is an urban development project consisting of 12000 new homes and 35000 new workspaces. Stockholm Royal Seaport has a strong environmental profile and when fully developed around 2030, this new part of Stockholm city will host many applications of networked systems, such as smart energy grids and smart home appliances. Image: Stockholms stad, http://bygg.stockholm.se.

Wireless sensor networks must be robust to changes in the environment or failure of individual nodes. Once the sensors are deployed they should organize themselves to create and maintain a network. This should not require pre-designated leader devices, because such leaders can become unavailable due to a change in the environment or device failure, thereby crippling the network. Instead, all nodes are peers and they collectively perform the computations necessary to organize themselves and perform their tasks. Furthermore, since the sensors are battery powered, the operational life span of the network crucially depends on the energy efficiency of the nodes. The most power intensive task of the nodes is communication. For these reasons, it is important to develop distributed algorithms that quickly converge to the correct solution.

A particular example of environmental and industrial process monitoring wireless sensor networks is investigated in the EU FP7 project Hydrobionets\(^2\), which monitors the microbial levels of water in large scale industrial plants that produce drinkable water. The project aims to develop novel algorithms combining different disciplines such as Networking, Signal Processing, Control and Biotechnology.

\(^2\)www.hydrobionets.eu
1.1. Motivating examples

Big data analysis

Big data is used to describe sets of data that are too large to be handled by traditional database and analysis techniques, such as least-squares estimation [5]. In our everyday life, big data forms an integral part of online services such as Google, Facebook, and Amazon. These services combine information supplied by the users, their contacts, recorded use of the service, among others, to provide personalized search queries, advertisements and product recommendations.

Big data appears also in natural sciences, where large amounts of data have to be analyzed to extract patterns, make forecasts, or validate models. For example Large Hadron Collider particle accelerator at CERN consists of about 150 million sensors that are capable of delivering data 40 million times per second [6]. Although only a fraction of this data is actually recorded, this amounts to one gigabyte of new data every second. Figure 1.3 illustrates the Square Kilometer Array (SKA), a planned radio telescope of unprecedented scale with a prototype stage planned 2016\(^3\). The system will consist of a large number of antennas (each antenna being a telescope on its own) spread out in an area. By carefully timing the measurements from each antenna (beam forming), SKA will examine several zones of the sky simultaneously. The number of beams and the field of view is essentially limited by signal processing and computing capacity, and SKA will need a supercomputer faster than the fastest computer available today. Once operational, it is estimated that SKA will output 10 gigabytes of processed data per second. Fast and distributed algorithms are

\(^3\)www.skatelescope.org
essential to efficiently analyze these big data sets.

1.2 Problem formulation

This thesis considers Fast-Lipschitz optimization, a specific class of problems where the structure simplifies the task of finding the optimal solution. This class of problems was sensed in [7] and formalized in [1]. In this thesis, we generalize the theory and provide new application results. In particular, we study problems of the form

$$\begin{align*}
\max & \quad f_0(x_1, \ldots, x_n) \\
\text{s.t.} & \quad x_i \leq f_i(x_1, \ldots, x_n), \quad \text{for } i = 1, \ldots, n.
\end{align*}$$

This problem is easily associated to a network of $n$ nodes. Each node $i$ controls a local variable $x_i$, subject to a local constraint $f_i$, with the goal of maximizing a global network utility $f_0$. If the local variables $x_i$ are collected as components of the vector $x$, and the local constraints $f_i(x)$ as components of the vector-valued function $f(x)$, the optimization problem can then be written as

$$\begin{align*}
\max & \quad f_0(x) \\
\text{s.t.} & \quad x \leq f(x).
\end{align*}$$

The essential result of Fast-Lipschitz optimization states that if the functions $f_0(x)$ and $f(x)$ fulfill certain qualifying conditions, then the optimal solution is the point $x^*$ given as the unique solution to the system of equations

$$x^* = f(x^*).$$

Furthermore, the qualifying conditions assure that $f(x)$ is contractive, whereby the iterations

$$x^{k+1} := f(x^k)$$

converge to the optimal point $x^*$ for any initial point $x^0$. This gives a remarkably easy and distributed method of calculating the optimal solution: At time instance $k + 1$, each node $i$ updates its local decision variable by computing its constraint function at the decision variables available at that time, i.e.,

$$x_i^{k+1} := f_i(x^k) = f_i(x_1^k, \ldots, x_n^k).$$

Note that $f_i(x)$ formally can depend on information $x_j$ from all other nodes. However, it is often the case that $f_i(x)$ only depends on the decision variables that are in the neighborhood of node $i$, or there is an oracle able to compute the function locally even though the decision variables of other nodes are not communicated.

We now end this section with two application examples of problem (1.1).
1.2. Problem formulation

Radio power control in wireless communication networks

Consider a wireless communication network, where several pairs of transmitters and receivers attempt to communicate. Suppose there are \( n \) links of transmitter-receiver pairs. Let \( p_i \) denote the transmit radio power of the transmitting node of link \( i \). An extensively applied approach to radio power control (see e.g., [8,9]) makes use of the gain matrix \( G \), where \( G_{ij} \) models the wireless channel attenuation from transmitter \( j \) to receiver \( i \) (see Figure 1.4). In addition to the useful signal \( G_{ii}p_i \) from transmitter \( i \), receiver \( i \) will also experience background noise \( \eta_i \) plus the interference of all other transmitters, \( \sum_{j \neq i} G_{ij}p_j \). For the communication attempt of transmit node \( i \) to be successful, the signal-to-interference-and-noise ratio (SINR) at receiver node \( i \) must be higher than some threshold \( \tau_i \),

\[
\frac{G_{ii}p_i}{\sum_{j \neq i} G_{ij}p_j + \eta_i} \geq \tau_i. \tag{1.4}
\]

If the transmit power of all links are collected in the vector \( p = [p_1, p_2, \ldots, p_n]^T \), the requirement (1.4) can be rewritten as

\[
p_i \geq \beta_i(p) \triangleq \frac{\tau_i}{G_{ii}} \left( \sum_{j \neq i} G_{ij}p_j + \eta_i \right). \tag{1.5}
\]

The function \( \beta_i(p) \) is called the interference function of transmitter \( i \), and sets the lowest transmit power that transmitter \( i \) can use while maintaining an acceptable SINR. From inequality (1.4) it is clear that a higher transmit power \( p_i \) will benefit the SINR of link \( i \), while it decreases the SINR of all other links \( j \neq i \). A higher transmit power also leads to increased energy consumption, which is a problem particularly when the transmitters are battery powered. A natural problem for

\[\textsuperscript{4}\text{We give a formal introduction of our notation in Section 2.1}\]
the system designer is therefore to minimize the power of each transmitter, while maintaining acceptable SINR in all links:

\[
\begin{align*}
\min_{\mathbf{p}} & \quad \mathbf{p} \\
\text{s.t.} & \quad p_i \geq \beta_i(\mathbf{p}) \quad i = 1, \ldots, n.
\end{align*}
\]

(1.6)

The problem above is a vector optimization problem, i.e., each element of the vector \( \mathbf{p} \) is an objective to be minimized. We give a brief introduction to the concepts of vector optimization in Appendix A. By introducing the new variable \( \mathbf{x} \triangleq -\mathbf{p} \), and the new functions \( f_0(\mathbf{x}) \triangleq x \) and \( f_i(\mathbf{x}) \triangleq -\beta_i(-\mathbf{x}) \), for \( i = 1, \ldots, n \), problem (1.6) is equivalent to

\[
\begin{align*}
\max_{\mathbf{x}} & \quad f_0(\mathbf{x}) \\
\text{s.t.} & \quad x_i \leq f_i(\mathbf{x}) \quad i = 1, \ldots, n.
\end{align*}
\]

(1.7)

This is a problem of the form (1.1), and when the functions \( f_0(\mathbf{x}) \) and \( f_i(\mathbf{x}) \) fulfill the qualifying conditions of Fast-Lipschitz optimization, the solution can be found by the iterations (1.3). We will discuss this and related problems further in Section 2.3 and Chapter 6.

Distributed norm condition checking

The paper [7] considers a network of \( n \) sensors that track a common time-varying scalar quantity \( d(t) \), where \( t \) indexes discrete time steps. Each sensor \( i \) makes local measurements and maintains a local estimate \( x_i(t) \). When the estimate \( x_i(t) \) is updated, node \( i \) uses a weighted combination of its own estimate and the estimates of its neighbors,

\[
\sum_{j=1}^{n} k_{ij} x_j(t - 1).
\]

Each node \( i \) therefore maintains a row vector \( \mathbf{k}_i = [k_{i1}, \ldots, k_{in}] \) (where \( k_{ij} = 0 \) if node \( i \) and \( j \) are not communicating neighbors). Let \( \mathbf{K} \) be the matrix with rows \( \{\mathbf{k}_i\}_{i=1}^{n} \). The analysis of [7] requires (for the estimates to converge) that the matrix norm \( \|\mathbf{K}\|_2 \) is less than one. This is a centralized condition, i.e., in order to check it one needs to know the row vectors \( \mathbf{k}_i \) of every node in the network. It is possible to bound

\[
\|\mathbf{K}\|_2 \leq \sqrt{n} \max_{i} \|\mathbf{k}_i\|_1 = \sqrt{n} \max_{i} \sum_{j=1}^{n} |k_{ij}|,
\]

but [7] argues that this is overly conservative, resulting in unnecessary low values of \( k_{ij} \) and bad performance. Instead, [7] introduces a distributed way to bound \( \|\mathbf{K}\|_2 \) as follows:
1.3 Thesis outline and contribution

**Proposition 1.1** ([7]). Let $\theta_i$ be the collection of nodes located at most two hops from node $i$. Suppose there exists a vector $y = [y_1, \ldots, y_n]^T > 0$ such that

$$y_i + \sqrt{y_i} \sum_{j \in \theta_i} \sqrt{y_j} \leq \gamma$$

for all $i = 1, \ldots, n$. If $\|k_i\|^2 \leq y_i$ for all $i$, then $\|K\|_2 < \gamma$.

It is clearly desirable to choose the $y_i$ as large as possible. One way to find such a vector $y$ is solving the distributed optimization problem

$$\max_y \sum_{i=1}^n y_i \quad \text{s.t.} \quad y_i + \sqrt{y_i} \sum_{j \in \theta_i} \sqrt{y_j} \leq \gamma, \; i = 1, \ldots, n.$$ 

Define the functions $f_0(y) \triangleq \sum_{i=1}^n y_i$ and $f_i(y) \triangleq \gamma - \sqrt{y_i} \sum_{j \in \theta_i} \sqrt{y_j}$. The optimization problem above can be thus be written

$$\max_y f_0(y) \quad \text{s.t.} \quad y_i \leq f_i(y), \; i = 1, \ldots, n,$$

which is a problem of the form (1.1). Therefore, provided that $f_0$ and $f_i$ fulfill the qualifying conditions, the problem can be solved if each node iterates

$$y_i^k = f_i(y_i^{k+1}, \ldots, y_n^{k+1})$$

as in the iterations (1.3). Note that the functions $f_i(y)$ do not require information from all other nodes of the network, but only the values of $y_j$ from the two-hop neighbors $j \in \theta_i$ of node $i$.

1.3 Thesis outline and contribution

This thesis is part of an ongoing effort to expand the Fast-Lipschitz framework, presented in the previous section. The following chapters will examine and loosen the previously known qualifying conditions and the requirements on the Fast-Lipschitz form, with the hope that more problems can be effectively solved with the framework. Furthermore, the thesis attempts to theoretically compare the convergence speeds when solving a problem with the Fast-Lipschitz framework and a traditional method based on dual variables. Finally, the thesis gives several examples of Fast-Lipschitz optimization. In particular, the framework is applied to the area of radio power control, and to solve a generic optimal control example.

This section outlines the contents of the thesis and lists the major contributions, which are based on the papers

1. Introduction


The thesis should be considered as a collection of papers, although the material has been rearranged and formatted as follows:

Chapter 2: Background

In this chapter we discuss related literature and present existing results on Interference function optimization and Fast-Lipschitz optimization.

Chapter 3: A New Set of Qualifying Conditions

This chapter is mainly based on [A]. It presents a new qualifying condition GQC that unifies and extends the existing conditions of [1]. A new proof is given based on duality and the KKT conditions. We discuss the general qualifying condition GQC and its relation to existing literature through the special cases Q₁⁻Q₆. Conditions Q₁⁻Q₅ appear in [A] while Q₆ appears in [D]. Preliminary versions of conditions Q₁⁻Q₄ and Q₆ have been published in [B].

Chapter 4: Relaxations of the Fast-Lipschitz form

In this chapter we present several relaxations of the required form of a Fast-Lipschitz problem. In section 4.1, we show that the qualifying conditions remain unchanged when switching between maximization and minimization problems (appears in [D]). Furthermore, we examine problems where the number of variables does not equal the number of constraints (as is required by the Fast-Lipschitz form). These results appear in [A]. Problems with fewer variables than constraints are treated in sections 4.2 and 4.4 while problems with more variables than constraints are treated in Section 4.3.

Chapter 5: Convergence Analysis

This chapter is based on [C] and compares the convergence rates of Fast-Lipschitz optimization to a first order Lagrangian method. The analysis considers a set of problems that can be solved by both methods, and bound the maximal and minimal
1.3. Thesis outline and contribution

Convergence speed of the Lagrangian method and the Fast-Lipschitz method respectively. This gives a sufficient condition for the Fast-Lipschitz method to outperform the traditional Lagrangian method.

Chapter 6: Transmit Radio Power Control

This chapter is based on [D], and relates Fast-Lipschitz optimization to existing methods in power control. In Section 6.2.2 we show that optimization problems with monotonic cost functions and standard constraints are contained in the Fast-Lipschitz framework. In Section 6.2.3 we give conditions for problems with type-II standard constraints to be Fast-Lipschitz. This allows to give solutions of certain type-II standard power control algorithms a notion of optimality, whereas existing theory only shows convergence. Finally, Section 6.3 applies Fast-Lipschitz theory to solve a transmit power control problem where the interference function is non-monotonic, i.e., neither standard, nor type-II standard.

Chapter 7: Illustrative Examples

In this chapter we apply the theory described in chapters 3–6. Section 7.1 illustrates the idea of Fast-Lipschitz optimization and the qualifying conditions through a simple two-dimensional example. Section 7.2 applies and the convergence results from Chapter 5 to a numerical example. Finally, Section 7.3 applies the theory of chapters 3 and 4 to solve a generic optimal control example.

Sections 7.1 and 7.3 appears in [A], while Section 7.2 appears in [C].

Chapter 8: Conclusions and Future Work

The thesis is concluded and future research directions are suggested.
This chapter establishes the required background for the rest of the thesis. Section 2.1 clarifies the notation, and Section 2.2 provides some background on distributed optimization. Section 2.3 introduces the topic of radio power control and Section 2.4 gives the formal definition of Fast-Lipschitz optimization.

2.1 Notation

Vectors and matrices are denoted by bold lower and upper case letters, respectively. The components of a vector \( \mathbf{x} \) are denoted \( x_i \) or \( [\mathbf{x}]_i \). Similarly, the elements of the matrix \( \mathbf{A} \) are denoted \( A_{ij} \) or \( [\mathbf{A}]_{ij} \). The transpose of a vector or matrix is denoted \( \cdot^T \). \( \mathbf{I} \) and \( \mathbf{1} \) denote the identity matrix and the vector of ones respectively. A vector or matrix where all elements are zero is denoted by \( \mathbf{0} \).

The gradient \( \nabla f(\mathbf{x}) \) is the transpose of the Jacobian matrix, i.e.,

\[
[\nabla f(\mathbf{x})]_{ij} = \frac{\partial f_j(\mathbf{x})}{\partial x_i},
\]

whereas \( \nabla_i f(\mathbf{x}) \) denotes the \( i \)th row of \( \nabla f(\mathbf{x}) \). If the variable is partitioned as \( \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{z}^T \end{bmatrix} \), then \( \nabla_\mathbf{y} f(\mathbf{x}) \) denotes the partial derivatives of \( f(\mathbf{y}, \mathbf{z}) \) with respect to \( \mathbf{y} \). Note that \( \nabla f(\mathbf{x})^k = (\nabla f(\mathbf{x}))^k \), which should not be confused with the \( k \)th derivative.

If \( \mu_i(\mathbf{A}) \) are the eigenvalues of a \( n \times n \) matrix \( \mathbf{A} \), they are ordered in descending order, \( |\mu_1| \geq |\mu_2| \geq \cdots \geq |\mu_n| \). The spectral radius of \( \mathbf{A} \) is denoted \( \rho(\mathbf{A}) \).

Vector norms are denoted \( ||\cdot|| \) and matrix norms are denoted \( \|\cdot\| \). Unless specified, \( ||\cdot|| \) and \( \|\cdot\| \) denote arbitrary norms. \( \|\mathbf{A}\|_\infty = \max_i \sum_j |A_{ij}| \) is the norm induced by the \( \ell_\infty \) vector norm.

All inequalities are intended element-wise, i.e., \( \mathbf{A} \geq \mathbf{B} \) means \( A_{ij} \geq B_{ij} \) for all \( i, j \). We will also use the element-wise natural logarithm \( \ln \mathbf{x} = [\ln x_1, \ldots, \ln x_n]^T \) and the element-wise exponential \( e^\mathbf{x} = [e^{x_1}, \ldots, e^{x_n}]^T \).
2. Background

Remark. The notation $\beta$ for interference functions does not follow the notational assumptions above. However, we will keep such notation to harmonize with existing literature on radio power control, e.g., [8–11].

2.2 Distributed optimization

In a general networked optimization problem, the network nodes or agents must coordinate their actions to optimize a network-wide objective function. When information such as nodes’ objectives, constraints and decision variables are distributed among the nodes and physically scattered across the network, or if the amount of information is too large to collect centrally, it can be impractical or even impossible to centrally compute the solution. For example, collecting information in one place might be too expensive if the network has limited communication resources, or it may be too slow if the solution is needed at the local nodes in real time. In these situations, fast distributed solution algorithms must be used.

Distributed optimization has a long history, and much of the recent developments build upon the work of Tsitsiklis [12,13]. Several approaches exist for solving these problems, such as primal and dual decomposition methods. In these methods, the primal or the dual problem is decomposed into local subproblems solved at the nodes. The subproblems are coordinated through a centralized master problem, which is usually solved by gradient or subgradient methods [14]. These methods have found many applications in network utility maximization, e.g., [15–17]. Due to the slow convergence of subgradient algorithms, recent works have explored higher order methods. For example, [18] replaces the subgradient step with a Newton-like step. An other decomposition approach which is faster and more robust than the standard decomposition methods is the alternating direction method of multipliers (ADMM). The method has recently attracted a substantial interest, especially for problems with large data sets [19,23].

Although the decomposition methods mentioned above distribute the computational workload over the network nodes, some coordination is still required for the solution of the subproblems. For example, dual decomposition based methods must update dual variables in a central master problem. This requires the network to iteratively 1) transfer information from all nodes to some central “master” node; 2) centrally update the dual variables; 3) broadcast the updated dual variables to all nodes of the network and back to 1) until convergence.

To avoid the centralized master node, peer-to-peer, or multi-agent, methods have been proposed to coordinate the subproblem through local neighbor interactions based on consensus algorithms, e.g., [24,29]. In these algorithms, nodes update their decision variables as convex combinations of the decision variables of their neighbors without a central master node or fusion center. In [30], the consensus algorithm has been combined with gradient descent to solve an unconstrained optimization problem where the objective is a sum of local, convex functions. The work is extended in [31,32], which investigate constraints and randomness. While the
previous papers solve the primal problem,\cite{33,35} use consensus-based algorithms for the dual problem. Higher order methods are considered also for peer-to-peer optimization, e.g.,\cite{36} solves an unconstrained primal problem, whereas\cite{37} solves a linearly constrained dual problem, both by approximating Newton’s algorithm through consensus.

Consensus based methods have many benefits, such as resilience to node failures and changing network topology. However, since every round of consensus requires message passings, these methods may suffer from communication overheads. A recent study of the tradeoff between communication and local computation can be found in\cite{38}. Communication overhead is a problem especially in large scale distributed networks or wireless sensor networks, where the energy expenditure for communication can be orders of magnitude larger than the energy for computation\cite{1}.

The methods discussed thus far assume convex problems. There are other classes of algorithms that do not rely necessarily on convexity, but on other structural properties. Three such classes are abstract optimization\cite{39}, which generalizes linear optimization, monotonic optimization\cite{40,42}, where the monotonicity of the objective function is used to iteratively refine a solution within bounds of the feasible region, and Interference Function optimization\cite{8,10,43,44}, which is the fundamental framework to solve radio power control problems over wireless networks. Given the importance of Interference Function optimization as a precursor of Fast-Lipschitz optimization, and considering that we will give some application examples later on in this thesis, we give below some technical details on such an optimization framework.

\section{Radio power control}

Radio power control is one of the essential radio resource management techniques in wireless networks. The power control problem faces a tradeoff between saving power and achieving sufficient communication rates. It is important to control the transmit radio powers to avoid interferences to undesired receivers and save the energy of the transmitters. Meanwhile, it is important also to use adequate levels of power to make sure that the transmitted signals can overcome the attenuation of the wireless channel and the interference caused by other transmitters.

Power control in wireless communication is a particularly successful instance of distributed optimization over networks. Specifically, in wireless networks a link is associated to one pair of nodes where a node is a transmitter and the other is a receiver. Suppose there are \( n \) transmitter-receiver pairs. Let \( p_i \) be the radio power of transmit node \( i \), for \( i = 1, \ldots, n \). Note that index \( i \) is used both for a transmitter and a receiver, so transmitter \( i \) and receiver \( i \) are two different nodes that are paired to communicate. A common approach to radio power control considers the gain matrix \( \mathbf{G} \), where \( G_{ij} \) is the channel attenuation from the transmit node \( j \) to the receiver node \( i \). In addition to the useful signal \( G_{ii}p_i \) from transmitter \( i \), receiver \( i \)
2. Background

will also receive a background noise \( \eta_i \) plus the interference of all other transmitters, \( \sum_{j \neq i} G_{ij} p_j \). For the communication attempt of transmit node \( i \) to be successful, the signal-to-interference-and-noise ratio (SINR) at receiver node \( i \) must be higher than some threshold \( \tau_i \),

\[
\frac{G_{ii} p_i}{\sum_{j \neq i} G_{ij} p_j + \eta_i} \geq \tau_i, \quad (2.1)
\]
or

\[
p_i \geq \beta_i(p) \triangleq \frac{\tau_i}{G_{ii}} \left( \sum_{j \neq i} G_{ij} p_j + \eta_i \right). \quad (2.2)
\]

Using the terminology of \[9\], we will refer to \( \beta_i(p) \) as the interference function of transmitter \( i \). This affine version of \( \beta_i(p) \) is the simplest and best studied type of interference function, and it is often the basis for extensions or modifications by other types of interference functions. The focus on achieving a sufficient SINR in \( (2.1) \) is justified since several measures of the quality of service are increasing functions of the SINR \[45\].

There are a number of ways of using the interference function in setting power control optimization problems:

- maximization of the SINR (i.e., quality of service) of the network, subject to power constraints;
- minimization of the power consumption subject to SINR constraints;
- maximization of some network utility function (e.g., throughput) of the network, subject to power constraints.

Early works on distributed power control in wireless networks have followed the first approach, and try to maximize the smallest signal-to-interference ratio of the network \[10,43\] (these works did not consider noise). With the inclusion of receiver noise in \[8\], the focus has shifted to the second approach, with the goal of minimizing the radio powers \( p_i \) while maintaining a minimum SINR \( \tau_i \) at each receiver, i.e.,

\[
\min_P p \quad \text{s.t.} \quad p_i \geq \beta_i(p) \quad \forall i \quad (2.3)
\]

When \( \beta_i(p) \) is given by equation \( (2.2) \), the Foschini-Miljanic algorithm of \[8\] solves problem \( (2.3) \) by iterating

\[
p_i^{k+1} := \beta_i(p^k). \quad (2.4)
\]

In vector form we write \( p^{k+1} := \beta(p^k) \), where

\[
\beta(p^k) \triangleq \begin{bmatrix} \beta_1(p^k) & \ldots & \beta_n(p^k) \end{bmatrix}^T.
\]

Given that the channel gains \( G_{ij} \) allows for a feasible solution, the iterates \( p^k \) will converge to a point \( p^* = \beta(p^*) \) that is optimal for problem \( (2.3) \).
2.3. Radio power control

The computation of the optimal solution for problem (2.3) by these iterations is much simpler than using the classical parallelization and decomposition methods of distributed optimization [46]. This is because there is no longer a need to centrally collect, compute and redistribute the primal and dual variables of the problem since \( \beta_i(p^k) \) can be known or estimated locally at receiver \( i \) [8,9]. Even in a centralized setting, iteration (2.4) is simpler than traditional distributed optimization methods, since no dual variables need to stored be and manipulated. The iterations require only that every receiver node successively updates using local knowledge of the function (interference function) of other nodes’ current decision variables (radio powers). Another advantage is that convergence is obtained even though such a knowledge is delayed, i.e., when the decision variables \( p_j^k \) of other nodes are updated with some delay [46,47].

This line of work has later been generalized to the framework of standard interference functions by Yates [9]. The function \( \beta(p) \) is standard if for all \( p, q \geq 0 \), the following properties are satisfied:

* **Positivity:** \( \beta(p) > 0 \),
* **Monotonicity:** \( p \geq q \Rightarrow \beta(p) \geq \beta(q) \),
* **Scalability:** \( c > 1 \Rightarrow \beta(cp) < (1/c)\beta(p) \).

When problem (2.3) above is feasible and \( \beta(p) \) is standard, the iterations

\[
p_i^{k+1} := \beta_i(p^k)
\]

(2.5)

converge to the fixed point \( p^* = \beta(p^*) \), and \( p^* \) is the unique optimal solution to problem (2.3) [9].

Problem formulation (2.3) makes sense when there is no additional value in achieving a higher SINR than the required level \( \tau \) (see equation (2.1)). This is the case for traditional voice applications, but it is no longer true in heterogeneous networks with voice and data applications. If one, instead of problem (2.3), tries to optimize the collective throughput of the system it turns out that the optimal strategy is to only transmit on the link with the best channel. This idea is used in opportunistic approaches, where mobiles decrease their transmit power when channel conditions worsen (see, e.g., [48–50]).

When mobiles follow an opportunistic power algorithm \( p^{k+1} := \beta(p^k) \), the update function \( \beta(p) \) no longer remains monotonically increasing, and the standard framework no longer applies. This is addressed in [49], where Sung and Leung complements Yates’ framework with the highly related type-II standard functions. A function \( \beta(p) \) is said to be a type-II standard interference function if for all \( p, q \geq 0 \), the following properties are satisfied:

* **Type-II Monotonicity:** \( p \leq q \Rightarrow \beta(p) \geq \beta(q) \),
* **Type-II Scalability:** \( c > 1 \Rightarrow \beta(cp) > (1/c)\beta(p) \).

Type-II standard functions converge in the same way as standard functions: if a fixed point \( p^* \) exists, then iteration (6.3) converges to \( p^* = \beta(p^*) \) [49 Theo. 3].
However, the type-II standard framework no longer provides statements about the optimality of $p^\ast$. In fact, the $\beta(p)$ considered is an update rule for the transmit powers and there is no longer a relation to problem (2.3).

Further work related to the interference function frameworks above has been presented in [51] and [52]. Specifically, [51] considers general interference functions, which replaces the scalability ($\beta(cp) < c\beta(p)$ for $c > 1$) of the standard framework with scale invariance, i.e., $\beta(cp) = c\beta(p)$. [51] provides a rigorous analysis of these general interference functions and show that they are level curves of closed comprehensive sets. A fixed point iteration similar to (2.5) is also shown to converge under some additional technical assumptions. Similarly, [52] considers contractive interference functions that replaces Yates’ scalability assumption with a weighted max-norm contractivity requirement. This allows to establish existence of fixed points and bound rates of convergence for the iterations (2.5).

The above works on power control have in common that the assumptions allow efficient solutions through fixed point iterations. However, focus is mainly on the convergence of the fixed point algorithm (2.5), and not the meaning of the fixed point in terms of optimality. This can be seen as a motivation for Fast-Lipschitz optimization, which is the topic of this thesis. Fast-Lipschitz optimization can be seen as an attempt to generalize the interference function approach, and classify the optimization problems that are easily solvable through fixed point iterations analogous to (2.4). The following section gives a formal definition of these problems.

### 2.4 Fast-Lipschitz optimization

We will now give a formal definition of Fast-Lipschitz problems, which was introduced in [1,53]. This provides the foundation for the further analysis and extensions of this thesis. For a thorough discussion of Fast-Lipschitz properties we refer the reader to the previously mentioned papers.

**Definition 2.1.** A problem is said to be on *Fast-Lipschitz form* if it can be written

$$\begin{align*}
\max_{x} & \quad f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x) \quad \forall i \in A \\
& \quad x_i = f_i(x) \quad \forall i \in B,
\end{align*}$$

where

- $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a differentiable scalar ($m = 1$) or vector-valued ($m \geq 2$) function,

- $A$ and $B$ are complementary subsets of $\{1, \ldots, n\}$, i.e., $A \cup B = \{1, \ldots, n\}$ and $A \cap B = \emptyset$,

- $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ are differentiable functions.
2.4. Fast-Lipschitz optimization

We will refer to problem (2.6) as our main problem. From the individual constraint functions we form the vector-valued function \( f : \mathbb{R}^n \to \mathbb{R}^n \) as

\[
f(x) = [f_1(x) \cdots f_n(x)]^T.
\]

**Remark 2.2.** The characteristic feature of Fast-Lipschitz form is a pairing such that each variable \( x_i \) is associated to one constraint \( f_i(x) \). The form \( x \leq f(x) \) is general, since any constraint on canonical form, \( g(x) \leq 0 \), can be written \( x \leq x - \gamma g(x) \) for some positive constant \( \gamma \).

**Definition 2.3.** We will restrict our attention to a bounding box

\[
D = \{ x \in \mathbb{R}^n \mid a \leq x \leq b \}.
\]

We assume \( D \) contains all candidates for optimality and that \( f \) maps \( D \) into \( D \), \( f : D \to D \). This box arise naturally in practice, since any real-world quantity or decision must be bounded.

**Definition 2.4.** A problem is said to be Fast-Lipschitz when it can be written in Fast-Lipschitz form and admits a unique Pareto optimal solution \( x^* \), defined as the unique solution to the system of equations

\[
x^* = f(x^*).
\]

A problem written in Fast-Lipschitz form is not necessarily Fast-Lipschitz. The following qualifying conditions guarantee that a problem in Fast-Lipschitz form is Fast-Lipschitz.

**Old Qualifying Conditions.** For all \( x \) in \( D \), \( f_0(x) \) and \( f(x) \) should fulfill at least one of the following cases (e.g., (0) and (i), or (0) and (ii)):

\[
(0) \quad \nabla f_0(x) > 0
\]

AND

\[
(i.a) \quad \nabla f(x) \geq 0
\]

\[
(i.b) \quad \|\nabla f(x)\| < 1
\]

OR

\[
(ii.a) \quad f_0(x) = c1^T x, \text{ with } c > 0
\]

\[
(ii.b) \quad \nabla f(x) \leq 0 \quad \text{(or more generally, } \nabla f(x)^2 \geq 0)\]

\[
(ii.c) \quad \|\nabla f(x)\|_\infty < 1
\]

OR

\[
(iii.a) \quad f_0(x) \in \mathbb{R}
\]

\[
(iii.b) \quad \|\nabla f(x)\|_\infty < \frac{\bar{\delta}}{\bar{\delta} + \Delta}, \text{ where } \bar{\delta} \triangleq \min_i \min_{x \in D} \nabla_i f_0(x), \quad \bar{\Delta} \triangleq \max_i \max_{x \in D} \nabla_i f_0(x)
\]

**Theorem 2.5** (\([\text{I]}\)). A problem in Fast-Lipschitz form that fulfills any pair of the Old Qualifying Conditions is Fast-Lipschitz, i.e., it has a unique Pareto optimal point given by \( x^* = f(x^*) \). Furthermore, \( x^* \) can be found as the limit of the iterations \( x^{k+1} = f(x^k) \).
Once it is known that a problem is Fast-Lipschitz, computing the solution becomes a matter of solving the system of equations (2.7), which in general is much easier than solving an optimization problem using Lagrangian multipliers. This is particularly evident when \( f(x) \) is contractive on \( D \), a property assured by the qualifying conditions. In this case, the iterations \( x^{k+1} := f(x^k) \) converge geometrically to the optimal point \( x^* \), starting from any initial point \( x^0 \in D \).

Remark 2.6. Note that the notation \( x^{k+1} := f(x^k) \) implies synchronous updates, i.e., that the \( i \)th node may not update

\[
x_i^{k+1} = f_i(x_1^k, \ldots, x_n^k)
\]

until all other nodes have computed and transmitted the \( k \)th update. However, the result holds true also in an asynchronous setting, where node \( i \) sometimes uses outdated information from other nodes. The natural assumption required is that the delays for all variables stay bounded, and the result (detailed in [1]) follows from the asynchronous convergence theorem of [46]. For the remaining parts of this thesis we will keep the clean notation for the synchronized case, although all qualifying conditions presented also imply asynchronous convergence.

Fast-Lipschitz optimization does not rely on convexity assumptions for the functions \( f_0(x) \) and \( f(x) \). Global optimality is assured through the monotonic and contractive properties required by the qualifying conditions. This means that Fast-Lipschitz problems need not be convex, but convex problems that fulfill the qualifying conditions can be rewritten and solved with the Fast-Lipschitz framework. Three examples of such problems are given in Sections 6.3, 7.1 and 7.3.

In this chapter we discussed distributed optimization in general, and the particular area of radio power control. In analogy to these radio power control problems, we formally defined the more general framework of Fast-Lipschitz optimization. The remaining chapters of this thesis will further develop and apply this theory, starting, in the following chapter, with a generalization of the qualifying conditions.
Chapter 3

A New Set of Qualifying Conditions

This chapter presents a new qualifying condition, which generalizes and unifies the conditions of Section 2.4. The new condition is introduced in Section 3.1, together with Theorem 3.1, which formally states the role of the condition. Section 3.2 shows that problems fulfilling the general qualifying condition of Section 3.1 have contractive constraints. These results enable finding the optimal point through fixed point iterations, and they also constitute a preliminary result to the proof of Theorem 3.1 in Section 3.3.

From this point and onwards we will call each set of related conditions a “Qualifying Condition”, rather than a “case” of the qualifying conditions. We will still use the notion of “case” when referring to the Old Qualifying Conditions (2.8). For example, “case (i)” will refer to the groups (2.8a)-(2.8c) of the old qualifying conditions, while “case (ii)” refers to (2.8a) and (2.8d)-(2.8f).

3.1 A general qualifying condition

Consider once again optimization problem (2.6) in Fast-Lipschitz form, surrounded by the bounding box $D$. Just as in Section 2.4, the qualifying conditions presented in this section ensure that a problem in Fast-Lipschitz form also is Fast-Lipschitz. In the upcoming qualifying conditions, we will use the ratio

$$q(x) \triangleq \min_j \frac{\min_i [\nabla f_0(x)]_{ij}}{\max_i [\nabla f_0(x)]_{ij}}.$$  \quad (3.1)

The value of $q(x)$ is the smallest ratio of any two elements from the same column of $\nabla f_0(x)$. When $\nabla f_0(x) \geq 0$, $q(x)$ is always non-negative. Furthermore, $q(x) \leq 1$ by construction, with equality if and only if all rows of $\nabla f_0(x)$ are identical. In fact,
3. A New Set of Qualifying Conditions

$q(x)$ can be seen as a penalty for when the objective function gradient points in a different direction than the vector 1.

We are now ready to state the new qualifying conditions. We will begin with the most general form of the qualifying conditions. Special cases of this condition will be discussed in Section 3.4.

<table>
<thead>
<tr>
<th>GQC</th>
<th>General Qualifying Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(GQC.a)</td>
<td>$\nabla f_0(x) \geq 0$ with non-zero rows</td>
</tr>
<tr>
<td>(GQC.b)</td>
<td>$</td>
</tr>
<tr>
<td>There exists a $k \in {1, 2, \ldots} \cup \infty$ such that</td>
<td></td>
</tr>
<tr>
<td>(GQC.c)</td>
<td>When $k &lt; \infty$, then $\nabla f(x)^k \geq 0$</td>
</tr>
<tr>
<td>(GQC.d)</td>
<td>When $k &gt; 1$, then $</td>
</tr>
</tbody>
</table>

In the condition above, we allow the parameter $k$ to be any positive integer, or infinity. Each value of $k$ corresponds to a different case in the proof of Theorem 3.1 below. In the extremes $k = \infty$ and $k = 1$, there is no need to fulfill conditions (GQC.c) and (GQC.d) respectively. Both these cases will be discussed further in Section 3.4.

We now give Theorem 3.1, which plays a role analogous to that of Theorem 2.5 in Section 2.4.

**Theorem 3.1.** Assume problem (2.6) is feasible, and that GQC holds for every $x \in D$. Then, the problem is Fast-Lipschitz, i.e., the unique Pareto optimal solution is given by $x^* = f'(x^*)$.

**Proof.** The theorem is proved in Section 3.3.

**Remark 3.2.** It should be emphasized that the qualifying conditions are only sufficient, i.e., a problem that does not fulfill the qualifying conditions can still be Fast-Lipschitz. For example, by considering certain transformations of the constraint functions, it is possible to relax condition the condition that contains the norm $|||\cdot|||_\infty$. One can also relax certain requirements on the Fast-Lipschitz form of problem (2.6), e.g. by considering problems with more variables than constraints or problem where the objective function only depends on a subset of the variables (see Chapter 4).
3.2 Contraction properties of Fast-Lipschitz problems

In this subsection we discuss the contractiveness of $f(x)$. This is important since it allows a Fast-Lipschitz optimization problem to be solved through fixed point methods. These results will also be used in the proof of Theorem 3.1 in the next subsection.

Once optimization problem (2.6) is shown to be Fast-Lipschitz, solving it becomes a matter of finding the point $x^* = f(x^*)$. In a centralized setting, one can use any suitable method for solving such a system of equations, e.g., Newton like methods. However, if $f(x)$ is contractive, the simplest way to solve $x^* = f(x^*)$ is to repeatedly evaluate $f(x)$. This method works both in a centralized and a distributed setting. Compared to other distributed methods, it has the benefit of being totally distributed, i.e., there is no master problem or coordinating node. Furthermore, the iterations converge even when some of the nodes, due to dropped or delayed packets, only have access to outdated information of the neighbors’ decisions, see [1, Proposition 3.6] or [46] for details. We show below that if the general qualifying condition GQC holds, then $f(x)$ is contractive.

Proposition 3.3 (Sec. 3.1 in [46]). Let $f$ be a mapping from a closed subset of $\mathbb{R}^n$ into itself, $f : \mathcal{X} \to \mathcal{X}$. If there is a norm $\| \cdot \|$ and a scalar $\alpha < 1$ such that $\|f(x) - f(y)\| \leq \alpha \|x - y\|$ for all $x, y \in \mathcal{X}$, then $f(x)$ is a contraction mapping.

As a result

- $x^* = f(x^*)$ is the unique fixed point of $f$ in $\mathcal{X}$.
- For every initial point $x^0 \in \mathcal{X}$, the sequence $x^{k+1} := f(x^k)$ converges linearly to $x^*$.

Since we know $f : D \to D$, $f$ is a contraction mapping if we can find a vector norm such that $\|f(x) - f(y)\| \leq \alpha \|x - y\|$ for all $x, y \in D$.

Lemma 3.4. If the general qualifying condition GQC holds, $f(x)$ is contractive.

Proof. Parameterize the line between $x, y \in D$ by

$$u(t) = tx + (1 - t)y, \quad 0 \leq t \leq 1,$$

and let $g(t) = f(u(t))$. Then,

$$\frac{dg(t)}{dt} = \nabla f(u(t))^T(x - y),$$

wherefore

$$f(x) - f(y) = g(1) - g(0) = \int_0^1 \frac{dg(t)}{dt} dt = \int_0^1 \nabla f(u(t))^T dt (x - y) \triangleq A(x - y),$$

23
3. A New Set of Qualifying Conditions

where each element in $A$ is the integral of the corresponding element in $\nabla f(u(t))$. Let $\|\cdot\|_a$ be the matrix norm that satisfies condition (GQC.b) and define $\|\cdot\|_b$ such that $\|A\|_b \triangleq \|A^T\|_a$. It is straightforward to show that $\|\cdot\|_b$ inherits the matrix norm properties of $\|\cdot\|_a$.\(^1\) We can now bound

$$\|A\|_b = \left\| \int_0^1 \nabla f(u(t))^T \, dt \right\|_b \leq \int_0^1 \left\| \nabla f(u(t))^T \right\|_b \, dt \leq \int_0^1 \max_{z \in D} \left\| \nabla f(z)^T \right\|_b \, dt = \max_{z \in D} \left\| \nabla f(z)^T \right\|_b = \max_{z \in D} \left\| \nabla f(z) \right\|_a \triangleq \alpha.$$\

The first inequality above is the triangle inequality. The second inequality holds since $D$ is convex whereby $u(t) \in D$ for all $t$. The maximum exists since $D$ is compact, and $\alpha < 1$ by (GQC.b).

From [54, Theorem 5.6.26], we know that there exists an induced matrix norm $\|\cdot\|_V$ such that $\|M\|_V \leq \|M\|_a$ for every matrix $M \in \mathbb{R}^{n \times n}$. Let $\|\cdot\|_v$ be the vector norm that induces $\|\cdot\|_V$. By the properties of induced matrix norms we get

$$\|f(x) - f(y)\|_v = \|A(x - y)\|_v \leq \|A\|_V \|(x - y)\|_v \leq \|A\|_a \|(x - y)\|_v \leq \alpha \|(x - y)\|_v$$

as desired. Since $f(x) : D \to D$, $f(x)$ is a contraction mapping. This concludes the proof. \(\square\)

We are now ready for the proof of Theorem 3.1, to which we devote the next section.

3.3 Proof of the general qualifying condition

This section contains the main theoretical contribution of the chapter and proves Theorem 3.1. Although the central equations has the same structure as the proof in [1], the underlying principle is new: The proof in [1] is based on the mean value theorem, while the proof in this thesis is based on duality and the KKT conditions. The proof will be given as a series of lemmas as outlined in the following.

Proof of Theorem 3.1. Consider optimization problem (2.6). When the general qualifying condition GQC holds for all $x \in D$, then the following steps ensure that $x^* = f(x^*)$ is the unique optimal solution.

1. First, we restrict ourselves to optimization problems in Fast-Lipschitz form with only inequality constraints, without loss of generality by Lemma 3.5 below.

\(^1\)See Appendix B for a short proof.
2. The inequality-only constrained optimization problem allows us to show that all feasible points of the optimization problem are regular [55], wherefore any optimal point \( \hat{x} \) must fulfill the KKT-conditions, see Lemma 3.7 below.

3. Any point that fulfills the KKT-conditions must be a fixed point of \( f(x) \), see Lemma 3.9.

4. Finally, we show that there exists a unique fixed point \( x^* = f(x^*) \) by Proposition 3.3 and Lemma 3.4. Therefore, \( x^* \) is the only point fulfilling the KKT-conditions and \( x^* \) must be the optimum.

The first lemma will allow us to focus on problems consisting only of inequality constraints. This is an important step that permits us to use the KKT conditions (see, e.g., [46, 3.1.1]) to establish the existence an uniqueness of optimal solutions.

**Lemma 3.5.** If the inequality-only constrained optimization problem

\[
\begin{align*}
\max_x & \quad f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x) \quad \forall i \in \{1, \ldots, n\}
\end{align*}
\]  

is Fast-Lipschitz, then so is any problem

\[
\begin{align*}
\max_x & \quad f_0(x) \\
\text{s.t.} & \quad x_i \leq f_i(x) \quad \forall i \in A \\
& \quad x_i = f_i(x) \quad \forall i \in B, \quad A \cup B = \{1, \ldots, n\}, \quad A \cap B = \emptyset
\end{align*}
\]  

obtained by switching any number of the inequalities for equalities.

**Proof.** Let \( \mathcal{F}_{3.2} \) and \( \mathcal{F}_{3.3} \) be the feasible regions of problem (3.2) and problem (3.3) respectively. The point \( x^* = f(x^*) \) is feasible in both problems, and by definition it is uniquely optimal for problem (3.2) since it is Fast-Lipschitz. Suppose, contrary to the lemma, that problem (3.3) is not Fast-Lipschitz. Then there exists some feasible point \( \hat{x} \in \mathcal{F}_{3.2} \cap \mathcal{F}_{3.3} \) such that \( f_0(\hat{x}) \geq f_0(x^*) \) which contradicts the unique optimality of \( x^* \) in problem (3.2).

With Lemma 3.5 in mind, we develop the rest of the proof of Theorem 3.1 by focusing on the inequality-only constrained problem (3.2) instead of main problem (2.6). The inequality-only problem on canonical form is

\[
\begin{align*}
\min_x & \quad -f_0(x) \\
\text{s.t.} & \quad g_i(x) = x_i - f_i(x) \leq 0 \quad \forall i.
\end{align*}
\]  

**Definition 3.6.** In problem (3.4), a point \( x \) is regular if the gradients of all active constraints at \( x \) form a linearly independent set (see [55]).
Lemma 3.7. If problem (3.2) fulfills the general qualifying condition GQC, then every point \( x \in \mathcal{D} \) in problem (3.4) is regular.

Proof. The gradients of the individual constraints \( g_i(x) \) are the columns of
\[
\nabla g(x) = I - \nabla f(x).
\]
Since condition (GQC.b) implies \( \rho(\nabla f(x)) \leq |||\nabla f(x)||| < 1 \), the eigenvalues of \( \nabla g(x) \) lie in a ball of radius \( \rho(\nabla f(x)) < 1 \), centered at 1. Hence, no eigenvalue of \( \nabla g(x) \) is zero and \( \nabla g(x) \) is invertible, wherefore the constraint gradients \( \nabla g_i(x) \) (the columns of \( \nabla g(x) \)) form a linearly independent set, which concludes the proof. 

We now scalarize problem (3.4), by considering
\[
\begin{align*}
\min_{x} & \quad -\mu^T f_0(x) \\
\text{s.t.} & \quad g_i(x) = x_i - f_i(x) \leq 0 \quad \forall i.
\end{align*}
\]
for a positive vector \( \mu \in \mathbb{R}^m \) (see Appendix A). Since an arbitrary scaling of \( \mu \) does not change the solution of the problem, we restrict ourselves to \( \mu \) fulfilling \( \sum_k \mu_k = 1 \).

Introduce dual variables \( \lambda \in \mathbb{R}^n \), and form the Lagrangian function
\[
L(x, \lambda) = -\mu^T f_0(x) + \lambda^T (x - f(x)).
\]
Given that every \( x \in \mathcal{D} \) in problem (3.5) is regular (Lemma 3.7), any pair \( (\hat{x}, \hat{\lambda}) \) of locally optimal variables must satisfy the KKT-conditions (see e.g., [46, 3.1.1]). In particular, \( \hat{x} \) must be a minimizer of \( L(x, \hat{\lambda}) \), which requires
\[
\nabla_x L(\hat{x}, \hat{\lambda}) = -\nabla f_0(\hat{x}) \mu + \hat{\lambda} - \nabla f(\hat{x}) \hat{\lambda} = 0,
\]
and complementarity must hold, i.e.,
\[
\hat{\lambda}_i(\hat{x}_i - f_i(\hat{x})) = 0 \quad \forall i.
\]
We will soon show that GQC implies \( \hat{\lambda} > 0 \). To this end, the following remark is useful.

Remark 3.8. Let \( \lambda = Ac \). If \( c > 0 \) and \( A \geq 0 \) with non-zero rows, then \( \lambda > 0 \).

The statement above is trivial, but we give it as Remark 3.8 since we will refer to it several times throughout the thesis. Note that \( A \geq 0 \) and \( c > 0 \) is not sufficient for \( Ac > 0 \), this is the reason that condition (GQC.a) requires non-zero rows.

The following lemma is the main part of the proof of Theorem 3.1 and establishes that the optimal dual variable is strictly positive.

Lemma 3.9. Whenever GQC holds, any pair \( (\hat{x}, \hat{\lambda}) \) with \( \hat{x} \in \mathcal{D} \) satisfying equation (3.6) must have \( \hat{\lambda} > 0 \).
3.3. Proof of the general qualifying condition

Proof. For notational convenience, let us fix one \( x \in D \), which we denote \( \hat{x} \), and introduce

\[
A \triangleq \nabla f(\hat{x}) \in \mathbb{R}^{n \times n} \quad \text{and} \quad c \triangleq \nabla f_0(\hat{x}) \mu \in \mathbb{R}^n. \tag{3.8}
\]

Note that condition \((\text{GQC.a})\) and Remark 3.8 give \( c > 0 \) for every \( \mu > 0 \).

With the new notation, equation (3.6) can be written as

\[
-c + \hat{\lambda} - A \hat{\lambda} = 0, \quad \text{or} \quad \hat{\lambda} = (I - A)^{-1} c \tag{3.9}
\]

whenever \( I - A = I - \nabla f(\hat{x}) \) is invertible, which is true for all \( x \in D \) (proof of Lemma 3.7). The expansion of this inverse gives

\[
\hat{\lambda} = (I + A + A^2 + \ldots) c \tag{3.10a}
\]

\[
= (I + A^k + A^{2k} + \ldots)(I + A + \cdots + A^{k-1}) c
= \sum_{l=0}^{\infty} (A^k)^l (I + A + \cdots + A^{k-1}) c. \tag{3.10b}
\]

The first step is showing that the “left matrix” above is non-negative with non-zero rows. If \( k \to \infty \), the “right vector” becomes identical to the right hand side of (3.10a). The “left matrix” must therefore equal identity \( I \), which is non-negative with non-zero rows. For all other \( k < \infty \), the “left matrix” is ensured non-negative by condition \((\text{GQC.c})\) (with non-zero rows guaranteed by the first term \( A^0 = I \)).

When the “left matrix” is non-negative with non-zero rows, a sufficient condition for \( \hat{\lambda} > 0 \) is that the right vector is positive (Remark 3.8), i.e.,

\[
(I + A + \cdots + A^{k-1}) c > 0 \quad \iff \quad -B c < c. \tag{3.11}
\]

When \( k = 1 \), then \( B = 0 \) and (3.11) holds trivially, since \( c > 0 \). For \( k > 1 \), let \( c_{\min} \) and \( c_{\max} \) be the minimum and maximum elements of \( c \) and consider row \( i \) of inequality (3.11). We can now bound the right side by \( c_{\min} \leq c_i \) and the left side by

\[
\left| (-Bc)_i \right| = \left| \sum_{j=1}^{n} B_{ij} c_j \right| \leq \sum_{j=1}^{n} |B_{ij}| |c_j| \leq \max_i \sum_{j=1}^{n} |B_{ij}| c_{\max} = \|B\|_\infty c_{\max}. \tag{3.12}
\]

Therefore, equation (3.11) holds if

\[
\|B\|_\infty < \frac{c_{\min}}{c_{\max}}. \tag{3.13}
\]

\(2\) A different way to see this is by noting that \( \lim_{k \to \infty} A^k = 0 \) since \( \rho(|A|)1 \) by condition \((\text{GQC.b})\). The “left matrix” therefore evaluates to \( \sum_{l=1}^{\infty} 0^l \), where only the term \( 0^0 = I \) gives a contribution.
3. A New Set of Qualifying Conditions

Let \( \nabla_i f_0(\hat{x}) \) be the \( i \)th row of \( \nabla f_0(\hat{x}) \) and define

\[
a(\mu) = \arg\min_{\nabla_i f_0(\hat{x})} \mu_i, \quad \text{and} \quad b(\mu) = \arg\max_{\nabla_i f_0(\hat{x})} \mu_i.
\]

Introduce

\[
d_k(\mu) = \frac{a_k(\mu)}{b_k(\mu)}
\]

and express the components of \( a \) as \( a_k(\mu) = d_k(\mu)b_k(\mu) \). Since \( c = \nabla f_0(\hat{x})\mu \), we have

\[
c_{\min} = a(\mu) \nabla f_0(\hat{x}) = \sum_k a_k(\mu)\mu_k = \sum_k d_k(\mu)b_k(\mu)\mu_k,
\]

and

\[
c_{\max} = b(\mu) \nabla f_0(\hat{x}) = \sum_k b_k(\mu)\mu_k.
\]

The fraction in (3.13) can therefore be bounded by

\[
\frac{c_{\min}}{c_{\max}} = \frac{\sum_k a_k(\mu)\mu_k}{\sum_k b_k(\mu)\mu_k} \geq \frac{\sum_k d_{\min}(\mu)\mu_k}{\sum_k b_k(\mu)\mu_k} = d_{\min}(\mu),
\]

where

\[
d_{\min}(\mu) = \min_k d_k(\mu) = \min_k \frac{a_k(\mu)}{b_k(\mu)} \geq \min_k \frac{\min_i [\nabla f_0(\hat{x})]_{ik}}{\max_i [\nabla f_0(\hat{x})]_{ik}} = q(\hat{x})
\]

with \( q(x) \) defined in (3.1). By the definition of \( B \) and condition (GQC.d) we get

\[
\|B\|_\infty = \left\| \sum_{l=1}^{k-1} \nabla f(\hat{x})^l \right\|_\infty < q(\hat{x}),
\]

which together with inequalities (3.13)-(3.15) ensures

\[
\|B\|_\infty = \left\| \sum_{l=1}^{k-1} \nabla f(\hat{x})^l \right\|_\infty < q(\hat{x}) \leq d_{\min} \leq \frac{c_{\min}}{c_{\max}},
\]

wherefore \( \hat{\lambda} > 0 \) by inequalities (3.11)-(3.13) and Remark 3.8.

Now, we now that every pair \((\hat{x}, \hat{\lambda})\) satisfying the KKT conditions must have \( \hat{\lambda} > 0 \), provided that GQC holds. Furthermore, strict complementarity (equation (3.7)) must hold. Since \( \hat{\lambda}_i > 0 \), we have \( \hat{\lambda}_i(\hat{x}_i - f_i(\hat{x})) = 0 \) only if \( \hat{x}_i - f_i(\hat{x}) = 0 \), i.e., \( \hat{x} = f(\hat{x}) \). In other words, any candidate for primal optimality must be a fixed point of \( f(x) \). It remains to show that there always exists a fixed point, and that
3.4 Special cases

this fixed point is unique. But this is already done, since GQC and Lemma 3.4 ensure \( f(x) \) is contractive, and the result follow from Proposition 3.3.

By now, we have shown that the unique point \( x^* = f(x^*) \) is the only possible optimum of problem (3.5). Since this is true for any scalarization vector \( \mu > 0 \), \( x^* \) is the unique Pareto optimal point of problem (3.2). Finally, Lemma 3.5 extends this result to the originally considered problem (2.6). By this, we have taken all the steps to prove Theorem 3.1.

In the next section we revisit the GQC, through a number of special cases.

### 3.4 Special cases

In this section, we discuss the general qualifying condition (GQC) in more detail. Moreover, we present several new qualifying conditions, each of which implies GQC.

GQC has the benefit of giving a unified view of the qualifying conditions. This is convenient for proving properties of Fast-Lipschitz problems, and also for giving an overall understanding for what the qualifying conditions ensure. However, GQC may not always be suitable when determining whether or not a given problem (or class of problems) is Fast-Lipschitz. This is because the generality of GQC comes at the price of analytical and computational complexity and cumbersome notation. For example, conditions (GQC.c) and (GQC.d) become increasingly tedious to verify as the integer \( k \) grows. However, as we will see, the special cases can yield clean and easily verifiable conditions, which are much easier to use in practice than GQC.

Furthermore, the specialized cases provide easy comparison to the old qualifying conditions, and other related work such as the standard and type-II standard function of [9] and [49]. We start with the simplest case \( Q_1 \) given below.

<table>
<thead>
<tr>
<th>Qualifying Condition 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q.1.a) ( \nabla f_0(x) \geq 0 ) with non-zero rows</td>
</tr>
<tr>
<td>(Q.1.b) (</td>
</tr>
<tr>
<td>(Q.1.c) ( \nabla f(x) \geq 0 )</td>
</tr>
</tbody>
</table>

Qualifying condition \( Q_1 \) is the special case of GQC when \( k = 1 \). It is the simplest case of Fast-Lipschitz optimization, and only requires a monotonic objective function \( f_0(x) \) and a monotonic, contractive constraint function \( f(x) \). \( Q_1 \) is highly related to standard interference functions [9]. In fact, any problem (2.6) with a monotonic objective function and standard constraints is Fast-Lipschitz [53, Theorem 4.2], [56]. The difference between \( Q_1 \) and case (i) of the Old Qualifying Conditions (2.8) lies in condition (Q.1.a) where we now allow \( \nabla f_0(x) \geq 0 \) as long as no row consists only of zeros.

Qualifying Condition 2 is a simplified version of GQC with \( k = 2 \).
3. A New Set of Qualifying Conditions

<table>
<thead>
<tr>
<th>Qualifying Condition 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q\textsubscript{2.a}) $\nabla f_0(x) &gt; 0$</td>
</tr>
<tr>
<td>(Q\textsubscript{2.b}) $\nabla f(x)^2 \geq 0$, (e.g., $\nabla f(x) \leq 0$)</td>
</tr>
<tr>
<td>(Q\textsubscript{2.c}) $|\nabla f(x)|_\infty &lt; q(x) \triangleq \min_j \frac{\min_i</td>
</tr>
</tbody>
</table>

Proposition 3.10. Qualifying condition Q\textsubscript{2} implies GQC

Proof. Condition (GQC.a) is implied by (Q\textsubscript{2.a}). Note that if any element of $\nabla f_0(x)$ is zero, then $q(x) = 0$ and condition (Q\textsubscript{2.c}) cannot be fulfilled. We can therefore, without loss of generality, use a strict inequality in (Q\textsubscript{2.a}). Condition (GQC.b) is fulfilled since $\|\nabla f(x)\|_\infty < q(x)$ by condition (Q\textsubscript{2.c}) and $q(x) \leq 1$. Finally, conditions (GQC.c) and (GQC.d) are (for $k = 2$) given exactly by (Q\textsubscript{2.b}) and (Q\textsubscript{2.c}) respectively. \hfill \Box

Condition (Q\textsubscript{2.b}) requires that the square of the gradient is non-negative, but is particularly easy to verify when $\nabla f(x) \leq 0$. Note that also $\nabla f(x) \geq 0$ fulfills (Q\textsubscript{2.b}) i.e., (Q\textsubscript{2.b}) is more general than (Q\textsubscript{1.c}). However, this generalization comes at a cost since (Q\textsubscript{2.c}) is more restrictive than (Q\textsubscript{1.b}) in the sense that it requires the specific norm $\|\cdot\|_\infty$ and that $q(x)$ in general is less than one.

The formulation where $\nabla f(x) \leq 0$ corresponds to a non-increasing objective function, and the norm $\|\nabla f(x)\|_\infty$ is small enough, Q\textsubscript{2} is closely related to type-II standard functions [56].

<table>
<thead>
<tr>
<th>Qualifying Condition 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q\textsubscript{3.a}) $\nabla f_0(x) &gt; 0$</td>
</tr>
<tr>
<td>(Q\textsubscript{3.b}) $|\nabla f(x)|_\infty &lt; \frac{q(x)}{1 + q(x)}$</td>
</tr>
</tbody>
</table>

Qualifying condition Q\textsubscript{3} can be seen as the special case of GQC when $k = \infty$, as we see in Proposition 3.11. In contrast to the other qualifying conditions, Q\textsubscript{3} does not require non-negativity of $\nabla f(x)$ (or $(\nabla f(x))^k$). It is only required that $f_0$ is monotonic and the infinity norm of $\nabla f(x)$ is small enough. However, (Q\textsubscript{3.b}) is significantly stricter than (Q\textsubscript{2,c}) because $0 \leq q(x) \leq 1$ implies $\|\nabla f(x)\|_\infty < q(x)/(1 + q(x)) \leq 1/2$ in (Q\textsubscript{3.b}).

Proposition 3.11. Qualifying condition Q\textsubscript{3} implies GQC.

Proof. Condition (Q\textsubscript{3.a}) implies condition (GQC.a) and condition (Q\textsubscript{3.b}) implies $\|\nabla f(x)\|_\infty < 1/2$, whereby condition (GQC.b) is fulfilled. Condition (GQC.c) is
irrelevant when $k = \infty$. Finally, condition (GQC.d) is implied by (Q3.b) because
\[ \|\nabla f(x)\|_\infty < \frac{q(x)}{1 + q(x)} \implies \frac{\|\nabla f(x)\|_\infty}{1 - \|\nabla f(x)\|_\infty} < q(x), \quad (3.16) \]
where the implication above follows from that
\[
\|\sum_{l=1}^{\infty} \nabla f(x)^l\|_\infty \leq \sum_{l=1}^{\infty} \|\nabla f(x)^l\|_\infty \leq \sum_{l=1}^{\infty} \|\nabla f(x)\| \leq \frac{\|\nabla f(x)\|_\infty}{1 - \|\nabla f(x)\|_\infty} < q(x).
\]
The first two inequalities follow from sub-additive and sub-multiplicative properties of matrix norms. The equality follows from the geometric series because $\|\nabla f(x)\|_\infty < 1$, and the last inequality is expression (3.16). This concludes the proof.

The qualifying conditions can be further simplified by introducing
\[ \delta(x) \triangleq \min_{i,j} [\nabla f_0(x)]_{ij} \]
and
\[ \Delta(x) \triangleq \max_{i,j} [\nabla f_0(x)]_{ij}. \]
These are the smallest and largest elements of $\nabla f_0(x)$, regardless of column. The difference compared to $\delta$ and $\Delta$ in (2.8h) is that we now evaluate $\delta$ and $\Delta$ for each $x$, instead of taking the extremes over all $x$. We can now bound
\[ q(x) = \min_j \frac{\min_i [\nabla f_0]_{ij}}{\max_i [\nabla f_0]_{ij}} \geq \frac{\min_{ij} [\nabla f_0]_{ij}}{\max_{ij} [\nabla f_0]_{ij}} = \frac{\delta(x)}{\Delta(x)}. \quad (3.17) \]
Since both $q(x)$ and $q(x)/(1 + q(x))$ are increasing in $q$ (recall $q(x) \geq 0$), we can lower bound $q(x)$ by $\delta(x)/\Delta(x)$ in any one of the qualifying conditions above. This gives the remaining qualifying conditions of this section. They are all special cases of previous conditions — easier to verify and analyze, at the expense of being more conservative.

\[ ^3 \] This is discussed in the proof of Theorem [3.1] after equation (3.10).
3. A New Set of Qualifying Conditions

<table>
<thead>
<tr>
<th>Qualifying Condition 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q₄.a) ( \nabla f_0(x) &gt; 0 )</td>
</tr>
<tr>
<td>(Q₄.b) ( \nabla f(x)^2 \geq 0 ), ( \text{e.g., } \nabla f(x) \leq 0 )</td>
</tr>
<tr>
<td>(Q₄.c) ( |\nabla f(x)|_\infty &lt; \frac{\delta(x)}{\Delta(x)} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Qualifying Condition 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q₅.a) ( \nabla f_0(x) &gt; 0 )</td>
</tr>
<tr>
<td>(Q₅.b) ( |\nabla f(x)|_\infty &lt; \frac{\delta(x)}{\delta(x) + \Delta(x)} )</td>
</tr>
</tbody>
</table>

Qualifying conditions \( Q₄ \) and \( Q₅ \) are obtained when inserting inequality (3.17) in (Q₂.c) and (Q₃.b) respectively. They imply \( Q₂ \) and \( Q₃ \) which in turn imply GQC by construction. Note that qualifying conditions \( Q₄ \) and \( Q₅ \) have previously appeared as [57, case (ii)-(iii)].

We now give the final special case of this section. Qualifying condition \( Q₆ \) was constructed in the process of examining type-II problems in Chapter 6. It is an adaptation of qualifying condition \( Q₂ \) to the case when the infinity norm of the gradient cannot be precisely bound. Instead, \( Q₆ \) taylors the requirement for the cost function gradient to lie along a line defined by the constraints.

<table>
<thead>
<tr>
<th>Qualifying Condition 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q₆.a) ( \nabla f(x)^2 \geq 0 ), ( \text{e.g., } \nabla f(x) \leq 0 )</td>
</tr>
<tr>
<td>(Q₆.b) (</td>
</tr>
<tr>
<td>(Q₆.c) ( f_0(x) = h(\sum_i s_i x_i) ) for a strictly increasing function ( h(z) ) where ( s = (I - B)^{-1}c ) and ( c &gt; 0 )</td>
</tr>
</tbody>
</table>

Note that \( |\nabla f(x)| \) in condition (Q₆.b) represents a matrix with the (element-wise) absolute values of \( \nabla f(x) \).

**Proposition 3.12.** Qualifying Condition \( Q₆ \) implies GQC.

**Proof.** The proof will show how \( Q₆ \) implies \( Q₂ \) for an equivalent problem. This in turn implies GQC (Proposition 3.10). From (Q₆.b) we have \( \rho(B) < 1 \), so it follows that

\[
s = (I - B)^{-1}c = (I + B + B^2 + \ldots) c > 0 \quad \geq 0 \text{ with non-zero rows}
\]

32
by Remark 3.8. Let $S = \text{diag}(s)$, whereby $S^{-1}$ exists and has positive diagonal elements $1/s_i$. Introduce the variable transform $y = Sx \implies x = S^{-1}y$ and insert this into optimization problem (2.6):

$$\max_y f_0(S^{-1}y)$$

s.t. $S^{-1}y \leq f(S^{-1}y)$.

Since $S$ is a positive diagonal matrix, the problem above is equivalent to

$$\max_y g_0(y) \triangleq f_0(S^{-1}y)$$

s.t. $y \leq g(y) \triangleq Sf(S^{-1}y)$.

(3.18)

Because $f_0(x) = h(s^Tx)$ by the assumption (Q.6.c), the objective function is

$$g_0(y) = f_0(S^{-1}y) = h(s^TS^{-1}y) = h(1^Ty)$$

with gradient

$$[\nabla g_0(y)]_{ij} = \nabla h_j(1^Ty) \frac{\partial 1^Ty}{\partial y_i} = \nabla h_j(1^Ty).$$

(3.19)

Since $h(z)$ is increasing in $z$, the gradient $\nabla h(z)$ is positive for any $z$, so $\nabla g_0(y) > 0$ and problem (3.18) fulfills (Q.2.a).

The constraint function gradient is given by

$$\nabla g(y) = S^{-1}\nabla f(S^{-1}y)S,$$

so the square of the constraint function is

$$\nabla g(y)^2 = \left( S^{-1}\nabla f(S^{-1}y)S \right)^2 = S^{-1}\nabla f(S^{-1}y)^2S.$$

By condition (Q.6.a) we have $\nabla f(S^{-1}y)^2 \geq 0$, whereby $\nabla g(y)^2 \geq 0$ and problem (3.18) fulfills (Q.2.b).

Condition (Q.2.c) requires

$$\|\nabla g(y)\|_\infty < \min_j \frac{\min_i [\nabla g_0(y)]_{ij}}{\max_i [\nabla g_0(y)]_{ij}},$$

(3.20)

which is true by construction. To see this, note that inserting (3.19) in the right hand side above results in

$$\min_j \frac{\min_i [\nabla g_0(y)]_{ij}}{\max_i [\nabla g_0(y)]_{ij}} = \min_j \frac{\min_i \nabla h_j(1^Ty)}{\max_i \nabla h_j(1^Ty)} = 1.$$

Equation (3.20) therefore requires

$$\|\nabla g(y)\|_\infty = \|S^{-1}\nabla f(S^{-1}y)S\|_\infty = \max_i \sum_j \left| \nabla f_j(S^{-1}y) \frac{s_j}{s_i} \right| < 1$$

33
3. A New Set of Qualifying Conditions

for all \( y \). Since \( s_i, s_j > 0 \) and \( |\nabla_i f_j (x)| \leq B_{ij} \) for all \( x \), this holds if

\[
\max_i \sum_j B_{ij} s_j = s_i < 1,
\]
or equivalently, if \( \sum_j B_{ij} s_j < s_i \) for all \( i \). This is the \( i \)th row of \( (I - B)s > 0 \), which holds by construction since

\[
(I - B)s = (I - B)(I - B)^{-1} c = c > 0
\]

by the definition of \( s \). This concludes the proof.

We will now end this section with the observation that the old qualifying conditions are a special case of GQC.

**Proposition 3.13.** The Old Qualifying Conditions (2.8) imply GQC.

**Proof.** We will show how each case of the old qualifying conditions imply one of the qualifying conditions 1-5 above, which in turn implies GQC.

**case (i) \( \Rightarrow \) Q \( 1 \):** Condition (Q \( 1 \).a) is implied by (2.8a), (Q \( 1 \).b) and (Q \( 1 \).c) are the same as (2.8b) and (2.8c).

**case (ii) \( \Rightarrow \) Q \( 4 \):** Conditions (Q \( 4 \).a) and (Q \( 4 \).b) are the same as (2.8a) and (2.8e). Condition (2.8d) implies \( \nabla f_0 (x) = c1 \), whereby \( \delta (x) = \Delta (x) = c \). Condition (Q \( 4 \).c) therefore requires that \( \|\nabla f(x)\|_\infty < 1 \), which is ensured by (2.8f).

**case (iii) \( \Rightarrow \) Q \( 5 \):** Condition (2.8g) requires \( f_0 (x) \) to be scalar valued, whereby \( \nabla f_0 (x) \) only has one column. The deltas of condition (2.8h) can therefore be written as \( \delta = \min_{x \in \mathcal{D}} \delta (x) \leq \delta (x) \) and \( \Delta = \max_{x \in \mathcal{D}} \Delta (x) \geq \Delta (x) \). This gives

\[
\frac{\delta}{\Delta} \leq \frac{\delta (x)}{\Delta (x)} ,
\]

whereby

\[
\|\nabla f(x)\|_\infty < \frac{\delta}{\delta + \Delta} = \frac{\delta}{1 + \delta/\Delta} \leq \frac{\delta (x)/\Delta (x)}{1 + \delta (x)/\Delta (x)} = \frac{\delta (x)}{\delta (x) + \Delta (x)}
\]

and (Q \( 5 \).b) is ensured. The first inequality is condition (2.8h), the second inequality follows from (3.21) because \( h(a) = a/(1 + a) \) is an increasing function of \( a \).

We have now showed how each case of the Old Qualifying Conditions implies GQC, through the implication chain

\[
\text{GQC} \iff \text{Q4} \iff \text{Q2} \iff \text{Q5} \iff \text{Q1} \iff \text{Q3} \iff \text{Q1}. \]

This concludes the proof. \( \square \)
This chapter presented a new general qualifying condition for Fast-Lipschitz optimization, which unifies and extends the previously existing set of qualifying conditions. The general qualifying condition was then analyzed through several special cases, which are easier to verify from an analytical and computational point of view, and easier to compare to the previously known qualifying conditions. The new qualifying conditions only provide sufficient requirements, so a problem that fails to fulfill the qualifying conditions may still be Fast-Lipschitz. For example, less stringent conditions can be found through changes of variable or reformulation of the constraints. The qualifying conditions of this chapter assumed the problem to be in Fast-Lipschitz form. In the next chapter, we examine relaxations of the Fast-Lipschitz form.
Chapter 4

Relaxations of the Fast-Lipschitz form

This chapter considers relaxations of the Fast-Lipschitz form given in equation (2.6) that, e.g., require the same number of constraints as variables. Section 4.1 shows how the qualifying conditions (which are constructed for maximization problems) remain the same also for minimization problems. Section 4.2 shows a technique for handling more constraints than variables, and Section 4.3 shows a situation with fewer constraints than variables. Finally, Section 4.4 discusses the case when the objective function does not depend on all variables.

4.1 Fast-Lipschitz minimization

The framework of Fast-Lipschitz optimization is formulated for maximization problems. Through a change of variables, any minimization problem can be formulated as an equivalent maximization problem. This is useful when dealing with power control problems, which are normally written by minimization. The following lemma shows how minimization problems fit into the Fast-Lipschitz framework.

Lemma 4.1 (Fast-Lipschitz minimization). Consider

\[
\begin{align*}
\min \ & g_0(x) \\
\text{s.t.} \ & x_i \geq g_i(x) \ \forall i \in A \\
& x_i = g_i(x) \ \forall i \in B,
\end{align*}
\]

(4.1)

where \( g_0(x), g(x) = [g_i(x)] \), \( A \) and \( B \) fulfill the assumptions of Definition 2.1 Then, problem (4.1) is Fast-Lipschitz if \( g_0(x) \) and \( g(x) \) fulfill the qualifying conditions.
Proof. Let \( x = -y \) and form the equivalent problem

\[
\begin{align*}
\text{max} \quad & -g_0(-y) = f_0(y) \\
\text{s.t.} \quad & y_i \leq -g_i(-y) = f_i(y) \quad \forall i \in \mathcal{A} \\
& y_i = -g_i(-y) = f_i(y) \quad \forall i \in \mathcal{B}.
\end{align*}
\]

(4.2)

In order to check the qualifying conditions one needs \( \nabla f_0(y) \) and \( \nabla f(y) \). But

\[
(\nabla_y f_0(y))^T = \frac{\partial f_0(y)}{\partial y} = -\frac{\partial (-g_0(x))}{\partial x} \frac{\partial x}{\partial y} = -\frac{\partial g_0(x)}{\partial x}(-1) = (\nabla_x g_0(x))^T,
\]

and analogously, \( \nabla_y f(y) = \nabla_x g(x) \). Since \( g_0(x) \) and \( g(x) \) fulfill the qualifying conditions, the equivalent problem (4.2) is Fast-Lipschitz.

4.2 Additional constraints

In this section we complement the Fast-Lipschitz form (2.6) with an additional set \( \mathcal{X} \). Hence, we consider the following problem:

\[
\begin{align*}
\text{max} \quad & f_0(x) \\
\text{s.t.} \quad & x_i \leq f_i(x) \quad \forall i \in \mathcal{A} \\
& x_i = f_i(x) \quad \forall i \in \mathcal{B} \\
& x \in \mathcal{X}.
\end{align*}
\]

(4.3)

Corollary 4.2. If GQC holds, and \( \mathcal{X} \) contains a point \( x^* = f(x^*) \), then problem (4.3) is Fast-Lipschitz.

Proof. Relax the problem by removing the new constraint \( x \in \mathcal{X} \). The relaxed problem is our main problem (2.6), whereby the qualifying conditions and Theorem 3.1 ensure \( x^* = f(x^*) \) is the unique optimum. Since \( x^* \in \mathcal{X} \), this is also the unique optimum of problem (4.3).

In theory, we can handle any set \( \mathcal{X} \) provided we can show \( x^* \in \mathcal{X} \). For example, \( \mathcal{X} \) does not need to be bounded, convex, or even connected (i.e., \( \mathcal{X} \) can consist of mutually disconnected subsets). In practice however, the most common form of \( \mathcal{X} \) is a box constraint, for example a requirement of non-negativity, \( \mathcal{X} = \{ x : 0 \leq x \} \). In these cases, \( \mathcal{X} \) becomes the natural choice for the imagined bounding box \( \mathcal{D} \).

4.3 Fewer constraints than variables – Constant constraints

The Fast-Lipschitz form in problem (2.6) requires one (and only one) constraint \( f_i \) for each variable \( x_i \). In this section we will look at the case when the number of constraints \( (f_i) \) are fewer than the number of variables. We will assume that the individual variables are upper and lower bounded, which is always the case for problems of engineering interest, such as wireless networks. This means we get an
extra constraint set $\mathcal{X}$ as discussed in Section 4.2. We investigate the case when all constraints are inequalities. It follows from Lemma 3.5 that the following results are true also for problems with equality constraints.

Consider a partitioned variable $x = [y^T \ z^T]^T \in \mathcal{X} \subset \mathbb{R}^n$ and the problem

$$
\begin{align*}
\max & \quad f_0(x) \\
\text{s.t.} & \quad y \leq f_y(x) \quad \text{with} \quad \mathcal{X} = \left\{ x : \left\{ \begin{array}{l} y \in \mathcal{X}_y = \{ y : a_y \leq y \leq b_y \} \\
 z \in \mathcal{X}_z = \{ z : a_z \leq z \leq b_z \} \end{array} \right. \right\}. \quad (4.4)
\end{align*}
$$

In the formulation above, there are no constraints $f_i$ for the variables $z_i$. However, by enforcing $z \in \mathcal{X}_z$ twice we get the equivalent problem

$$
\begin{align*}
\max & \quad f_0(x) \\
\text{s.t.} & \quad y \leq f_y(x) \quad \text{with} \quad \mathcal{X} = \left\{ x : \left\{ \begin{array}{l} y \in \mathcal{X}_y = \{ y : a_y \leq y \leq b_y \} \\
 z \leq f_z(x) = b_z \\
 x \in \mathcal{X} \right. \right\}. \quad (4.5)
\end{align*}
$$

This problem has the right form (4.3) and is Fast-Lipschitz if

$$
\nabla f = \begin{bmatrix}
\nabla y f_x(x) & \nabla y f_z(x) \\
\nabla z f_y(x) & \nabla z f_z(x)
\end{bmatrix}
$$

and

$$
\nabla f_0 = \begin{bmatrix}
\nabla y f_0(x) \\
\nabla z f_0(x)
\end{bmatrix}
$$

fulfills GQC. Since $f_z(x) = b_z$ is constant, $\nabla f(x)$ simplifies to

$$
\nabla f = \begin{bmatrix}
\nabla y f_y(x) & 0 \\
\nabla z f_y(x) & 0
\end{bmatrix}.
$$

the qualifying conditions. Moreover, they add nothing to either of $\|\cdot\|_1$ or $\|\cdot\|_\infty$ and does not have an impact on the qualifying conditions. The special structure of $\nabla f_0(x)$ can be exploited to construct less restrictive qualifying conditions.

First, consider problem (4.4) with a fixed $z \in \mathcal{X}_z$. The problem can be written

$$
\begin{align*}
\max & \quad f_0|_z(y) \\
\text{s.t.} & \quad y \leq f_y|_z(y) \quad \text{with} \quad \mathcal{X} = \left\{ y \in \mathcal{X}_y \right. \right\}. \quad (4.6)
\end{align*}
$$

The notation $f_0|_z(y)$ and $f_y|_z(y)$ simply means $f_0(x)$ and $f_y(x)$ as in problem (4.4), but indicates that $z$ (which is fixed) should be thought of as a parameter rather than a variable. We will refer to problem (4.6) as the subproblem $(f_0|_z, f_y|_z)$.

**Proposition 4.3.** Consider problem (4.5). Suppose that

(a) the subproblem $(f_0|_z, f_y|_z)$ fulfills GQC for all $z \in \mathcal{X}_z$, and it holds for all $x \in \mathcal{X}$, that either

(b.i) $\nabla z f_0(x) \geq 0$, and $\nabla z f_y(x) \geq 0$ with non-zero rows, or
4. Relaxations of the Fast-Lipschitz form

\( (b.ii) \quad \nabla_z f_0(x) \succeq 0 \) with non-zero rows, and \( \nabla_z f_y(x) \succeq 0 \), or

\( (b.iii) \quad \frac{\|\nabla_z f_y(x)\|_\infty}{1 - \|\nabla_y f_y(x)\|_\infty} < \frac{\delta_z(x)}{\Delta_y(x)} \),

where \( \delta_z(x) = \min_{ij} [\nabla_z f_0(x)]_{ij} \) and \( \Delta_y(x) = \max_{ij} [\nabla_y f_0(x)]_{ij} \).

Then, problem \( (4.4) \) is Fast-Lipschitz.

Proof. We refer to the arguments of Section 3.3 where we modify Lemma 3.9 as follows.

The particular form and partitioning remains in the definitions (3.8), giving

\[
A = \begin{bmatrix}
\nabla_y f_y(\hat{x}) & 0 \\
\nabla_z f_y(\hat{x}) & 0
\end{bmatrix}
= \begin{bmatrix}
A_{11} & 0 \\
A_{21} & 0
\end{bmatrix}
\quad \text{and} \quad
c = \begin{bmatrix}
\nabla_y f_0(\hat{x}) \\
\nabla_z f_0(\hat{x})
\end{bmatrix}
\mu = \begin{bmatrix}
c_1 \\
c_2
\end{bmatrix}.
\]

(4.7)

Consider again equation (3.9) and denote \( \mathbf{E} \triangleq (I - A)^{-1} \), i.e., \( \lambda = \mathbf{E} c \). As in the proof of Lemma 3.9, \( \mathbf{E} \) is well defined if \( \rho(A) < 1 \). This is the case, since the eigenvalues of a block triangular matrix are the union of the eigenvalues of the diagonal blocks, wherefore \( \rho(A) = \rho(A_{11}) < 1 \) by assumption (a).

As in the proof of Lemma 3.9 we must show \( \lambda > 0 \). This time we will make use of the block structure of \( A \) and \( c \).\(^1\) From the block matrix inverse formula we get

\[
\lambda = (I - A)^{-1} c = \begin{bmatrix}
E_{11} & E_{12} \\
E_{21} & E_{22}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix},
\]

where \( E_{11} = (I - A_{11})^{-1} \), \( E_{12} = 0 \), \( E_{21} = A_{21} E_{11} \) and \( E_{22} = I \), i.e.,

\[
\begin{bmatrix}
\lambda_1 \\
\lambda_2
\end{bmatrix}
= \begin{bmatrix}
E_{11} c_1 \\
E_{21} c_1 + c_2
\end{bmatrix}
= \begin{bmatrix}
E_{11} c_1 \\
A_{21} E_{11} c_1 + c_2
\end{bmatrix}.
\]

Note that \( \lambda_1 = E_{11} c_1 = (I - A_{11})^{-1} c_1 \), so \( \lambda_1 > 0 \) since the subproblem \((f_0|x, f_y|x)\) fulfills Lemma 3.9 by assumption (a). The second block component is

\[
\lambda_2 = A_{21} E_{11} c_1 + c_2 = A_{21} \lambda_1 + c_2.
\]

(4.8)

Given that \( \lambda_1 > 0 \), we need to show \( \lambda_2 > 0 \) if either of assumptions (b.i)-(b.iii) hold.

We start with assumption (b.i), which ensures \( c_2 \succeq 0 \) and \( A_{21} \succeq 0 \) with non-zero rows (so \( A_{21} \lambda_1 > 0 \) by Remark 3.8), wherefore \( \lambda_2 > 0 \) by equation (4.8).

Assumption (b.ii) assures \( A_{21} \succeq 0 \) and \( c_2 > 0 \), wherefore \( \lambda_2 > 0 \) by equation (4.8).

\(^1\)Formulas for the inverse of a block matrix, as well as the products of two block matrices can be found in [58].
Finally, assuming (b.iii) is fulfilled, we see that
\[ \lambda_2 > 0 \text{ if } c_2 > -A_{21}E_{11}c_1. \]
Analogous to equations (3.12) and (3.13), this holds if
\[ \min_i [c_2]_i > \|A_{21}E_{11}\|_\infty \max_i [c_1]_i, \]
or since \( c_1 > 0 \) and \( E_{11} = (I - A_{11})^{-1} = \sum_{k=0}^{\infty} A_{11}^k \), when
\[ \frac{\min_i [c_2]_i}{\max_i [c_1]_i} > \|A_{21} \sum_{k=0}^{\infty} A_{11}^k\|_\infty. \]
(4.9)

By the triangle inequality, the sub-multiplicative property of matrix norms and the geometric series, the right side of equation (4.9) can be upper bounded by
\[ \frac{\|A_{21}\|_\infty}{1 - \|A_{11}\|_\infty} \geq \|A_{21} \sum_{k=0}^{\infty} A_{11}^k\|_\infty \]
(4.10)

Finally the definitions of \( c_i \) and \( \sum_j \mu_j = 1 \) gives
\[ \frac{\min_i [c_2]_i}{\max_i [c_1]_i} = \frac{\min_i \sum_j [\nabla_z f_0(x)]_{ij} \mu_j}{\max_i \sum_j [\nabla_y f_0(x)]_{ij} \mu_j} \geq \frac{\min_ij [\nabla_z f_0(x)]_{ij} \sum_j \mu_j}{\max_ij [\nabla_y f_0(x)]_{ij} \sum_j \mu_j} = \frac{\delta_z(x)}{\Delta_y(x)}. \]
(4.11)

By combining inequalities (4.9)-(4.11), a sufficient condition ensuring \( \lambda_2 > 0 \) is
\[ \frac{\|A_{21}\|_\infty}{1 - \|A_{11}\|_\infty} < \frac{\delta_z(x)}{\Delta_y(x)}, \]
which is guaranteed by assumption (d). This concludes the proof. \( \Box \)

4.4 Non-strictly monotonic objective function – Missing variables in the objective function

Sometimes it is practical or necessary to formulate problems where not all variables appear in the objective function.

For example, the problem
\[
\begin{align*}
\max & \quad f_0(x) \\
\text{s.t.} & \quad x \leq f_z(x, z) \\
& \quad z \leq f_z(x, z)
\end{align*}
\]
has some variables not affecting the objective function and is not in Fast-Lipschitz form. Redefining \( f_0 = f_0(x, z) \) gives a problem of the right form (2.6), but this gives \( \nabla_z f_0(x, z) = 0 \) everywhere. Therefore, condition (GQC.a) and Theorem 3.1 cannot be used to classify the problem as Fast-Lipschitz.
The situation above is a special case of the following problem. Consider a partitioned optimization variable \((x, z)\) and the problem

\[
\text{max } f_0(x, z) \\
\text{s.t. } x \leq f_x(x, z) \\
z \leq f_z(x, z).
\]

(4.12)

Suppose \(f_0\) is monotonic, i.e. \(\nabla f_0 \geq 0\), but partitioned such that

\[
\nabla f_0(x, z) = \begin{bmatrix} \nabla_x f_0(x, z) \\ \nabla_z f_0(x, z) \end{bmatrix},
\]

where \(\nabla_x f_0(x, z)\) has non-zero rows for all \((x, z) \in \mathcal{D}\), while \(\nabla_z f_0(x, z)\) can have zero rows. By partitioning the objective function gradient, one can find situations when problem (4.12) is actually Fast-Lipschitz.

**Proposition 4.4.** Consider problem (4.12). Suppose that, for all \(x \in \mathcal{D}\),

1. \(\nabla_x f_0(x, z) > 0\) and \(\nabla_z f_0(x, z) \geq 0\)
2. \(\|\nabla f(x, z)\| < 1\) for some matrix norm, and
3. \(\nabla_z f_x(x, z)\) has non-zero rows.

Then, problem (4.12) is Fast-Lipschitz.

**Remark 4.5.** The condition that the \(i\)th row of \(\nabla_z f_x(x, z)\) is non-zero means that an increase in the variable \(z_i\) will allow an increase of some variable \(x_j\), which in turn will influence the objective.

**Proof.** The proof of Theorem 3.1 can be reused, with some alterations to Lemma 3.9.

The partitioning of \(\nabla f\) and \(\nabla f_0\) remains in \(A\) and \(c\), i.e.,

\[
A = \begin{bmatrix} \nabla_x f_x(x, z) & \nabla_x f_z(x, z) \\ \nabla_z f_x(x, z) & \nabla_z f_z(x, z) \end{bmatrix}, \quad c = \begin{bmatrix} \nabla_x f_0(x, z) \\ \nabla_z f_0(x, z) \end{bmatrix}, \quad \mu = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}.
\]

Assumption (a) and \(\mu > 0\) gives \(c_1 > 0\) and \(c_2 \geq 0\). Just as in Lemma 3.9, assumptions (b) and (c) guarantee the existence and non-negativity of \(E = (I - A)^{-1}\), so \(\lambda = (I - A)^{-1} c \geq 0\) is well defined and non-negative. Thus, it remains to show \(\lambda = Ec > 0\).

Expressing the inverse \(E = (I - A)^{-1}\) block-wise, we have

\[
(I - A)^{-1} = \begin{bmatrix} I - A_{11} & A_{12} \\ A_{21} & I - A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix},
\]

2In contrast to Lemma 3.9, we only have the weak inequality \(c \geq 0\).
4.4. Non-strictly monotonic objective function

where

\[ E_{11} = \left( I - \left( A_{11} + A_{12} (I - A_{22})^{-1} A_{21} \right) \right)^{-1} \]

and

\[ E_{21} = (I - A_{22})^{-1} A_{21} E_{11}. \]

We now have

\[
\begin{bmatrix}
\lambda_1 \\
\lambda_2
\end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} E_{11} c_1 \\ E_{21} c_1 \end{bmatrix} + \begin{bmatrix} E_{12} \\ E_{22} \end{bmatrix} c_2.
\]

Since \( E \succeq 0 \), all blocks \( E_{ij} \) are non-negative. The second term is always non-negative and can be ignored, it is enough to show that the first term is strictly positive.

Since \( c_1 > 0 \), and \( E_{11} \succeq 0 \), we have that \( \lambda_1 > 0 \) if \( E_{11} \) has non-zero rows (Remark 3.8). This is always the case since \( E_{11} \) (defined as an inverse) is invertible. The second component can be expressed in terms of the first component: \( \lambda_2 = (I - A_{22})^{-1} A_{21} \lambda_1 \). When \( \lambda_1 > 0 \), then \( A_{21} \lambda_1 > 0 \) if \( A_{21} = \nabla_z f(x, z) \) has non-zero rows (Remark 3.8). This is true by assumption (d). Since \( (I - A_{22})^{-1} \) is invertible it has non-zero rows and \( \lambda_2 = (I - A_{22})^{-1} (A_{21} \lambda_1) > 0 \), which concludes the proof.

We end this section by a general example.

**Example**

Start with problem (2.6). For a lighter notation, we assume all constraints are inequalities. Transforming the problem to an equivalent problem on epigraph form gives

\[
\begin{align*}
\max & \quad t \\
\text{s.t.} & \quad t \leq f_0(x), \\
& \quad x \leq f(x).
\end{align*}
\]

This problem has a (non-strictly) monotonic objective, regardless of \( f_0(x) \). By writing this as

\[
\begin{align*}
\max & \quad g_0(t, x) = t \\
\text{s.t.} & \quad t \leq g_0(t, x) = f_0(x) \\
& \quad x \leq g_0(t, x) = f(x)
\end{align*}
\]

we obtain

\[
\nabla g_0(t, x) = \begin{bmatrix} \nabla_t g_0(t, x) \\ \nabla_x g_0(t, x) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}, \quad \text{and}
\]

\[
\nabla g(t, x) = \begin{bmatrix} \nabla_t g(t, x) \\ \nabla_x g(t, x) \end{bmatrix} = \begin{bmatrix} 0 \\ \nabla f_0(x) \end{bmatrix}.
\]

Proposition 4.4 can now be applied, and the problem is Fast-Lipschitz if \( \nabla g \succeq 0 \), \( \rho(\nabla g) < 1 \) and \( \nabla f_0 \) has non-zero rows. Since the eigenvalues of a block triangular
matrix are the union of the eigenvalues of the diagonal blocks, we have \( \rho(\nabla g) = \rho(\nabla f) \), wherefore the problem is Fast-Lipschitz if

- \( \nabla f(x) \geq 0, \| \nabla f(x) \| < 1 \), and

- \( \nabla f_0(x) \geq 0 \) with non-zero rows.

This is precisely qualifying condition \( Q_1 \) applied to the original problem (2.6), i.e., no generalization was achieved by considering the epigraph form of the problem.

### 4.5 Summary

In this chapter we examined several variations of the Fast-Lipschitz form. In particular, we introduced a Fast-Lipschitz form also for minimization problems, and considered problems not having the same number of constraints as variables, or having variables that do not affect the objective function. For these cases we established conditions for which they are still Fast-Lipschitz. Based on these new results, a larger set of convex and non-convex optimization problems can be solved by the framework of this thesis, both in a centralized and in a distributed set-up.
Chapter 5

Convergence Analysis

It is often the case that a distributed optimization problem needs to be continually solved, e.g., when a set of nodes want to track or interact with a changing environment. Therefore, the solution algorithms should converge to the optimal solution as fast as possible. Solving a Fast-Lipschitz optimization problem amounts to solving the system $\mathbf{x}^* = f(\mathbf{x}^*)$, which is computationally much easier than solving the original optimization problem. This suggests that the fixed point iterations of Fast-Lipschitz optimization converge faster than traditional optimization problems. The convergence advantage of Fast-Lipschitz optimization has been seen in practice [1]. However, no theoretical comparison of the convergence speed of Fast-Lipschitz optimization has been shown in the previous chapters. This chapter constitutes a first step towards such an analysis.

The characterization of the convergence speed of optimization algorithms is notoriously hard [55]. Ultimately, it reduces to the convergence analysis of, in general, non linear dynamical discrete systems, and it depends on which norm is used. We restrict ourselves to considering the spectral radius of the matrix associated to the dynamical system. In fact, the spectral radius is sometimes known as the asymptotic convergence speed. The intuition is that the spectral radius is the smallest induced norm. Thus, a smaller spectral radius is an indicator of a faster convergence. In the following, we give a lower bound on the spectral radius of Lagrangian methods and we compare it to the Fast-Lipschitz convergence speed.

5.1 Problem formulation

In the analysis, we consider only problems with equality constraints. This technical assumption allows us to use the most general Lagrangian method [55, Sec. 4.4].

Consider a problem on canonical form:

$$\begin{align*}
\min & \quad f_0(\mathbf{x}) \\
\text{s.t.} & \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^n.
\end{align*}$$

(5.1)
This problem can be written in Fast-Lipschitz form as

$$\max \ -f_0(x)$$

s.t.  \[ x = f(x) = x - \beta h(x). \] (5.2)

We assume that strong duality holds for this problem, so that we can apply Lagrangian methods to compute the optimal solution. Note that Lagrangian duality applies to both convex and non-convex problems, provided that constraint qualifications hold \[55\].

In order for problem (5.2) to be Fast-Lipschitz, \( f(x) \) must satisfy the qualifying conditions and the fixed point \( x^* = f(x^*) \) will therefore be unique. This means that there is one feasible point \( (x^*) \). The point must fulfill

$$h(x) = 0,$$ (5.3)

and the optimization problem reduces to solving a system of equations. This is precisely what usually gives Fast-Lipschitz optimization an advantage. However, it is not obvious that optimization problem (5.1) has a unique solution through equation (5.3) if one does not know that the problem is Fast-Lipschitz. Also, the solution has to be calculated in a distributed manner, wherefore problem (5.1) is of substantial interest.

Once the problem is known to be Fast-Lipschitz, the optimal solution can be found by iterating

$$x^{k+1} := f(x^k),$$ (5.4)

as discussed in Section 2.4. We will refer to the fixed point iterations (5.4) as the Fast-Lipschitz method, in order to differentiate it from the Lagrangian method which we introduce in the following section.

### 5.2 Lagrangian method

In this subsection we study the best convergence speed of the Lagrangian method for problem (5.1). We then compare such a bound to the Fast-Lipschitz method in the following subsection.

For the sake of generality, we consider the most general Lagrangian method available for equality constrained problems, which can thus be applied both to convex and non-convex optimization problems. Such a method is a first order one. It is possible to use higher order Lagrangian methods to get faster convergence. However, one can also claim that higher order methods can be used for solving the system \( x^* = f(x^*) \), arising in Fast-Lipschitz optimization. Therefore, it is reasonable to restrict ourselves to first order methods.

The Lagrangian of the problem is given by

$$L(x, \lambda) = f_0(x) + \lambda^T h(x)$$
where $\lambda \in \mathbb{R}^n$, $\lambda \neq 0$, are the Lagrangian multipliers. The optimal primal-dual variables $(x^*, \lambda^*)$ must fulfill
\begin{equation}
\begin{aligned}
\nabla x L(x^*, \lambda^*) &= \nabla f_0(x^*) + \nabla h(x^*) \lambda^* = 0 \\
\nabla \lambda L(x^*, \lambda^*) &= h(x^*) = 0.
\end{aligned}
\end{equation}

This is a system with $2n$ equations in $2n$ variables, whose solutions gives the optimal solution to the problem. The gradient method solves this by taking steps
\begin{equation}
\begin{aligned}
x^{k+1} &= x^k - \alpha \nabla x L(x^k, \lambda^k) \\
\lambda^{k+1} &= \lambda^k + \alpha \nabla \lambda L(x^k, \lambda^k)
\end{aligned}
\end{equation}

Strictly speaking, this is a centralized solution method. It is easy to show that any distributed solution method converges slower or with the same speed as this one \[46\]. For a shorter notation, introduce $y = \begin{bmatrix} x^T & \lambda^T \end{bmatrix}^T$ and
\begin{equation}
\begin{aligned}
M_\alpha(y) &= \begin{bmatrix} x - \alpha \nabla x L(x, \lambda) \\
\lambda + \alpha \nabla \lambda L(x, \lambda) \end{bmatrix} \\
&= \begin{bmatrix} x \\
\lambda \end{bmatrix} - \alpha \begin{bmatrix} \nabla f_0(x) + \nabla h(x) \lambda \\
-h(x) \end{bmatrix}
\end{aligned}
\end{equation}
for some small enough $\alpha > 0$. Equations (5.6) can now be written as
\begin{equation}
y^{k+1} = M_\alpha(y^k)
\end{equation}

Clearly $M_\alpha(y^*) = y^*$, where $y^* = (x^*, \lambda^*)$. If $M_\alpha(y)$ is contractive in some neighborhood around $y^*$, any sequence $y^{k+1} = M_\alpha(y^k)$ within this neighborhood will converge to $y^*$.

The following proposition is a simplified version of \[55\, \text{Prop. 4.4.1}\].

**Proposition 5.1.** Assume that $y^* = M_\alpha(y^*)$ and that the eigenvalues of $\nabla M_\alpha(y^*)$ have magnitudes strictly less than one. Then, for some $y^0$ sufficiently close to $y^*$, the sequence $y^k$ given by $y^{k+1} = M_\alpha(y^k)$ will converge to $y^*$.

The proposition states that the method converges when $\rho(\nabla M_\alpha(y^*)) < 1$, and from the proof in \[55\] it is also clear that convergence becomes faster as the spectral radius becomes smaller. Thus, in order to lower bound the convergence speed of this gradient method, we need the minimal spectral radius of $\nabla M_\alpha(y^*)$. With this goal in mind, we need two technical results before giving the main result of the subsection, which is Proposition 5.5.

Taking the gradient of equation (5.7) gives
\begin{equation}
\nabla M_\alpha(y^*) = I - \alpha B,
\end{equation}
where
\begin{equation}
B = \begin{bmatrix}
\nabla_{xx} L(x^*, \lambda^*) & \nabla h(x^*) \\
-\nabla h(x^*)^T & 0
\end{bmatrix}
\end{equation}
is the gradient of the update direction. Note that $B$ is a $m \times m$ matrix, where $m = 2n$. 

47
Convergence Analysis

Figure 5.1: Three different ±ρ^i_M(α) are shown. The slopes are the eigenvalues μ_i. The first and last (i = m and i = 1) correspond to the lowest and highest eigenvalue μ_i. The middle represents any ρ^i_M(α) where 1 < i < m.

Lemma 5.2. Consider P(α) = I − αB and assume that B has real, positive eigenvalues μ_1 ≥ μ_2 ≥ ⋯ ≥ μ_m > 0. Then, the spectral radius ρ(P(α)) is minimized for

\[ \hat{\alpha} = \frac{2}{\mu_1 + \mu_m}, \]

and the minimal value is

\[ \hat{\rho}_P = \rho(P(\hat{\alpha})) = \frac{\mu_1 - \mu_m}{\mu_1 + \mu_m}. \]

Proof. The ith eigenvalue of P(α) is given by 1 − α_μ_i. Since the spectral radius is the maximal eigenvalue magnitude, we have ρ_P(α) = max_i \{ |1 − α_μ_i| \} = max_i ρ^i_M(α). Given that the eigenvalues of B are real and positive, ρ_P is the piecewise maximum of m linear functions ρ^i_P(α), and there will always exist a small enough α > 0 such that ρ_P < 1. The behavior of ρ_M(α) can be seen in Figure 5.1 from which it should be clear that the minimal value is obtained at the intersection of ρ^1_M and ρ^m_M. Hence, the optimal α, denoted \( \hat{\alpha} \), fulfills 1 − \( \hat{\alpha} \mu_m = \hat{\alpha} \mu_1 - 1 \) and \( \hat{\alpha} = 2/(\mu_1 + \mu_m) \).
wherefore the minimal $\rho_P$ is given by

$$\rho_P(\hat{\alpha}) = \rho_P^m(\hat{\alpha}) = 1 - \frac{2(\mu_1 - \mu_m)}{\mu_1 + \mu_m},$$

which concludes the proof.

**Remark 5.3.** For analytical tractability, the proposition assumes that $B$ has real eigenvalues. The positive eigenvalue assumption is inherited from the Lagrangian method, and it is a necessary condition for convergence. The case when $B$ has complex eigenvalues is left for future work.

Let $\gamma_P = \mu_1/\mu_m$ be the condition number of $B$. Then, the smallest spectral radius of $\nabla M_\alpha$ is

$$\hat{\rho}_M = \rho_M(\hat{\alpha}) = \frac{\gamma_B - 1}{\gamma_B + 1}. \quad (5.9)$$

Note that $d\hat{\rho}_M / d\gamma_B > 0$, i.e., $\hat{\rho}_M$ is an increasing function of $\gamma_B$ in the considered interval $(1, \infty)$. When $\gamma_B = 1$, $\hat{\rho}_M = 0$ and the Lagrangian method converges in one iteration. As $\gamma_B$ grows towards infinity, $\hat{\rho}_M$ will approach 1 and the convergence will stall.

Note that $\hat{\rho}_M$ can be lower bounded by choosing appropriately the matrix $B$, which gives the fastest convergence of the Lagrangian method. Thus, it remains to lower bound $\hat{\rho}_M$, i.e., lower bound the condition number of $B$. We need a second technical results before computing such a lower bound.

**Lemma 5.4.** Let $P$ be a $m \times m$ matrix with real, positive eigenvalues. Then,

$$\gamma_P \geq \frac{\text{trace}(P)}{m \sqrt[\text{m}]{\text{det}(P)}}. \quad (5.10)$$

**Proof.** The eigenvalues $\nu_i$ of any matrix satisfy

$$\sum_i \nu_i = \text{trace}(P) \quad \text{and} \quad \prod_i \nu_i = \text{det}(P). \quad (5.10)$$

We have

$$\text{trace}(P) = \sum_{i=1}^m \nu_i \begin{cases} \leq m \nu_1 & \leftrightarrow \nu_1 \geq \text{trace}(P)/m \\ \geq m \nu_m & \leftrightarrow \nu_m \leq \text{trace}(P)/m. \end{cases}$$

In the same manner we have

$$\text{det}(P) = \prod_{i=1}^m \nu_i \begin{cases} \leq (\nu_1)^m & \leftrightarrow \nu_1 \geq \sqrt[m]{\text{det}(P)} \\ \geq (\nu_m)^m & \leftrightarrow \nu_m \leq \sqrt[m]{\text{det}(P)}. \end{cases}$$

$\text{trace}(P)/m$ and $\sqrt[m]{\text{det}(P)}$ are the arithmetic and geometric means of the eigenvalues, and the bounds above simply state that the means must lie between the
5. Convergence Analysis

smallest and the largest eigenvalue. For a collection of real, positive numbers, the arithmetic mean is smaller than the geometric mean. Therefore, it holds that

$$\nu_m \leq \frac{1}{\sqrt[1/m]{\det(P)}} \leq \frac{\text{trace}(P)}{m} \leq \nu_1,$$

and

$$\gamma_P = \frac{\nu_1}{\nu_m} \geq \frac{\text{trace}(P)}{m \sqrt[1/m]{\det(P)}}.$$

This concludes the proof. \qed

We are now in the position of finding an upper bound for the convergence speed of the Lagrangian method:

**Proposition 5.5.** Let $\nabla M_\alpha(y^*) = I - \alpha B$, with

\[
B = \begin{bmatrix} Q & A \\ -A^T & 0 \end{bmatrix}.
\] (5.11)

Consider the minimum spectral radius $\hat{\rho}_M = \min_{\alpha} \rho(\nabla M_\alpha(y^*))$. Suppose that $B$ has real and positive eigenvalues. Then

$$\hat{\rho}_M \geq \frac{\text{trace}(Q) - m \sqrt{\det(A)^2}}{\text{trace}(Q) + m \sqrt{\det(A)^2}}.$$

**Proof.** The eigenvalues of $B$ fulfill the assumptions of Lemma 5.2 so equation (5.9) gives

$$\hat{\rho}_M = \frac{\gamma_B - 1}{\gamma_B + 1}.$$

For the special structure of $B$, we have $\text{trace}(B) = \text{trace}(Q)$, and $\det(B) = \det(A)^2$ \[58\]. Lemma 5.4 gives

$$\gamma_B \geq \frac{\text{trace}(B)}{m \sqrt[1/m]{\det(B)}} = \frac{\text{trace}(Q)}{m \sqrt[1/m]{\det(A)^2}}.$$

Since $\hat{\rho}_M$ is increasing at $\gamma_B$, this implies

$$\hat{\rho}_M = \frac{\gamma_B - 1}{\gamma_B + 1} \geq \frac{\text{trace}(Q) - m \sqrt{\det(A)^2}}{\text{trace}(Q) + m \sqrt{\det(A)^2}},$$

which concludes the proof. \qed

By the previous proposition, we have an upper bound on the convergence speed of the Lagrangian method. We can now compare it to the speed of the Fast-Lipschitz method in the following section.
5.3 Convergence comparison

If we know that an optimization problem is Fast-Lipschitz, we know that the solution fulfills $x^* = f(x^*)$ and that $f(x) = x - \beta h(x)$ is contractive. Note that this system of equations is half the size of Eq. (5.5). By the argument of the previous subsection, we are interested in the asymptotic convergence speed $\rho_f = \rho(\nabla f(x^*))$. In this sense, the Fast-Lipschitz method is faster than the Lagrangian method if $\rho_f$ is smaller than the smallest possible $\hat{\rho}_M$, that is if

$$\rho_f < \frac{\text{trace}(Q) - m \sqrt{\text{det}(A)^2}}{\text{trace}(Q) + m \sqrt{\text{det}(A)^2}} \leq \hat{\rho}_M.$$  (5.12)

Although equation (5.12) gives a condition that is verifiable, it can be simplified if one allows for a more conservative condition by assuming that $B$ is positive definite with real eigenvalues:

**Proposition 5.6.** Assume $B$ given in (5.11), with $Q = \nabla^2 f_0(x^*) + \sum_k \nabla^2 h_k(x^*) \lambda_k^*$ and $A = \nabla h(x^*)$, has real, positive eigenvalues. Then, the Fast-Lipschitz method (5.4) is faster than the Lagrangian method (5.6) if

$$\frac{\text{trace}(Q)}{m} > \frac{1}{\beta^2} \frac{(1 + \rho_f)^3}{1 - \rho_f}.$$  (5.13)

**Proof.** We need $\rho_f < \hat{\rho}_M$ or, from equation (5.12),

$$\rho_f < \frac{\text{trace}(Q) - m \sqrt{\text{det}(A)^2}}{\text{trace}(Q) + m \sqrt{\text{det}(A)^2}},$$

This can be rewritten as

$$\text{trace}(Q) > \frac{1 + \rho_f}{1 - \rho_f} m \sqrt{\text{det}(A)^2}.$$  (5.13)

This expression is always defined, since the problem is Fast-Lipschitz, which implies $\rho_f < 1$. Recalling that $f(x) = x - \beta h(x)$, we obtain $A = \nabla h(x^*) = (I - \nabla f(x^*)) / \beta$, and we can relate $\text{det}(A)$ to $\rho_f$ as follows: Let $\nu_i(\cdot)$ be the $i$th eigenvalue of a matrix. We have

$$\text{det}(A) = \prod_{i=1}^m \nu_i(A) = \prod_{i=1}^m \nu_i\left(\frac{1}{\beta}(I - \nabla f(x^*))\right) = \frac{1}{\beta^m} \prod_{i=1}^m (1 - \nu_i(\nabla f(x^*))),$$

wherefore

$$m \sqrt{\text{det}(A)^2} = \frac{1}{\beta^2} \left(\prod_{i=1}^m (1 - \nu_i(\nabla f(x^*)))^2\right) \leq \frac{1}{\beta^2} \prod_{i=1}^m (1 + \rho_f)^2 = \frac{(1 + \rho_f)^2}{\beta^2}.$$
This means that inequality (5.13) holds if
\[
\text{trace}(Q) > \frac{1 + \rho_f}{1 - \rho_f} m \frac{(1 + \rho_f)^2}{\beta^2},
\]
which concludes the proof.

The previous proposition gives a criterion to establish when a Fast-Lipschitz method is faster than the Lagrangian method. However, both \(Q = \nabla^2 f_0(x^*) + \sum_k \nabla^2 f_k(x^*) \lambda_k^*\) and \(\rho_f = \rho(\nabla f(x^*))\) depend on the optimal points \(x^*\) and \(\lambda^*\). If this criterion is to be used in practice to assess when distributed optimization should be done by a Fast-Lipschitz method or by a Lagrangian method, it is only interesting before the problem is actually solved (when \(x^*\) and \(\lambda^*\) are unknown). The spectral radius dependence of \(x^*\) can be circumvented by considering \(\max_{x \in D} \rho(\nabla f(x))\). Avoiding the dual optimum \(\lambda^*\) is harder, since the only characterization is \(\lambda^* \neq 0\). Perhaps a bound could be found through the KKT-conditions, or specific properties of the problem could be used. For example, one could make further assumptions on the problem structure, e.g., as follows:

**Corollary 5.7.** Consider problem (5.1). Let \(f_0(x) = -x^T H x - s^T x\), where \(H\) is a positive definite matrix and \(s\) a vector with positive entries. Let \(f(x) = C x + c\), where the matrix \(C\) and vector \(c\) both are constant. Suppose that strong duality holds. Then, the Fast-Lipschitz method (5.4) is faster than the Lagrangian method (5.6) if
\[
\frac{\text{trace}(H)}{m} > \frac{1}{\beta^2} \frac{(1 + \rho_f)^3}{1 - \rho_f}.
\]

In a centralized setting, this linearly constrained quadratic problem is easily solved, wherefore the choice of method is less important. However, distributed applications can still be challenging and require iterative methods, where the convergence rate, and therefore the choice of method, is of high importance. An example that applies the theory of this chapter is given in Section 7.2.

### 5.4 Summary

In this chapter, we provided an initial step for the analysis of the convergence speed of the Fast-Lipschitz optimization method. The method was compared to a general Lagrangian method that can be applied to both convex and non-convex optimization problems. Specifically, we defined the spectral radius of the solution method as the measure of the convergence speed. Under the assumption of real eigenvalues of the Hessian associated to the solution method, a general condition showing when Fast-Lipschitz method is faster than Lagrangian methods was established. Finally, the condition was applied to problems with quadratic cost function and linear constraints.
Chapter 6

Transmit Radio Power Control

In this chapter we apply the theory of Fast-Lipschitz optimization to transmit radio power control. We resume the discussion from Section 2.3 in the background chapter, where we discussed the problem

\[
\min_p \ p \\
\text{s.t.} \quad p_i \geq \beta_i(p) \quad \forall i. \tag{6.1}
\]

Recall that the constraint functions \(\beta_i(p)\) of this problem originally had the interpretation “interference that transmitter \(i\) must overcome to reach the SINR target”. All the power control algorithms discussed in Section 2.3 updated the transmit powers as

\[p_i^{k+1} = \beta_i(p^k),\]

and these iterations converged to a fixed point \(p^* = \beta(p^*)\). This was the case even if \(\beta(p)\) no longer had the physical interpretation of interference to overcome, and \(\beta(p)\) no longer had a clear connection to an optimization problem. In this chapter we establish this connection, so that the fixed points \(p^*\) once again can be given a meaning of optimality.

6.1 Problem formulation

We investigate a general form of the power minimization problem (6.1),

\[
\min_p \ \kappa(p) \\
\text{s.t.} \quad p \geq \beta(p). \tag{6.2}
\]

Throughout the chapter we assume that \(\kappa(p)\) and \(\beta(p)\) are differentiable. The cost function \(\kappa(p)\) can be scalar or vector-valued. Examples are \(\kappa(p) = p\) or \(\kappa(p) = p^T1\). In practice, the powers must be positive and there is a maximum
power that each transmitter can generate. Therefore, we will implicitly assume
that there are the natural constraints \( p \in D_p = \{ p : p_{\min} \leq p \leq p_{\max} \} \), where \( p_{\min} \geq 0 \) and \( p_{\max} \) are given constants.

The main problem this chapter is concerned with, is when the iterations
\[
p^{k+1} := \beta(p^k)
\] (6.3)
solve optimization problem (6.2). We show under which conditions the general
power control problem (5) is Fast-Lipschitz, which will allow us to establish the
optimality of iterations (6). In particular, when problem (6.2) is Fast-Lipschitz,
then the iterations (6.3) will converge to \( p^* = \beta(p^*) \) and \( p^* \) is optimal for problem
(6.2).

Remark 6.1. Note that the formulation of the iterations (6.3) is synchronous, i.e.,
every node must finish the computations and communications of round \( k \) before
the next round \( k + 1 \) can start. The algorithm we consider also converges asyn-
chronously, under the assumption of bounded delays \[1,46,47,49\]. However, since
convergence properties are not the main focus of this chapter, we restrict ourselves
to the less cumbersome synchronous notation of (6.3).

6.2 Two-sided scalable problems

In this section, we examine the relations between standard functions (in 6.2.2),
type-II standard functions (in 6.2.3), and Fast-Lipschitz optimization. This will
allow us to establish the core results that

1) all standard power control problems are Fast-Lipschitz;

2) there exist non standard interference functions whose fixed point is the optimal
of a power control problem; and

3) type-II power control iterations are optimal within some conditions.

We begin by briefly recalling the concept of two-sided scalability, which we will use
to put standard and type-II standard functions in the Fast-Lipschitz framework.

6.2.1 Preliminary results on two-sided scalable functions

This subsection presents preliminary results for the upcoming sections on standard
and type-II standard problems. The main result of this section is Lemma 6.4 which
will allow us to establish the contractivity of standard and type-II standard func-
tions. Contractivity is one of the main components of the Fast-Lipschitz qualifying
conditions.

Definition 6.2 (49). A function \( \beta(p) \) is two-sided scalable if for all \( c > 1 \) and all
\((1/c)p \leq q \leq cp\), it holds that
\[
(1/c)\beta(p) < \beta(q) < c\beta(p).
\] (6.4)
Proposition 6.3 ([49, Prop. 4]). If a function is either standard or type-II standard, then it is also two-sided scalable.

The following lemma shows that two-sided scalable functions are shrinking maps with gradients of one-norm less than one. The lemma is based on the logarithmic transformations proposed in [59]. It will be used in the main results of sections 6.2.2 and 6.2.3.

Lemma 6.4. Let \( x \triangleq \ln p \) and \( f(x) \triangleq \ln \beta(x) \). If \( \beta(p) \) is two-sided scalable, then
\[
\|f(x) - f(y)\|_\infty < \|x - y\|_\infty
\]
for all \( x, y \), and
\[
\|\nabla f(x)\|_1 < 1.
\]

Proof. By [49, Lemma 7], two-sided scalability implies
\[
\max_i \left\{ \max \left\{ \frac{\beta_i(x)}{\beta_i(y)}, \frac{\beta_i(y)}{\beta_i(x)} \right\} \right\} < \max_i \left\{ \max \left\{ \frac{x_i}{y_i}, \frac{y_i}{x_i} \right\} \right\}
\]
for all \( i \). Since the logarithm is strictly increasing, this is equivalent to
\[
\max_i \left\{ \max \left\{ \ln \left( \frac{\beta_i(p)}{\beta_i(q)} \right), \ln \left( \frac{\beta_i(q)}{\beta_i(p)} \right) \right\} \right\} < \max_i \left\{ \max \left\{ \ln \left( \frac{p_i}{q_i} \right), \ln \left( \frac{q_i}{p_i} \right) \right\} \right\}.
\]
Inserting \( p = e^x \) and \( q = e^y \) gives
\[
\max_i \left\{ |\ln \beta_i(e^x) - \ln \beta_i(e^y)| \right\} < \max_i \left\{ |\ln e^{x_i} - \ln e^{y_i}| \right\},
\]
\[
\Leftrightarrow |f_i(x) - f_i(y)| < |x_i - y_i|.
\]
Since this holds for all components \( i \), we have
\[
\|f(x) - f(y)\|_\infty < \|x - y\|_\infty
\]
by definition of the infinity norm. Denote \( v = \arg \max_{\|u\|_\infty = 1} \||\nabla f(x)^T u\|_\infty \). By definition, we have \( \|v\|_\infty = 1 \) and
\[
\|\nabla f(x)^T v\|_\infty = \||\nabla f(x)^T\|_\infty = \|\nabla f(x)\|_1.
\]
By defining \( y = x + \epsilon v \), with \( \epsilon \) positive scalar, we have
\[
1 > \frac{\|f(y) - f(x)\|_\infty}{\|y - x\|_\infty} = \frac{\|f(x + \epsilon v) - f(x)\|_\infty}{\|\epsilon v\|_\infty} = \frac{\|f(x + \epsilon v) - f(x)\|_\infty}{\epsilon},
\]
and in the limit \( \epsilon \to 0 \),
\[
1 > \lim_{\epsilon \to 0} \frac{\|f(y) - f(x)\|_\infty}{\|y - x\|_\infty} = \lim_{\epsilon \to 0} \frac{\|f(x + \epsilon v) - f(x)\|_\infty}{\epsilon} = \|\nabla f(x)^T v\|_\infty = \|\nabla f(x)\|_1.
\]
This concludes the proof. \( \square \)
The result shows that two-sided scalable functions are shrinking maps, but it does not establish the amount of slack in \( \| \nabla f(x) \|_1 < 1 \). This slack is useful for several reasons. The first reason is that any amount of slack makes the function \( f(x) \) a contraction, thereby guaranteeing a unique fixed point. This is assumed in the Fast-Lipschitz qualifying conditions, e.g., in \((GQC.b)\). The lack of knowledge on the amount of slack is not a problem in practice, since the bounds \( p_{\text{min}} \leq p \leq p_{\text{max}} \) form a closed bounded region of \( \mathbb{R}^n \). Therefore, \( \| \nabla f(x) \|_1 \) attains a minimum (call this value \( \alpha \)) in that region, so \( \| \nabla f(x) \|_1 \leq \alpha < 1 \) for those \( x \) that are of interest. Secondly (and more importantly), the qualifying conditions other than \( Q_1 \) typically require \( \| \nabla f(x) \|_\infty < c \) for some \( c \in (0, 1] \). Lemma \( 6.4 \) is therefore only of use if \( c = 1 \). This special case is exploited in Section \( 6.2.3 \). Now, based on this lemma, we are in the position to show that power control problems \( (6.2) \) with standard interference functions are a special case and Fast-Lipschitz optimization.

### 6.2.2 Standard functions and Fast-Lipschitz optimization

In this section we recall Yates’ framework of standard functions and show that problem \( (6.2) \) with standard interference function constraints has an equivalent problem formulation that is Fast-Lipschitz.

**Definition 6.5 \([9]\).** The function \( \beta(p) \) is *standard* if for all \( p, q \geq 0 \), the following properties are satisfied.

\[
\begin{align*}
\text{Positivity:} & \quad \beta(p) > 0 \quad (6.5a) \\
\text{Monotonicity:} & \quad p \geq q \Rightarrow \beta(p) \geq \beta(q) \quad (6.5b) \\
\text{Scalability:} & \quad c > 1 \Rightarrow \beta(cp) < c\beta(p) \quad (6.5c)
\end{align*}
\]

The monotonicity requirement \( (6.5b) \) can equivalently be formulated \( \nabla \beta(p) \geq 0 \) for all \( p \geq 0 \).

There is a relation between standard functions and two-sided scalable functions. Note that \( (6.4) \) multiplied by a positive scalar \( c > 0 \) implies \( (c^2 - 1) \beta(p) > 0 \), so any two-sided scalable function must also be positive, i.e., \( \beta(p) > 0 \) \([49, \text{Lemma 6}]\). If \( q \) in Definition \( 6.2 \) is chosen as \( q = cp \), inequalities \( (6.4) \) become

\[
(1/c)\beta(p) < \beta(cp) < c\beta(p), \quad (6.6)
\]

so a two-sided scalable function is always scalable \( (6.5c) \). Any two-sided scalable function \( \beta(p) \) is therefore standard if \( \nabla \beta(p) \geq 0 \).

**Proposition 6.6 \([9]\).** Assume that the power optimization problem \( (2.3) \) is feasible. Then, the standard interference function \( \beta(p) \) has a unique fixed point \( p^* \) that is the solution to \( (2.3) \).

We now show that problem \( (2.3) \) with standard interference constraints fall under the Fast-Lipschitz framework. To this end we consider problem \( (6.2) \), with the
general cost function $\kappa(p)$, which we assume be differentiable. All Fast-Lipschitz qualifying conditions require that the norm of the constraint function gradient be small enough. We will show this through Lemma 6.4 where we investigate problem (6.2) after a change of variables.

To this end, we let $x \triangleq \ln p$ as the logarithm of the power variables. This gives $p \triangleq e^x$ and the equivalent problems

$$\min_x \kappa(e^x) \quad \text{s.t.} \quad e^x = \beta(e^x)$$

and, because the logarithm is strictly increasing,

$$\min_x f_0(x) \triangleq \kappa(e^x) \quad \text{s.t.} \quad x \geq f(x) \triangleq \ln \beta(e^x).$$

(6.7)

If problem (6.7) is Fast-Lipschitz, then $x^* = f(x^*)$ is the unique Pareto optimal point for (6.7), whereby

$$e^{x^*} = p^* = \beta(p^*)$$

is optimal for problem (6.2).

The following result shows how power control problems with standard constraint functions, if differentiable, have an equivalent Fast-Lipschitz problem formulation.

**Theorem 6.7.** Consider problem (6.2) and let $\beta(p)$ be differentiable and standard. If $\nabla \kappa(p) \geq 0$ with non-zero rows, then the equivalent problem (6.7) is Fast-Lipschitz and $p^* = \beta(p^*)$ is optimal in problem (6.2).

**Proof.** We show that problem (6.7) is Fast-Lipschitz by qualifying condition Q1. The gradients of problem (6.7) are given by

$$\nabla f_0(x) = \text{diag}(p) \nabla \kappa(p),$$

(6.8a)

$$\nabla f(x) = \text{diag}(p) \nabla \beta(p) \text{diag}(1/\beta(p)),$$  

(6.8b)

where $p = e^x$ and $\text{diag}(1/\beta(p))_{ii} = 1/\beta_i(p)$. Since $p = e^x \geq 0$, $\beta(p) > 0$ and $\nabla \beta(p) \geq 0$, the gradients (6.8) fulfill $\nabla f(x) \geq 0$ and $\nabla f_0(x) \geq 0$ with non-zero rows (these are conditions [Q1.c] and [Q1.a] respectively). If $\beta(p)$ is standard, it is also two-sided scalable by Proposition 6.3 so $\|\nabla f(x)\|_1 < 1$ by Lemma 6.4 (condition [Q1.b]). Problem (6.7) is therefore Fast-Lipschitz by qualifying condition Q1, and $x^* = f(x^*)$. Taking the exponential of the previous relation gives $p^* = \beta(p^*)$. This concludes the proof. 

While Proposition 6.6 states that the fixed point of standard constraints minimize the powers in a Pareto sense, i.e., $\kappa(p) = p$, Theorem 6.7 accepts any non-decreasing $\kappa(p)$. The requirement that $\nabla \kappa(p)$ have non-zero rows simply means that each variable $p_i$ has an effect on at least one component of the cost at each
p. For scalar values cost functions, this is the same as requiring \( \kappa \) to be strictly increasing, \( \nabla \kappa (p) > 0 \).

Theorem 6.7 is not a generalization of Proposition 6.6 in practice, since a minimization of \( p \) is equivalent to a minimization of an increasing function of \( p \). The novelty here is instead that standard problems fall within the broader class of Fast-Lipschitz problems. Therefore, we can have non standard interference functions that can lead to optimally by distributed iterative power control algorithms. In the next subsection we will continue to show how type-II standard functions relates to Fast-Lipschitz optimization.

### 6.2.3 Type-II standard functions and Fast-Lipschitz optimization

As the standard functions are monotonically increasing, transmit nodes following (6.3) will always increase their power when their transmission environment is worsened by higher interference. A receiver node experiencing a deep fade will therefore need a very high transmit power, thereby increasing interference for the other receiver nodes in the network. This is not a good strategy, for example, in delay tolerant applications, where transmit nodes can adjust their transmission rates and higher throughput can be achieved by prioritizing receiver nodes experiencing low interference. One such strategy is to keep the signal-to-interference product constant, which results in update functions (6.3) that are monotonically decreasing, and therefore not standard. This is addressed in [49], where Sung and Leung extends Yates’ framework with type-II standard functions.

**Definition 6.8 ([49]).** The function \( \beta(p) \) is **type-II standard** if for all \( p, q \geq 0 \), the following properties are satisfied:

\[
\begin{align*}
\text{Type-II Monotonicity}: & \quad p \leq q \Rightarrow \beta(p) \geq \beta(q) \tag{6.9a} \\
\text{Type-II Scalability}: & \quad c > 1 \Rightarrow \beta(cp) > (1/c)\beta(p) \tag{6.9b}
\end{align*}
\]

As in the case of standard functions, the monotonicity property (6.9a) can be written \( \nabla \beta(p) \leq 0 \) for all \( p \). Note also from (6.6) that all two-sided scalable functions are type-II scalable (6.9b) so any two-sided scalable function \( \beta(p) \) where \( \nabla \beta(p) \leq 0 \) is also type-II standard. Type-II standard functions converge in the same way as standard functions - if a fixed point \( p^* \) exists, then iteration (6.3) converges to \( p^* \) [49 Theo. 3].

When considering opportunistic algorithms, \( \beta_i(p) \) no longer has the interpretation of “interference that receiver node \( i \) must overpower asking transmit node \( i \) to use a power \( p_i \) high enough”. There are no longer any explicit constraints \( p \geq \beta(p) \) underlying the algorithm, and \( \beta \) might not even have a physical meaning. The framework of two-sided scalable functions guarantees that the iterations (6.3) converge to a fix point also in the case of type-II standard interference functions, but the optimality meaning of this fixed point is no longer clear. Therefore, in the
6.2. Two-sided scalable problems

following we consider a function $\beta$ of type-II and assume it comes from a problem of the form (6.2). With the framework of Fast-Lipschitz optimization we characterize type-II standard power control problems to show that the fixed point is also optimal for optimization in the form (6.2). This is an important result that we can establish by Fast-Lipschitz optimization.

As in the of standard functions in Section (6.2.2), we examine the problem in logarithmic variables $x = \ln p$ and arrive at the equivalent problem (6.7), with gradients given by (6.8).

**Theorem 6.9.** Assume $\beta(p)$ be differentiable and type-II standard, and consider $f(x) = \ln \beta(e^x)$. Let $B = [B_{ij}]$ be a matrix with elements

$$B_{ij} = \max_x |\nabla_i f_j(x)| = \max_p \left| \nabla_i \beta_j(p) \frac{p_i}{\beta_j(p)} \right|,$$

and assume $\rho(B) < 1$. Let $c > 0$ be an arbitrary (positive) vector in $\mathbb{R}^n$ and let $h(z) \in \mathbb{R}^m$ be any strictly increasing function of one variable. Then, problem (6.2) is Fast-Lipschitz if

$$s = (I - B)^{-1} c \quad (6.10)$$

and

$$\kappa(p) = h(\prod_i p_i^{s_i}) \quad (6.11)$$

**Proof.** Since $\beta(p)$ is type-II standard, $\nabla \beta(p) \leq 0$ and $\beta(p) > 0$. The equivalent problem (6.7) therefore has a non-positive constraint gradient $\nabla f(x) \leq 0$ by equation (6.8b). Qualifying condition $Q_6$ now directly applies to show that problem 6.7 is Fast-Lipschitz. The result follows immediately from qualifying condition $Q_6$ in Section 3.4.

The form $\kappa(p) = h(z(p))$, with $h$ being an increasing function, implies that all cost functions $\kappa$ that can be handled with Theorem 6.9 are equivalent to the scalar cost $\kappa_0(p) = \prod_i p_i^{s_i}$ obtained when $h(z) = z$. If one instead chooses $h(z) = \ln z$ the cost becomes $\kappa(p) = s^T \ln p$, i.e., a weighted sum of the power logarithms. Equation (6.10) states that the weighting vector $s$ should lie in the interior of the cone spanned by the columns of $(I - B)^{-1}$.

The assumption $\rho(B) < 1$ is crucial for Theorem 6.9 to hold, since it guarantees that the vector $s$ of (6.10) is well defined and positive. The assumptions assure that $\rho(\nabla f(x)) \leq \|\nabla f(x)\|_1 < 1$, so $\rho(B) < 1$ surely holds if there is a point $x^B$ such that $B = |\nabla f(x^B)|$. This means that all elements of $\nabla f(x)$ are minimized at the common point $x^B$. The simplest case where this is true is when $\nabla f(x) = A^T$ is
constant. This requires an $\beta(p)$ of the form

$$
\beta_i(p) = \exp(A \ln p + b) = e^{bi} \exp \left( \sum_j A_{ij} \ln p_j \right) = e^{bi} \exp \left( \sum_j \ln p_j^{A_{ij}} \right)
$$

$$
= e^{bi} \exp \left( \ln \prod_j p_j^{A_{ij}} \right) = e^{bi} \prod_j p_j^{A_{ij}},
$$

i.e., $\beta_i(p)$ should be a monomial. If problem (6.2) has the basic cost function $\kappa_0(p) = \prod_i p_i^{x_i}$ from above, which also is a monomial, the problem is a geometric optimization problem [60]. Interestingly, geometric problems become convex with the change of variables $x \equiv \ln p$, the same variable transformation used throughout this section.

6.3 Non-monotonic interference functions

In the previous sections we examined two-sided scalable functions that where monotonically increasing (standard) and monotonically decreasing (type-II standard). In the following we give an example of a problem formulation where the constraints are not monotonic, hence neither standard, nor type-II standard. We show convergence and optimality through Fast-Lipschitz optimization, which was not known before.

The example builds upon the problem formulation in [61]. Once again we consider problem (6.2) and assume that the cost function $\kappa(p)$ is increasing in $p$. The formulation in [61] starts with the affine SINR model (1.5), but adds a stochastic channel and outage as follows. Let

$$
\beta_i(p) = \frac{\tau_i}{g_{ii}} \left( \sum_{j \neq i} g_{ij}p_j + \eta_i \right)
$$

represent the expected power needed to reach the SINR target $\tau_i$, and model the stochastic gain from transmitter $i$ to receiver $i$ by $g_{ii}\Theta_i$ where $\Theta_i$ is a stochastic variable describing the fading of the wireless channel. Furthermore, allow each transmitter to send only if the required power (to reach the SINR target) is lower than some bound $b$. Combining the two effects gives the new power control law

$$
\frac{p_i^{k+1}}{\theta_i} = h \left( \frac{\beta_i(p^k)}{\Theta_i} \right),
$$

where

$$
h(x) = \begin{cases} 
  x & \text{if } x \leq b, \\
  0 & \text{otherwise.}
\end{cases}
$$
The fast timescale of the fading $\Theta_i$ makes it hard to track and measure in practice. Instead, let each transmitter node update its transmit power according to the expectation (6.13), i.e.,

$$p_i^{k+1} = \mathbb{E}_{\Theta_i} \left[ h \left( \frac{\beta_i(p^k)}{\Theta_i} \right) \right] \triangleq \Phi_i \left( \beta_i(p^k) \right). \quad (6.14)$$

The expectation acts to smooth the discontinuous properties of $h(\cdot)$, and $\Phi_i(\beta_i(p))$ is called the smoothed interference function of node (or mobile equipment) $i$. The iterations in (6.14) can be seen as a possible solution algorithm for a power control problem of the type

$$\min \kappa(p) \quad \text{s.t.} \quad p_i \geq f_i(p) \triangleq \Phi_i(\beta_i(p)) \quad \forall i. \quad (6.15)$$

However, the nature of $h(x)$ will make $f_i(p)$ non-monotonic, regardless of underlying assumptions on $\beta_i(p)$. Therefore, neither the standard, nor the type-II standard interference function approach applies here. To study the convergence properties of iterations based on these functions, [61] introduces absolutely subhomogeneous functions, fulfilling

$$e^{-|a| \Phi(x)} \leq \Phi(e^{a}x) \leq e^{|a| \Phi(x)}$$

for every $x \geq 0$ and all scalars $a$. Note that absolute subhomogeneity is implied by two-sided scalability. In [61] it is shown that, if for each $i,$

- $\beta_i(p)$ is standard, and
- $\Phi_i(x) = \mathbb{E}_{\Theta_i} \left[ h(x/\Theta_i) \right]$ is bounded and absolutely subhomogeneous,

then the sequence (6.14) will converge to a fixed point. However, nothing is said in [61] about the optimality of this fixed point.

Our approach is to use Fast-Lipschitz optimization and qualifying condition $Q_3$, which has no requirements on the monotonicity of $f(x)$. Consider again problem (6.15). If $f(p) = [f_1(x), \ldots, f_n(x)]^T$ and $\kappa(p)$ fulfill (Q3,b), i.e., if $\nabla \kappa(p) > 0$ and

$$\|\nabla f(p)\|_\infty < \frac{q(p)}{1 + q(p)},$$

where

$$q(p) = \min_j \frac{\min_i \nabla_i \kappa_j(p)}{\max_i \nabla_i \kappa_j(p)},$$

then problem (6.15) is Fast-Lipschitz and the iterations (6.14) will converge to the optimal solution of (6.15). In the previous sections, we used properties of standard and type-II standard functions to show that the gradient norm $\|\nabla f\|_\infty$ was small enough. In this section, we obtain the bound directly from $\|\nabla \beta\|_\infty$ by using the following result:
Lemma 6.10. Let $\theta_j(y)$ be the pdf of the channel fading coefficient $\Theta_j$, consider $z > 0$ and define

$$\Omega_j(z) \triangleq \int_z^{\infty} \frac{\theta_j(y)}{y} dy - \theta_j(z).$$

(6.16)

Then, the infinity norm of the constraint function of problem (6.15) and the infinity norm of the underlying interference function $\beta(p)$ in (6.12) fulfill

$$||\nabla f(p)||_\infty \leq \max_i |\Omega_i(\beta_i(p)/b)| \cdot ||\nabla \beta(p)||_\infty.$$  

(6.17)

Proof. Dropping the explicit $p$-dependence of $\beta_j$ and $f_j$, we have

$$f_j = E_{\Theta_j}[h(\beta_j/\Theta_j)] = \int_0^{\infty} h(\beta_j/y)\theta_j(y) \, dy = \int_0^{\beta_j/b} h(\beta_j/y)\theta_j(y) \, dy + \int_{\beta_j/b}^{\infty} h(\beta_j/y)\theta_j(y) \, dy = \beta_j \int_{\beta_j/b}^{\infty} \frac{\theta_j(y)}{y} \, dy,$$

because

$$h(\beta_j/y) = \begin{cases} \beta_j/y & \text{if } \beta_j/y \leq b \iff y \geq \beta_j/b, \\ 0 & \text{otherwise.} \end{cases}$$

It follows that

$$\frac{df_j}{d\beta_j} = \frac{d}{d\beta_j} \left( \beta_j \int_{\beta_j/b}^{\infty} \frac{\theta_j(y)}{y} \, dy \right) = \int_{\beta_j/b}^{\infty} \frac{\theta_j(y)}{y} \, dy - \theta(\beta_j/b) \triangleq \Omega_j(\beta_j/b).$$

Returning to full notation, we have

$$\frac{\partial f_j(p)}{\partial p_i} = \frac{df_j(p)}{d\beta_j} \frac{\partial \beta_j(p)}{\partial p_i} = \Omega_j(\beta_j(p)/b) \frac{\partial \beta_j(p)}{\partial p_i}.$$

It follows that

$$||\nabla f(p)||_\infty = \max_j \left| \sum_i \left| \frac{\partial f_j(p)}{\partial p_i} \right| \right| \leq \max_j |\Omega_j(\beta_j(p)/b)| \cdot \max_i \left| \sum_j \left| \frac{\partial \beta_j(p)}{\partial p_i} \right| \right| \leq \max_j |\Omega_j(\beta_j(p)/b)| \cdot ||\nabla \beta(p)||_\infty,$$

as is stated by (6.17). This concludes the proof.

Note that Lemma 6.10 is true regardless of the underlying interference model $\beta(p)$, e.g., $\beta(p)$ does not need to be monotonic. We will use Lemma 6.10 in a simplified form as follows:
Corollary 6.11. Suppose optimization problem (6.2) fulfill qualifying condition (Q3.b) up to a scaling factor $\alpha > 0$, i.e., if

$$\alpha \|\beta(p)\|_{\infty} < \frac{q(p)}{1 + q(p)}.$$  

Then, optimization problem (6.15) is Fast-Lipschitz if

$$\max_{i,z} |\Omega_i(z)| \leq \alpha.$$  

This corollary allows us to say that problem (6.15), regardless the underlying interference model $\beta(p)$, is Fast-Lipschitz if

$$\max_{i,z} |\Omega_i(z)| < \frac{1}{\|\beta(p)\|_{\infty}} \frac{q(p)}{1 + q(p)} \quad \forall p.$$  

(6.18)

For fading coefficients from an arbitrary distribution, the function $\Omega_i(z)$ in equation (6.16) might not be expressed on closed form. However, the max-value of $\Omega_i(z)$ can be found through numerical calculations. We now apply Corollary 6.11 to two different distributions of the channel fading $\Theta$, one is analyzed analytically and one is studied numerically.

6.3.1 Fading models

In what follows we consider two different fading models. First we investigate the case where the channel fading coefficient $\Theta$ follows a Rayleigh distribution, whereby the worst-case value of $\Omega$ can be determined analytically. Thereafter, we investigate the case when $\Theta$ follows an exponential distribution. In this case we find the worst-case value of $\Omega$ through numeric calculation.

Rayleigh distribution

Assume $\Theta_i$ is follows a Rayleigh distribution with parameter $\lambda_i$ and with pdf

$$\theta_i(y) = \frac{y}{\lambda_i^2} e^{-y^2/2\lambda_i^2}, \quad \lambda_i > 0.$$  

(6.19)

Recalling the definition of $\Omega_i(z)$ in (6.16), we calculate the first term of $\Omega_i(z)$ as

$$\int_{z}^{\infty} \frac{\theta_i(y)}{y} dy = \int_{z}^{\infty} \frac{e^{-y^2/2\lambda_i^2}}{\lambda_i^2} dy.$$  

By the substitution $y = \sqrt{2\lambda_i} t$ we get $dy = \sqrt{2\lambda_i} dt$ and

$$\int_{z}^{\infty} \frac{\theta_i(y)}{y} dy = \int_{z/\sqrt{2\lambda_i}}^{\infty} \frac{e^{-t^2}}{\lambda_i^2} \sqrt{2\lambda_i} dt = \sqrt{\frac{\pi}{2\lambda_i}} \text{erfc} \left( \frac{z}{\sqrt{2\lambda_i}} \right),$$  

where $\text{erfc}$ is the complementary error function.
where erfc(·) is the complementary error function. Therefore, we have

\[
\Omega_i(z) = \sqrt{\frac{\pi}{2\lambda_i^2}} \text{erfc} \left( \frac{z}{\sqrt{2}\lambda_i} \right) - \frac{z}{\lambda_i^2} e^{-z^2/2\lambda_i^2}.
\]

and

\[
\frac{d\Omega_i(z)}{dz} = \frac{e^{-z^2/2\lambda_i^2}}{\lambda_i^2} \left( 2 - \frac{z^2}{\lambda_i^2} \right),
\]

which is smooth, and equal to zero only when \( z = \sqrt{2}\lambda_i \). Therefore, the extreme values of \( \Omega_i(z) \) must occur as \( z \to 0 \), \( z = \sqrt{2}\lambda_i \), or \( z \to \infty \). Evaluating \( \Omega_i \) at these points gives

\[
\Omega_i(z) \to \sqrt{\frac{\pi}{2}} \frac{1}{\lambda_i}
\]
as \( z \to 0 \),

\[
\Omega_i(\sqrt{2}\lambda_i) = \frac{1}{\lambda_i} \left( \sqrt{\frac{\pi}{2}} \text{erfc}(1) - \sqrt{2} e^{-1} \right) > - \frac{1}{3\lambda_i}, \approx -0.323
\]
and \( \Omega_i(z) \to 0 \) as \( z \to \infty \) respectively. It follows that \( \max_{i,z} |\Omega_i(z)| \leq \alpha \) if

\[
\alpha \geq \max \left\{ \sqrt{\frac{\pi}{2}} \frac{1}{\lambda_i}, \frac{1}{3} \frac{1}{\lambda_i} \right\} \Leftrightarrow \lambda_i \geq \sqrt{\frac{\pi}{2}} \frac{1}{\alpha}
\]
for all \( i \). This means that if

a) the original (deterministic and outage-free) problem \( 6.2 \) is Fast-Lipschitz by qualifying condition \( Q_3 \), i.e., \( \alpha \leq 1 \) in Corollary \( 6.11 \) and

b) the channel fading \( \Theta_i \) follows a Rayleigh distribution \( 6.19 \) with parameter \( \lambda_i \geq \sqrt{\pi/2} \),

then problem \( 6.15 \) is Fast-Lipschitz by Corollary \( 6.11 \). It follows that the iterations \( 6.14 \) converge to \( p^* \), and \( p^* \) is the unique optimal solution of the optimization problem \( 6.15 \).

**Exponential distribution**

We now give an application of Corollary \( 6.11 \) to the case when the channel fading coefficients \( \Theta_i \) are exponentially distributed,

\[
\Theta_i \sim \theta_i(y | \lambda) = \lambda e^{-\lambda y}.
\]

(6.20)

This is known as Rayleigh fading. Denote \( z \triangleq \beta(p)/b \) (we will drop the transmitter index \( i \) to get a clearer notation), and highlight the \( \lambda \)-dependence of \( \Omega \) by writing

\[
\Omega(z, \lambda) = \int_{\frac{z}{\lambda}}^{\infty} \frac{\theta(y)}{y} dy - \theta(z | \lambda) = \lambda \left( \int_{\frac{z}{\lambda}}^{\infty} \frac{e^{-t}}{t} dt - e^{-\lambda z} \right) = \lambda \phi(\lambda z)
\]

64
6.3. Non-monotonic interference functions

Figure 6.1: The smoothed mobile behavior function $\Phi(\beta_i)$ for different $\lambda_i$, when $\Theta_i$ follows a Rayleigh distribution. The dashed lines show the best approximations that are absolutely subhomogeneous, as required in [61]. The dotted line shows the function $h_m(x)$ when $b = 1$.

Figure 6.2: This figure show the behaviour of $\Omega_i(z)$ for different $\lambda_i$. When $\alpha = 1$, $\lambda_i = \sqrt{\pi/2} \approx 1.26$ is the lower limit of $\lambda_i$ for which Corollary 6.11 applies (i.e., $|\Omega_i(z)| < 1 \forall z$).
6. Transmit Radio Power Control

where

\[ \psi(\xi) \triangleq \int_{\xi}^{\infty} \frac{e^{-t}}{t} dt - e^{-\xi}. \]  

(6.21)

The function \( \psi(\xi) \) is shown in Figure 6.3.

To use the result in (6.18) we must show that the absolute value of \( \Omega \) is small enough. We will see that this is typically the case, except when \( z = \beta/b \) goes to zero. This cannot happen in practice, since the non-zero background noise \( \eta \) always lower bounds the interference. Therefore, we assume that \( z = \beta/b \) is lower bounded by some \( z_{\text{min}} \). For any given lower bound \( z_{\text{min}} \), introduce

\[ \sigma_{z_{\text{min}}} (\lambda) = \max_{z \geq z_{\text{min}}} |\Omega(z, \lambda)|. \]

The function \( \sigma_{z_{\text{min}}} \) is the worst case value over all possible values of \( \lambda \), given that \( z \geq z_{\text{min}} \). To find \( \sigma_{z_{\text{min}}} (\lambda) \), let \( \xi = \lambda z \), whereby

\[ |\Omega(z, \lambda)| = \lambda |\psi(\lambda z)| = \lambda |\psi(\xi)|. \]

For a fixed \( \lambda \), it is sufficient to find the \( z \geq z_{\text{min}} \) that maximizes \( |\psi(\lambda z)| \) or, equivalently, the \( \xi \geq \xi_{\text{min}} = \lambda z_{\text{min}} \) that maximizes \( |\psi(\xi)| \). Consider the plot of \( \psi(\xi) \) is shown in Figure 6.3. The derivative

\[ \frac{d\psi}{d\xi} = e^{-\xi} \left( 1 - \frac{1}{\xi} \right) \]

is zero only when \( \xi = 1 \), and the second derivative is always positive. The dashed lines highlight where \( \xi = 1 \) and

\[ \xi = \xi_1 = \{ t : \psi(t) = -\psi(1) \}. \]
In order to maximize $|\psi(\xi)|$, it is clear that $\xi$ should be chosen as

$$\xi = \begin{cases} 1 & \text{if } \xi_1 \leq \xi_{\min} \leq 1, \\ \xi_{\min} & \text{otherwise.} \end{cases}$$

In terms of the variables $\lambda$ and $z$ we therefore have

$$\sigma_{z_{\min}}(\lambda) = \max_{z \geq z_{\min}} |\Omega(z, \lambda)| = \begin{cases} \lambda \psi(\lambda z_{\min}) = \Omega(z_{\min}, \lambda), & \text{if } \lambda < \frac{\xi_1}{z_{\min}}, \\ -\lambda \psi(1), & \text{if } \frac{\xi_1}{z_{\min}} \leq \lambda \leq \frac{1}{z_{\min}}, \\ -\lambda \psi(\lambda z_{\min}) = -\Omega(z_{\min}, \lambda), & \text{if } \frac{1}{z_{\min}} < \lambda. \end{cases}$$

It is clear that any stationary point of $\sigma_{z_{\min}}$ must also be a stationary point of $\Omega(z_{\min}, \lambda)$, with derivative

$$\frac{d\Omega(z_{\min}, \lambda)}{d\lambda} = \frac{d}{d\lambda} (\lambda \psi(\lambda z_{\min})) = \psi(\lambda z_{\min}) + \lambda \frac{d\psi(\lambda z_{\min})}{d\lambda}$$

$$= \psi(\lambda z_{\min}) + \lambda \left( z_{\min} e^{-\lambda z_{\min}} (1 - \frac{1}{\lambda z_{\min}}) \right)$$

$$= \psi(\lambda z_{\min}) + e^{-\lambda z_{\min}} (\lambda z_{\min} - 1).$$

Setting the expression above to zero and solving numerically gives the two solutions

$$\begin{cases} \lambda z_{\min} = v_1 \approx 0.1184 & \text{and} \\ \lambda z_{\min} = v_2 \approx 1.5656, \end{cases}$$

i.e., when $\lambda = v_1/z_{\min}$ and $\lambda = v_2/z_{\min}$. Inserting these values into $\sigma_{z_{\min}}(\lambda)$ gives the values

$$\begin{cases} \sigma_{z_{\min}}(\frac{v_1}{z_{\min}}) = \frac{v_1}{z_{\min}} \psi(v_1) \approx 0.093 \approx \frac{z_{\min}}{z_{\min}} & \text{and} \\ \sigma_{z_{\min}}(\frac{v_2}{z_{\min}}) = \frac{v_2}{z_{\min}} \psi(v_2) \approx 0.185 \approx \frac{z_{\min}}{z_{\min}} \end{cases}$$

of the two local maxima shown in the Figure 6.4. Assuming $z \geq z_{\min}$, we therefore have

$$|\Omega(z, \lambda)| \leq \max_{z \geq z_{\min}} |\Omega(\lambda, z)| = \sigma_{z_{\min}}(\lambda) \leq 0.185/z_{\min}$$

for any parameter value $\lambda$ of the fading coefficient distribution parameter. In particular, Corollary 6.11 states that problem (6.15) is Fast-Lipschitz if

$$\|\nabla \beta(p)\|_{\infty} < \frac{0.185}{z_{\min}} \frac{q(p)}{1 + q(p)}$$

for all $p \geq 0$, where

$$q(p) = \min_j \frac{\min_i \nabla_i \kappa_j(p)}{\max_i \nabla_i \kappa_j(p)}.$$
6. Transmit Radio Power Control

\[\sigma_{z_{\min}}(\lambda)\]

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure64.png}
\caption{Plot of \(\sigma_{z_{\min}}(\lambda)\), note that the x-axis is scaled by \(z_{\min}\).
\end{figure}

is given by the characteristics of the cost function \(\kappa(p)\).

This example has showed how power control problems without monotonicity properties can be analyzed by Fast-Lipschitz optimization. The price one has to pay to ensure optimality is the tighter bound

\[\|\nabla \beta(p)\|_\infty < q(p)/(1 + q(p)) \leq 1/2,\]

as opposed to requiring \(\|\nabla \beta(p)\|_\infty < 1\) for monotonic functions, which is sufficient to show contractivity.

6.4 Summary

In this chapter we applied Fast-Lipschitz optimization to transmit power control problems. First, we examined the conditions under which power control algorithms with standard and type-II standard functions fall under the Fast-Lipschitz framework. It was shown that standard problems have an equivalent problem (after a logarithmic change of variables) that is Fast-Lipschitz by qualifying condition \(Q_{11}\).

Similarly, provided that the extra conditions of Theorem 6.9 hold, certain type-II standard problems have an equivalent problem that is Fast-Lipschitz. Finally, we studied problems that were non-monotonic and therefore neither standard, nor type-II standard. Instead, the analysis relied on qualifying condition \(Q_{23}\), which replaces the requirement of monotonicity with a stricter requirement on the infinity norm of the constraint function gradient. The analysis of this chapter allowed us
to give the studied problems a richer notion of optimality. Specifically, we classified cost functions for which the fixed point algorithm provides the optimal solution.
Chapter 7

Illustrative Examples

In this chapter we illustrate and apply the theory developed in the previous chapters. We begin with a simple illustration of how the concepts of Fast-Lipschitz optimization can be applied in Section 7.1. Similarly, Section 7.2 illustrates the results of Chapter 5 for a convergence speed comparison. Finally, Section 7.3 applies contributions of this thesis to solve a non-convex generic optimal control problem.

7.1 Simple non-convex example

Consider the problem

$$\begin{align*}
\text{max} & \quad f_0(x) \\
\text{s.t.} & \quad x \leq f(x) \\
& \quad x \in X = \{x : 0 \leq x \leq 1\},
\end{align*}$$

(7.1)

where $x \in \mathbb{R}^2$,

$$
f_0(x) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 2x_2 \end{bmatrix}, \quad \text{and} \quad f(x) = 0.5 \begin{bmatrix} 1 + ax_2^2 \\ 1 + bx_1^2 \end{bmatrix}. $$

If either $a$ or $b$ are positive, the problem is not convex (the canonical constraint functions $x_i - f_i(x)$ become concave).

At this point we do not know if $x^* = f(x^*) \in X$. However, following the results of Section 7.2 we can assume this is the case and examine whether

$$\begin{align*}
\text{max} & \quad f_0(x) \\
\text{s.t.} & \quad x \leq f(x)
\end{align*}$$

(7.2)

fulfills GQC (or Q5). The qualifying conditions must apply in the box $D$, which we select equal to $X$ (no point outside of $D = X$ is feasible, hence all feasible points lie in $D$).

71
The gradients of the objective and constraint functions are
\[
\nabla f_0 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad \nabla f = \begin{bmatrix} 0 & bx_1 \\ ax_2 & 0 \end{bmatrix}.
\]

Since \( \nabla f_0 > 0 \) for all \( x \), the assumptions \((GQC.a)\) on the objective function is always fulfilled. We will now check all sign combinations of \( a \) and \( b \).

\( a, b \geq 0 \): If both \( a \) and \( b \) are non-negative, then \( \nabla f(x) \geq 0 \) for all \( x \) in \( D \) and condition \((Q_1.c)\) holds. To verify condition \((Q_1.b)\), one must find a norm \( \| \cdot \| \) such that \( \| \nabla f(x) \| < 1 \) for all \( x \) in \( D \). When the \( \infty \)-norm is used, we get
\[
\max_{x \in D} \| \nabla f(x) \|_\infty = \max_{x \in D} \{ |bx_1|, |ax_2| \} \leq \max \{ b, a \} < 1
\]
if \( a, b < 1 \). Thus, when \( 0 \leq a, b < 1 \), the problem is Fast-Lipschitz by \( Q_1 \).

\( a, b \leq 0 \): If \( a \) and \( b \) are instead non-positive, we have \( \nabla f(x) \leq 0 \) for all \( x \) in \( D \) and condition \((Q_4.b)\) is fulfilled. In order to verify \((Q_4.c)\) we need \( \delta(x) \) and \( \Delta(x) \). These are defined pointwise in \( x \), as the smallest and largest (in absolute value) element of \( \nabla f_0 \), i.e., \( \delta(x) = 1 \) and \( \Delta(x) = 2 \) for all \( x \). Condition \((Q_4.c)\) now requires, for all \( x \) in \( D \), that
\[
\| \nabla f(x) \|_\infty < \frac{\delta(x)}{\Delta(x)} = \frac{1}{2}.
\]
From the previous case we know that \( \| \nabla f(x) \|_\infty \leq \max \{ |a|, |b| \} \) for all \( x \in D \), so the problem is guaranteed Fast-Lipschitz by qualifying condition \( Q_4 \), provided that \( -1/2 < a, b \leq 0 \). Note that this would not have met the old case (ii) in (2.8d), since \( f_0(x) \neq c^T x \).

\( ab < 0 \): When \( a \) and \( b \) have different signs, neither \((Q_1.c)\) nor \((Q_2.b)\) holds. Instead, one can try qualifying condition \( Q_5 \) which does not place any sign restrictions on \( \nabla f(x) \). It is only required, in \((Q_3.b)\), that
\[
\| \nabla f(x) \|_\infty < \frac{\delta(x)}{\delta(x) + \Delta(x)}.
\]
These quantities are unchanged from the previous cases, so \( \delta(x) = 1, \Delta(x) = 2 \) and \( \| \nabla f(x) \|_\infty \leq \max \{ |a|, |b| \} \), so the problem is Fast-Lipschitz by qualifying condition \( Q_5 \) if both \( |a| \) and \( |b| \) are less than 1/3. Also in this case, the Old Qualifying Conditions would not have worked since case (iii) in (2.8g) requires a scalar objective function.
7.1. Simple non-convex example

Figure 7.1: Illustration of the feasible region of problem (7.2) with $a = -0.3$ and $b = 0.3$ (making the feasible region non-convex). The iterates (7.3) of the solution quickly converges to $x^* = f(x^*)$, where all constraints are active.

Figure 7.2: Convergence of the iterates (7.3), measured in the $\infty$-norm. The convergence is geometric (linear in the log domain) and the optimal solution is found within an accuracy of $10^{-6}$ after 8 iterations.
Solution of the problem  

If problem (7.2) is Fast-Lipschitz by any of the cases above, the optimal point $x^\star$ is found by solving $x^\star = f(x^\star)$. We now solve the problem when $a = -0.3$ and $b = 0.3$, by iterating

$$x^{k+1} := f(x^k).$$

(7.3)

This sequence will converge to $x^\star = f(x^\star)$ since the qualifying conditions imply that $f$ is contractive (Lemma 3.4). The iterates $x^k$ of (7.3), together with the feasible region of the problem is shown in Figure 7.1. Clearly, $x^\star = f(x^\star) \in X$, so Proposition 4.2 applies and $x^\star$ is optimal also for the original problem (7.1). The convergence the iterations (7.3) is shown in Figure 7.2.

7.2 Example of convergence comparison

In this section we now show how to apply the theoretical results of Chapter 5 by a simple numerical example. Recall that the results required knowledge of the optimal primal and dual variables $x^\star$ and $\lambda^\star$ in advance, which is not possible in practice. To circumvent this issue, we consider the result of Corollary 5.7. Specifically, we consider the linearly constrained quadratic problem:

$$\min f_0(x),$$

s.t. $h(x) = 0,$

$$x \in D = \{x : 0 \leq x \leq 1\},$$

where

$$f_0(x) = \frac{1}{2}x^T \begin{bmatrix} 7 & 3 \\ 3 & 6 \end{bmatrix} x - \begin{bmatrix} 1 & 1 \end{bmatrix} x$$

and

$$h(x) = \begin{bmatrix} 0.75 & -0.1 \\ -0.05 & 0.8 \end{bmatrix} x - \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}.$$

We remark that the results of Chapter 5 apply to problems of any dimension. We have chosen a simple problem with two decision variables and two constraints for illustrative purposes.

The equivalent optimization problem in Fast-Lipschitz form is

$$\max -f_0(x),$$

s.t. $x = f(x), \quad x \in D,$

(7.4)

where

$$f(x) = x - \beta h(x).$$

(7.5)

We will test two different choices for $\beta$, one corresponds to ignoring the possibility to tune $\beta$, i.e., setting $\beta = 1$, and the other one consists in choosing an optimal value. For the first choice, we get

$$f^{(1)}(x) = x - 1 \cdot h(x) = \begin{bmatrix} 0.25 & 0.1 \\ 0.05 & 0.2 \end{bmatrix} x + \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}.$$
7.2. Example of convergence comparison

By the theory of Chapter 3, one can show that the rewritten problem in Fast-Lipschitz form is Fast-Lipschitz. In particular,

\[
\nabla (-f_0(x)) = \frac{1}{2} \begin{bmatrix} 7x_1 + 3x_2 + 2 \\ 3x_1 + 6x_2 + 2 \end{bmatrix} > 0 \quad \forall x \geq 0,
\]

so condition \((Q_1.a)\) holds. Furthermore,

\[
\nabla \left( f^{(1)}(x) \right) = \begin{bmatrix} 0.25 & 0.05 \\ 0.1 & 0.2 \end{bmatrix} \geq 0 \quad \forall x,
\]

so condition \((Q_1.c)\) holds. Finally,

\[
\left\| \nabla f^{(1)}(x) \right\|_\infty = \max \{ |0.25| + |0.05|, |0.1| + |0.2| \} = 0.3,
\]

so \((Q_1.b)\) is fulfilled, whereby the problem is guaranteed Fast-Lipschitz by qualifying condition \(Q_1\).

If we instead apply Lemma 5.2 to the gradient of equation (7.5), we can choose \(\beta = 1.29\) as the \(\beta\) minimizing \(\rho(\nabla f)\),

\[
f^{(\beta)}(x) = x - 1.29 \cdot h(x) \approx \begin{bmatrix} 0.03 & 0.13 \\ 0.06 & -0.03 \end{bmatrix} x + \begin{bmatrix} 0.52 \\ 0.77 \end{bmatrix}.
\]

Again, one can see that the problem is Fast-Lipschitz by checking the qualifying conditions. This time condition \((Q_1.c)\) no longer applies, because

\[
\nabla \left( f^{(\beta)}(x) \right) \approx \begin{bmatrix} 0.03 & 0.06 \\ 0.13 & -0.03 \end{bmatrix} \not\geq 0.
\]

Instead, the special structure of \(\nabla \left( f^{(\beta)}(x) \right)\) gives

\[
\left( \nabla f^{(\beta)}(x) \right)^2 \approx \begin{bmatrix} 0.009 & 0 \\ 0 & 0.009 \end{bmatrix} \geq 0,
\]

so condition \((Q_4.b)\) is fulfilled. It remains to verify condition \((Q_4.a)\) (same as \((Q_1.a)\)) and condition \((Q_4.c)\) The infinity norm of \(\nabla \left( f^{(\beta)}(x) \right)\) is

\[
\left\| \nabla f^{(\beta)}(x) \right\|_\infty \approx |0.13| + |-0.03| = 0.16.
\]

For every \(x\), we have

\[
\delta(x) = \min_{i,j} [\nabla f_0(x)]_{ij} = \min_i \frac{1}{2} \begin{bmatrix} 7x_1 + 3x_2 + 2 \\ 3x_1 + 6x_2 + 2 \end{bmatrix}
\]

and

\[
\Delta(x) = \max_{i,j} [\nabla f_0(x)]_{ij} = \max_i \frac{1}{2} \begin{bmatrix} 7x_1 + 3x_2 + 2 \\ 3x_1 + 6x_2 + 2 \end{bmatrix}.
\]
7. Illustrative Examples

The smallest ratio \( \delta(x)/\Delta(x) \) on feasible set \( \mathcal{D} \) is therefore given by

\[
\min_{x \in \mathcal{D}} \frac{\delta(x)}{\Delta(x)} = \min_{x \in \mathcal{D}} \left\{ \frac{7x_1 + 3x_2 + 2}{3x_1 + 6x_2 + 2}, \frac{3x_1 + 6x_2 + 2}{7x_1 + 3x_2 + 2} \right\} = \frac{5}{9} \approx 0.56.
\]

We therefore have \( \| \nabla f^{(\beta)}(x) \|_\infty < \delta(x)/\Delta(x) \) for all \( x \) in \( \mathcal{D} \) and condition (Q4.c) holds, so the problem is Fast-Lipschitz by qualifying condition Q4.

By now, we have shown that the original problem (7.4) can also be solved as either one of two Fast-Lipschitz problems. In order to use Proposition 5.6 we need

\[
\text{trace}(Q) = \text{trace}(\nabla^2 f_0(x)) = \text{trace} \left( \begin{bmatrix} 7 & 3 \\ 3 & 6 \end{bmatrix} \right) = 13,
\]

and the spectral radii of \( \nabla (f^{(1)}(x^*)) \) and \( \nabla (f^{(\beta)}(x^*)) \), which are \( \rho_{f,1} = 0.30 \) and \( \rho_{f,\beta} \approx 0.10 \) respectively. As the original problem has two variables and two constraints, the dimension of \( B \) is \( m = 4 \).

Proposition 5.6 now guarantees that both the Fast-Lipschitz methods will outperform the first order Lagrangian method, since

\[
3.25 = \frac{\text{trace}(Q)}{m} > \frac{1}{1^2} \frac{(1 + \rho_{f,1})^3}{1 - \rho_{f,1}} \approx 3.14,
\]

and

\[
3.25 = \frac{\text{trace}(Q)}{m} > \frac{1}{\beta^2} \frac{(1 + \rho_{f,\beta})^3}{1 - \rho_{f,\beta}} \approx 0.88.
\]

To verify these results, the problem is solved by Lagrangian method and the two Fast-Lipschitz methods respectively. The initial primal point was randomly chosen to \( x_0 \approx [0.13 \ 0.08]^T \), and the initial dual point was set to \( \lambda_0 = 0 \). For the Lagrangian method, the optimal step length \( \hat{\alpha} \approx 0.2075 \), given by Lemma 5.2 was used in each iteration.

The solution converged by the Lagrangian method after 840 iterations within a max-norm error of \( 10^{-4} \). The Fast-Lipschitz problems were solved by iterating \( x^{k+1} = f(x^k) \), once for \( f = f^{(1)} \) and once for \( f = f^{(\beta)} \). The Fast-Lipschitz methods converged after 8 and 4 iterations, respectively, within the error tolerance used for the Lagrangian method.

The results from the three methods are plotted in Figure 7.3. Also shown, is the lower bound of the Lagrangian method convergence given by Lemma 5.5. From the figure it should be clear that this bound can be very conservative.

7.3 Optimal control example

Consider a dynamical system with a state \( s \in \mathbb{R}^n \) and control variable \( u \in \mathbb{R}^p \). The state evolves in discrete time and the state at time instance \( i + 1 \) is given by

\[
s^{i+1} = f(s^i, u^i) + w^i, \quad (7.6)
\]
7.3. Optimal control example

Figure 7.3: Log-log plot of $\|x^k - x^*\|_\infty$, the max-norm distance from the $k$th iterate to the optimal point. "LM" denotes the Lagrangian method, "FL" denotes the Fast-Lipschitz methods and "Bound" shows the lower bound on LM convergence given by Lemma 5.5.

where $f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ and $w^i \in \mathbb{R}^n$ is an additive, bounded disturbance. The control variables are required to be positive and bounded, i.e., $0 \leq u \leq u_{\text{max}}$. As a consequence of the bounded disturbance and control variables, the state $s^N$ after a finite number of $N$ iterations also remains bounded. At each time instance, the system has a cost $g(s^i, u^i)$ that is strictly increasing in all variables, i.e., all states represent something expensive and all controls are naturally associated with a positive cost. The design objective is to choose the control inputs $\{u^i\}_{i=1}^N$ that minimizes the accumulated cost over $N$ periods. The optimal control problem, given the disturbances $\{w^i\}_{i=1}^N$ and the initial state $s_{\text{init}}$, becomes

$$\min_{\{u^i\}_{i=1}^N} \sum_{i=1}^N g(s^i, u^i)$$

s.t. $s^1 = s_{\text{init}}$

$$s^{i+1} = f(s^i, u^i) + w^i, \quad i = 1, \ldots, N-1,$$

$$0 \leq u^i \leq u_{\text{max}}, \quad i = 1, \ldots, N.$$  \hfill (7.7)

Note that this can be seen as a centralized or as a distributed optimization problem. Suppose that $f(s, u)$ is increasing in $u$. The optimal control would then trivially be $u^i = 0$ for all $i$, since the cost $g(s^i, u^i)$ at time $i$ increases with $u^i$ and the cost at time $i + 1$ increases with $s^{i+1}$, which in turn increases with $u^i$. On the other hand, if $f(s, u)$ is instead decreasing in $u$, there would be a tradeoff between choosing a small $u^i$ in order to make $g(s^i, u^i)$ small, or a large $u^i$ to make $g(s^{i+1}, u^{i+1})$ small. Throughout the rest of this example we will assume the non-trivial case when $f(s, u)$ is decreasing in $u$. With Fast-Lipschitz optimization it is possible to
7. Illustrative Examples

determine conditions on the costs and dynamics of problem (7.7), under which the optimal control is simply given by \( u_i^* = 0 \) for all \( i \). This is a very useful result, which allows to substantially simplify the computation of the control decision. The following result is based on Proposition 4.3 and applies to dynamics given by a general function \( f(s, u) \):

**Result 7.1.** Consider problem (7.7) and assume that \( g(s, u) \) is increasing in all variables. Assume further that the pair \( \nabla_s f(s, u) \) and \( \nabla_s g(s, u) \) fulfill the GQC (in place of \( \nabla f(x) \) and \( \nabla f_0(x) \) respectively) and

\[
\frac{\max_{s,u} \|\nabla u f(s, u)\|_\infty}{1 - \max_{s,u} \|\nabla s f(s, u)(s, u)\|_\infty} < \frac{\min_{s,u} \min_i [\nabla u g(s, u)]_i}{\max_{s,u} \max_i [\nabla s g(s, u)]_i}
\]  

(7.8)

for all allowed \( u \) and all reachable \( s \). Then, the optimal solution \( \{u_i^*\}_{i=1}^N \) is given by \( u_i^* = 0 \) for all \( i \), regardless the problem horizon \( N \), the initial state \( s_{\text{init}} \), and the disturbances \( \{w_i\}_{i=1}^N \).

The result can be derived be considering a Fast-Lipschitz problem equivalent to (7.7) as follows. Introduce the vectors

\[
y = \begin{bmatrix} -s^1 \\ \vdots \\ -s^N \end{bmatrix} \in \mathbb{R}^{nN}, \quad z = \begin{bmatrix} -u^1 \\ \vdots \\ -u^N \end{bmatrix} \in \mathbb{R}^{pN}, \quad \text{and} \quad \bar{w} = \begin{bmatrix} w^1 \\ \vdots \\ w^N \end{bmatrix} \in \mathbb{R}^{nN},
\]

and let \( y^i = -s^i \), \( z^i = -u^i \). Furthermore, let \( x = [y^T \quad z^T]^T \). Problem (7.7) can then be transformed to the equivalent maximization problem

\[
\max_x f_0(x) \quad \text{s.t.} \quad y = f_y(x), \quad z_{\text{min}} \leq z \leq 0,
\]

(7.9)

where the \( i \)th \( (n \times 1) \)-block of \( f_y(x) \) is given by

\[
f_y[i](x) = \begin{cases} -s_{\text{init}} & i = 1 \\ -f(-y^i, -z^i) - w^i & i = 2, \ldots, N, \end{cases}
\]

and \( f_0(x) = -\sum_{i=1}^N g(-y_i, -z_i) \). By Corollary 4.2 and Lemma 3.5 we know that problem (7.9) is Fast-Lipschitz if the relaxed problem (obtained by replacing the equality constraints by inequality constraints, and holding the constraint \( z \geq z_{\text{min}} \) implicit)

\[
\max_x f_0(x) \quad \text{s.t.} \quad y \leq f_y(x), \quad z \leq 0,
\]

(7.10)

is Fast-Lipschitz. This is precisely a problem in the form of (4.5) in Section 4.3 wherefore Proposition 4.3 applies.
7.3. Optimal control example

To see this, we first denote the gradients of the original system dynamics \(7.6\) by \(\nabla_s f(s, u) \triangleq A(s, u)\) and \(\nabla_u f(s, u) \triangleq B(s, u)\). Simple calculations show that the gradient \(\nabla_y f_y(x)\) consists of \(N \times N\) blocks, each of dimension \(n \times n\). All blocks are zero except the block sub-diagonal, which is given by the blocks

\[
\begin{pmatrix}
A(-y^1, -z^1), \ldots, A(-y^{N-1}, -z^{N-1})
\end{pmatrix},
\]
i.e.,

\[
\nabla_y f_y(x) = \begin{bmatrix}
0 \\
A(-y^1, -z^1) & \cdots \\
\vdots & \ddots & \cdots \\
A(-y^{N-1}, -z^{N-1}) & 0
\end{bmatrix}. \quad (7.11)
\]

Similarly, the gradient \(\nabla_z f_y(x)\) consists of \(N \times N\) blocks of dimension \(n \times p\), with

\[
\begin{pmatrix}
B(-y^1, -z^1), \ldots, B(-y^{N-1}, -z^{N-1})
\end{pmatrix}
\]
on the block sub-diagonal and zeros everywhere else.

In order to use Proposition 4.3, one must first verify that the subproblem \((f_0|x, f_y|z)\), obtained by fixing \(z\) in problem \(7.10\), is Fast-Lipschitz for all permissible \(z\). This can be done by showing that \(\nabla_y f_y(y, z)\) and \(\nabla_y f_0(y, z)\) fulfills one of the qualifying conditions of Chapter 3. Next, one must verify part (b) of Proposition 4.3. As we have restricted ourselves to the non-trivial case when \(f(s, u)\) is decreasing in \(u\), we have \(B(s, u) \leq 0\). Consequently, \(\nabla_z f_y(x) \geq 0\), and neither condition (b.i), nor condition (b.ii) applies. We therefore use condition (b.iii). Due to the block structure in \((7.11)\), we have

\[
\|\nabla_y f_y(x)\|_\infty \leq \max_{s,u} \|A(s, u)\|_\infty = \max_{s,u} \|\nabla_s f(s, u)\|_\infty
\]

and

\[
\|\nabla_z f_y(x)\|_\infty \leq \max_{s,u} \|B(s, u)\|_\infty = \max_{s,u} \|\nabla_u f(s, u)\|_\infty.
\]

The maximizations above are carried out over all permissible controls \(u\), and the corresponding achievable states \(s\). Furthermore, we need \(\Delta_y(x) = \max_{ij} [\nabla_y f_0(x)]_{ij}\) and \(\delta_z(x) = \min_{ij} [\nabla_z f_0(x)]_{ij}\). Note that the gradients \(\nabla_y f_0(x)\) and \(\nabla_z f_0(x)\) are column vectors since \(f_0\) is scalar. This gives

\[
\Delta_y(x) = \max_i [\nabla_y f_0(x)]_i \leq \max_{s,u} \max_i [\nabla_s g(s, u)]_i
\]

and

\[
\delta_z(x) = \min_i [\nabla_z f_0(x)]_i \leq \min_{s,u} \min_i [\nabla_u g(s, u)]_i,
\]

where the optimizations are over the permissible controls \(u\) and the reachable states \(s\). Condition (b.iii) is now fulfilled if

\[
\frac{\|\nabla_z f_y(x)\|_\infty}{1 - \|\nabla_y f_y(x)\|_\infty} < \frac{\delta_z(x)}{\Delta_y(x)},
\]
i.e., if
\[
\frac{\max_{s,u} \| \nabla f(s,u) \|_{\infty}}{1 - \max_{s,u} \| \nabla g(s,u) \|_{\infty}} < \frac{\min_{s,u} \min_{i} [\nabla u g(s,u)]_{i}}{\max_{s,u} \max_{i} [\nabla s g(s,u)]_{i}}.
\] (7.12)
When the inequality above holds, problem (7.10) fulfills Proposition 4.3, whereby problems (7.10) and (7.9) are Fast-Lipschitz. This implies that the optimal solution is given by
\[
\begin{bmatrix}
y^* \\
z^*
\end{bmatrix} = \begin{bmatrix} f_{y^*}(y^*, z^*) \\
0
\end{bmatrix},
\]
i.e., the optimal solution of the original problem (7.7) is given by \( u^{i*} = -z^{i*} = 0 \) for all \( i \). We will now apply Result 7.1 on two concrete examples of the dynamics \( \nabla f(s,u) \), one linear and one non-linear first order system.

**Linear first order system**

Consider for illustrative purposes a linear first order system with a linear cost function. The optimal control problem (7.7) becomes
\[
\min_{\{u^i\}_{i=1}^N} \sum_{i=1}^N g(s_i, u_i) \\
\text{s.t.} \\
s^1 = s_{\text{init}} \\
s^{i+1} = f(s^i, u^i) + w^i, \quad i = 1, \ldots, N - 1, \\
0 \leq u^i \leq u_{\text{max}}, \quad i = 1, \ldots, N,
\] (7.13)

where \( f(s,u) = as - bu \) and \( g(s,u) = c_s s + c_u u, \)

with \( a, b, c_s, c_u > 0 \). In this case, the gradients are given by \( \nabla_s f(s,u) = a, \) \( \nabla_u f(s,u) = -b, \) \( \nabla_s g(s,u) = c_s \) and \( \nabla_u g(s,u) = c_u. \) Since the gradients are scalar and constant, it is straightforward to see that Result 7.1 applies to problem (7.13) provided that \( a < 1 \) and
\[
\frac{b}{1 - a} < \frac{c_u}{c_s}.
\]
If this is the case, the optimal solution must be \( u^{i*} = 0 \) for all \( i \). We remark that this is an important and non obvious result.

**Non-linear first order system**

Consider again problem (7.13), but with the dynamics given by
\[
f(s,u) = \frac{as}{1 + s} - bu.
\] (7.14)

We continue to assume \( a, b > 0 \), and the cost function \( g(s,u) \) is left unchanged from the linear case. The factor \( as/(s + 1) \) can be seen as a variable decay-rate, under which large states decay with a factor close to \( a \) while small states decay almost
Table 7.1: Shared problem parameter values.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$s_{\text{init}}$</th>
<th>${w^i}_{i=1}^N$</th>
<th>$a$</th>
<th>$c_s$</th>
<th>$c_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1</td>
<td>$w^i \sim \text{uni}[0,1]$</td>
<td>0.5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

instantly. Assume that the disturbances $w^i$ are non-negative, and let the upper limit of $u^i$ depend on the current state $s^i$,

$$u \leq u_{\text{max}}(s^i)$$

where $u_{\text{max}}(s) = \frac{as^2}{b(1+s)}$.

Note that Corollary [4.2] still applies when moving from problem (7.9) to problem (7.10) because $u^{i\star} = 0 \leq u_{\text{max}}(s^i)$, so the optimal values $u^{i\star} = 0$ of the relaxed problem (7.10) are always feasible in problem (7.9). The modified constraints $u_{\text{max}}(s^i)$ ensure $f(s^i, u^i) \geq 0$, whereby a non-negative $s_{\text{init}} \geq 0$ implies that all future states $s^i$ are non-negative. All gradients from the linear example are left unchanged, except

$$\nabla_s f(s, u) = a \frac{s^2 + 2s}{s^2 + 2s + 1}.$$ 

This gradient can be bounded by $0 \leq \nabla_s f(s, u) \leq a$, since the modified constraint $u_{\text{max}}(s^i)$ ensures that only non-negative states $s$ can be reached. Therefore, it holds that

$$\max_{s, u} \|\nabla_s f(s, u)\|_\infty \leq a,$$

whereby the conditions

$$|a| < 1 \quad \text{and} \quad \frac{b}{1-a} < \frac{c_u}{c_s} \quad (7.15)$$

remain unchanged from the linear example. When these conditions hold, we know that $u^i = 0$ for all $i$ is the optimal solution. Because of the non-affine equality constraints in (7.14), problem (7.15) is a non-convex problem in $N$ variables subject to $2N$ non-trivial constraints when $u_{\text{max}}$ is a function of $s^i$. However, by using the Fast-Lipschitz properties of the problem we have solved it without performing any calculations, except for those involved in verifying the assumptions.

We now conclude this example by numerically solving two instances of problem (7.13).

**Numerical example**

Consider two different instances of problem (7.13), with the non-linear dynamics given by (7.14). The two problems share all parameter values, given in Table 7.1 except for that of $b$; the first problem instance has $b = 0.3$ and the second instance has $b = 0.5$. Both instances fulfill $0 \leq a < 1$, but only the first instance fulfill

$$\frac{b}{1-a} = \frac{0.3}{1-0.5} = 0.6 < \frac{2}{3} = \frac{c_u}{c_s}.$$
Figure 7.4: The plots show the optimal control variables, and the resulting states and costs. The first system ($b = 0.3$) is marked with a solid blue line and dots. As expected, the optimal control is zero at all times. The second system ($b = 0.5$) is marked by a dashed red line and stars. The total accumulated cost (i.e., the sum over each stage-cost line) is 42.03 and 40.85 respectively. The second system ($b = 0.5$) has a lower total cost and this is expected, since the second system has a more powerful actuator (at the same actuation cost).
7.4. Summary

We can therefore guarantee that \( u^i = 0 \) for all \( i \) solves the first instance. However, we cannot say anything about the second instance because the qualifying conditions are only sufficient. The two problems are solved numerically with Matlab’s built-in solver \texttt{fmincon}, and the results are shown in Figure 7.4. As expected by Result 7.1, the optimal actions for the first system \( (b = 0.3) \) is to always keep the control variables at the lower bound. The second system \( (b = 0.5) \) illustrates a case when Result 7.1 does not apply.

7.4 Summary

This chapter gave three examples of the theory from Chapters 3, 4 and 5. The examples illustrated how to check whether the problems were Fast-Lipschitz, and how Fast-Lipschitz problems are easily solved. In the first two examples the solution was found through fixed point iterations, while the optimal control problem had the solution trivially given by the lower bound of the constraints. This illustrates the potential of Fast-Lipschitz optimization, and motivates further theoretical work to expand the class of problem that fall under this framework.
8.1 Summary

This work has presented several contributions to the theory of Fast-Lipschitz optimization. On the theoretical level, Chapter 3 presented a general qualifying condition that unifies and extends the previously known conditions. This general condition was proved by a new technique, highlighting that Fast-Lipschitz problems always have dual variables that are strictly positive. The general condition was analyzed through a series of special cases that are easier to verify. Furthermore, this work examined variations to the problem structure required by the Fast-Lipschitz framework. In particular, Chapter 4 analyzed cases where the number of variables were more, or less, than the number of constraints, and cases where the cost function was unaffected by a subset of the variables. Finally, this work gives a first theoretical comparison of the convergence speeds of the fixed point iterations available through the Fast-Lipschitz framework, and a traditional Lagrangian descent method.

This work also presented contributions to the area of radio power control in wireless networks. In Chapter 6 problems from the classes of standard and type-II standard algorithms were shown to be Fast-Lipschitz. As a result, the optimality of the fixed points computed by these algorithms were guaranteed, which was not known previously. Furthermore, Fast-Lipschitz optimization was applied to power-control problems that are non-monotonic and therefore neither standard, nor type-II standard. This analysis gave conditions for the problem under which the common fixed point algorithm converges to the optimal point of the problem, which was previously unknown.
8. Conclusions and Future Work

8.2 Future work

In this section, we suggest and discuss a number of future research directions.

Further refinement of the qualifying conditions

When problems are identified as Fast-Lipschitz the process of solving them is greatly simplified, especially in a distributed environment. However, many applications fail to fulfill the qualifying conditions. The fact that the qualifying conditions are sufficient and not necessary hints that further relaxations are possible. A necessary condition would be helpful not only in the process of verifying if a given problem is Fast-Lipschitz, but also as a starting point for development of new qualifying conditions.

Some lines of work that could refine the qualifying conditions are:

- **Do the qualifying conditions need to apply everywhere?**
  Currently for all points in the bounding box \( \mathcal{D} \), although one could relax this to only consider feasible points. However, if this is done, the qualifying conditions only ensure that \( \mathbf{x}^* = f(\mathbf{x}^*) \) is the optimal point — unless checked separately it is no longer guaranteed that the iterations \( \mathbf{x}^{k=1} = f(\mathbf{x}^k) \) converge to \( \mathbf{x}^* \), or that \( \mathbf{x}^* \) is the unique fixed point. A further possibility might be to only require that the qualifying conditions are fulfilled on the boundary of the feasible region, i.e., where at least one of the constraints are active.

- **Significance of equality constraints**
  Current conditions treat all constraints as inequalities, thereby losing information from the equality constraints that could be used to relax the conditions.

- **Transformation of the variables and constraints**
  The way a Fast-Lipschitz problem is written is not unique, in the sense that there always exists equivalent problems with the same solution. An example of this can be seen in Section 7.2 where \( \beta \) in equation (7.5) can be chosen arbitrarily. There is an arbitrary choice in the scaling of the constraints because \( x_i \leq f_i(\mathbf{x}) \) is equivalent to

\[
x_i \leq \frac{f_i(\mathbf{x}) - \gamma_i x_i}{1 - \gamma_i} \triangleq g_i(\mathbf{x}).
\]

By varying \( \gamma_i \) it is possible to find situations where the transformed problem (with \( x_i \leq g_i \)) fulfills the qualifying conditions although the original problem (with \( x_i \leq f_i \)) does not. However, finding the optimal scalars \( \gamma_i \) for all \( i \) is not straightforward. Similarly, one can make a change of variables and instead study the problem in terms of \( \mathbf{y} = \mathbf{Sx} \) for some invertible matrix \( \mathbf{S} \). An example of this technique can be seen in proof of Proposition 3.12.
8.2. Future work

Since the properties that make a problem Fast-Lipschitz should be invariant of the simple transformations above, so should the qualifying conditions. This is currently not the case, and finding conditions that are invariant under these (and possibly other) transformations should be a part of future research efforts.

- **Forming the qualifying conditions with more general cones**
  So far, only problems with \( \nabla f_0(x) \geq 0 \) have been considered. The standard inequality we have used is a partial ordering induced by the non-negative orthant \( \mathbb{R}_+^m \), i.e., \( \nabla f_0(x) \succeq_{\mathbb{R}_+^m} 0 \). A possible extension is to allow for problems where \( \nabla f_0(x) \succeq_{\mathcal{K}} 0 \) for a more general cone \( \mathcal{K} \). This would allow a tradeoff between conditions for \( \nabla f_0(x) \) and conditions on \( \nabla f(x) \), which may give more flexibility in the qualifying conditions.

  We have already seen a similar example of this, when case (ii) of the old conditions is generalized to \( Q_2 \). In this case, condition \((2.8d)\) required \( f_0(x) = c_1^T x \) for some \( c_1 > 0 \), which can also be stated

\[
f_0(x) \in \mathbb{R}, \quad \text{and} \quad \nabla f_0(x) \succeq_{\mathcal{K}_1} 0
\]

where \( \mathcal{K}_1 \) is the cone (ray) generated by the single vector \( 1 \). In \( Q_2 \), it is instead required that \( \nabla f_0(x) > 0 \), which is more general. This, however, comes at the price of a less general requirement on \( f(x) \), since \((Q_2,c)\) requires

\[
\|\nabla f(x)\|_\infty < q(x) \leq 1,
\]

while \((2.8f)\) only requires \( \|\nabla f(x)\|_\infty < 1 \).

- **Fast-Lipschitz optimization for non-smooth problems**
  The proof (in Section 3.3) of the general qualifying condition of this thesis makes repeated use of the gradients \( \nabla f_0(x) \) and \( \nabla f(x) \). We believe that the Fast-Lipschitz optimization framework can be extended to cover also non-smooth problems. A potential benefit of such an extension would be to form expressions such as \( x \leq f(x) = \min\{f_1(x), f_2(x)\} \), whereby problems with more constraints than variables \((f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^p \text{ where } p > n)\) could be considered.

**Convergence speed, computational effort and amount of communication**

This thesis gave a theoretical convergence speed comparison, although for the special case of equality constrained problems. This analysis only compares the number of iterations of the two algorithms and does not take into account the computational effort of each iteration. Furthermore, applications where energy is scarce, such as battery powered wireless sensor networks, require also the amount of communication to be kept small. In such a case the relevant comparison would be the number of radio transmissions required to calculate the optimal solution. Clearly, this will
depend on the structure of the network, and good theoretical results may be hard to obtain. In these cases it would be interesting to get insight from numerical comparisons through simulations.

Privacy preserving Fast-Lipschitz optimization

As a consequence of rapid growth of available data, and advancements of methods for extracting information from this data, the issue of privacy has become a hot research topic. The challenge is to construct distributed methods that allow agents to collaborate without having to reveal private sensitive data. Most available methods rely on cryptographical methods that usually scale poorly in problem size. However, the recent paper [62] shows that many of the existing methods for distributed optimization protect private data by construction, or may do so after a simple transformation. The paper also introduces a metric to quantify and compare the privacy of different methods.

An interesting research direction is to characterize the inherent privacy preserving properties of Fast-Lipschitz optimization, if any. Since the fixed point method in general required node $i$ to reveal the variable $x_i$ in each iteration, it may be challenging to provide privacy for the variables $x_i$. However, there might still be sensitive information in the structure or parameters of the constraint function $f_i$. Since the fixed point method is iterative, an eavesdropper could learn the value of $f_i(x^k)$ for several time instants $k$, and eventually try to reconstruct $f_i$. The analysis of this situation is closely related to the convergence speed of the iterations, since a faster convergence would give the eavesdropper less samples to use in the reconstruction. The analysis is further complicated since the algorithm also converges asynchronously, whereby node $i$ can complicate the task of the eavesdropper by intentionally updating with outdated information.
Appendix A

Vector optimization and Pareto optimality

In this thesis we are concerned with maximization of vector-valued objective functions \( f_0(x) \in \mathbb{R}^m \). A vector optimization problem involves a proper cone \( \mathcal{K} \) (see, e.g., [63, Section 4.7]). We focus on the case when \( \mathcal{K} \) is non-negative orthant
\[
\mathcal{K} = \mathbb{R}_+^m \triangleq \{ y : y_i \geq 0, \forall i = 1, \ldots, m \}.
\]

This maximization of a vector is formally expressed as
\[
\text{maximize (with respect to } \mathbb{R}_+^m) \ f_0(x). \tag{A.1}
\]

The goal of the maximization (A.1) is to find the decision vector \( x \) such that the components of \( f_0(x) \) are as big as possible with respect to the cone \( \mathbb{R}_+^m \). In particular, when comparing two vectors \( x \) and \( y \) with respect to the cone \( \mathbb{R}_+^m \), we say \( x \succeq_{\mathbb{R}_+^m} y \) if \( x - y \in \mathbb{R}_+^m \) and \( x \preceq_{\mathbb{R}_+^m} y \) if \( y - x \in \mathbb{R}_+^m \). Note that this corresponds exactly to the component-wise inequalities \( y \geq x \) and \( y \leq x \), and for this reason we will use \( \geq \) rather than \( \succeq \) throughout the thesis. Unlike scalars, where it must hold that either \( a \geq b \), or \( a < b \), two vectors might not be comparable. For example, for \( x = [1 \ 2]^T \) and \( y = [3 \ 1]^T \), we have \( x \npreceq y \) and \( y \npreceq x \). Based on the discussion above, \( x_1 \) is preferable to \( x_2 \) in problem \( (A.1) \) if \( f_0(x_1) \geq f_0(x_2) \).

**Definition A.1.** A feasible decision variable \( \hat{x} \) is said to be *Pareto optimal* for problem \( (A.1) \) if there is no other feasible vector \( x \) such that \( f_0(x) \succeq f_0(\hat{x}) \).

A problem can have several Pareto optimal points – in this case each Pareto optimal point is incomparable to any other Pareto optimal points, but preferable.

---

\( ^1 \) In this thesis we only consider optimization with respect to the cone \( \mathbb{R}_+^m \). Therefore, for notational simplicity we simply write max \( f_0(x) \), and the vector optimization \( (A.1) \) should be understood whenever \( f_0(x) \) is vector-valued.
A. Vector optimization and Pareto optimality

to any point that is not Pareto optimal. If a problem only has one Pareto optimal point \( \mathbf{x}^* \), then \( \mathbf{x}^* \) is unambiguously the best choice among the decision vectors.

Scalarization is a useful technique for generating Pareto optimal points. Scalarization is performed by picking a weight vector \( \mathbf{\mu} \) in the interior of \( \mathbb{R}_+^m \), i.e., \( \mathbf{\mu} > 0 \), and solving the scalar optimization problem

\[
\max_{\mathbf{x} \text{ feasible}} \ \mathbf{\mu}^T \mathbf{f}_0(\mathbf{x}) = \sum_{i=1}^m \mu_i [\mathbf{f}_0(\mathbf{x})]_i. \tag{A.2}
\]

Any point \( \hat{\mathbf{x}} \) that is optimal in problem (A.2) is Pareto optimal in problem (A.1), but the converse is not true \( \text{[63]} \). In other words, each choice of the scalarization vector \( \mathbf{\mu} > 0 \) in problem (A.2) gives a Pareto optimal point for problem (A.1), but there may be Pareto optimal points of (A.1) that do not correspond to a \( \mathbf{\mu} > 0 \) in problem (A.2).

We end this section with a case where the converse is true, i.e., when all Pareto optimal points can be found through scalarization.

**Lemma A.2.** If a point \( \mathbf{x}^* \) is optimal in problem (A.2) for all scalarization vectors \( \mathbf{\mu} > 0 \), i.e., if \( \mathbf{x}^* \) is independent of the choice of \( \mathbf{\mu} \), then \( \mathbf{x}^* \) is the unique Pareto optimal point in problem (A.1).

**Proof.** If \( \hat{\mathbf{x}} \) is not the unique Pareto optimal point, then there exists a point \( \tilde{\mathbf{x}} \) such that \([\mathbf{f}_0(\tilde{\mathbf{x}})]_j < [\mathbf{f}_0(\hat{\mathbf{x}})]_j\) for at least one \( j \). Define

\[
\Delta = \max_j [\mathbf{f}_0(\tilde{\mathbf{x}})]_j - [\mathbf{f}_0(\hat{\mathbf{x}})]_j
\]

and

\[
\delta = \min_j [\mathbf{f}_0(\tilde{\mathbf{x}})]_j - [\mathbf{f}_0(\hat{\mathbf{x}})]_j.
\]

The optimal values \( \Delta \) and \( \delta \) are sure to exist if \( \mathbf{f}_0(\mathbf{x}) \) is bounded. Pick an index \( k \) such that \( \Delta = [\mathbf{f}_0(\tilde{\mathbf{x}})]_k - [\mathbf{f}_0(\hat{\mathbf{x}})]_k \), and a constant \( C > m\delta/\Delta \). Then, there exists a

\[
\tilde{\mathbf{\mu}} = 1 + C \mathbf{e}_k,
\]

where \( \mathbf{e}_k \) is the \( k \)th unit vector, such that

\[
\tilde{\mathbf{\mu}}^T (\mathbf{f}_0(\tilde{\mathbf{x}}) - \mathbf{f}_0(\hat{\mathbf{x}})) = \sum_{i=1}^m \left( [\mathbf{f}_0(\tilde{\mathbf{x}})]_i - [\mathbf{f}_0(\hat{\mathbf{x}})]_i \right) + C \left( [\mathbf{f}_0(\tilde{\mathbf{x}})]_k - [\mathbf{f}_0(\hat{\mathbf{x}})]_k \right) \\
\geq m\delta + C\Delta > 0.
\]

This gives

\[
\mathbf{\mu}^T \mathbf{f}_0(\hat{\mathbf{x}}) > \mathbf{\mu}^T \mathbf{f}_0(\mathbf{x}),
\]

wherefore \( \hat{\mathbf{x}} \) is not optimal for all scalarization vectors. \( \square \)
Lemma B.1. Let $\|\cdot\|_a$ be a matrix norm. Then $\|\cdot\|_b$, defined as $\|A\|_b = \|A^T\|_a$ is also a matrix norm.

Proof. A function $\|\cdot\|$ is a matrix norm if it fulfills

1. $\|A\| \geq 0$
2. $\|A\| = 0 \iff A = 0$
3. $\|cA\| = c \|A\|$ for all scalars $c$
4. $\|A + B\| \leq \|A\| + \|B\|$
5. $\|AB\| \leq \|A\| \|B\|$

The norm $\|A\|_b = \|A^T\|_a$ fulfills all the matrix norm properties from above, since

1. $\|A\|_b = \|A^T\|_a \geq 0$
2. $\|A\|_b = \|A^T\|_a = 0 \iff A^T = 0 \iff A = 0$
3. $\|cA\|_b = \|cA^T\|_a = c \|A^T\|_a = c \|A\|_b$ for all scalars $c$
4. $\|A + B\|_b = \|A^T + B^T\|_a \leq \|A^T\|_a + \|B^T\|_a = \|A\|_b + \|B\|_b$
5. $\|AB\|_b = \|B^T A^T\|_a \leq \|B^T\|_a \|A^T\|_a = \|A\|_b \|B\|_b$

Each of the chains above uses only the definition of $\|\cdot\|_b$, and the corresponding property that holds for the matrix norm $\|\cdot\|_a$ (marked by $*$).
Bibliography


