Accelerating Convergence of Large-scale Optimization Algorithms

EUHANNA GHADIMI

Doctoral Thesis
KTH Royal Institute of Technology
School of Electrical Engineering
Department of Automatic Control
SE-100 44 Stockholm, SWEDEN

TRITA-EE 2015:012
ISSN 1653-5146
ISBN 978-91-7595-485-1

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i reglerteknik onsdag den 29 april 2015, klockan 10.00 i sal F3, Kungliga Tekniska högskolan, Lindstedtsvägen 26, Stockholm.

Abstract

Several recent engineering applications in multi-agent systems, communication networks, and machine learning deal with decision problems that can be formulated as optimization problems. For many of these problems, new constraints limit the usefulness of traditional optimization algorithms. In some cases, the problem size is much larger than what can be conveniently dealt with using standard solvers. In other cases, the problems have to be solved in a distributed manner by several decision-makers with limited computational and communication resources. By exploiting problem structure, however, it is possible to design computationally efficient algorithms that satisfy the implementation requirements of these emerging applications.

In this thesis, we study a variety of techniques for improving the convergence times of optimization algorithms for large-scale systems. In the first part of the thesis, we focus on multi-step first-order methods. These methods add memory to the classical gradient method and account for past iterates when computing the next one. The result is a computationally lightweight acceleration technique that can yield significant improvements over gradient descent. In particular, we focus on the Heavy-ball method introduced by Polyak. Previous studies have quantified the performance improvements over the gradient through a local convergence analysis of twice continuously differentiable objective functions. However, the convergence properties of the method on more general convex cost functions has not been known. The first contribution of this thesis is a global convergence analysis of the Heavy-ball method for a variety of convex problems whose objective functions are strongly convex and have Lipschitz continuous gradient. The second contribution is to tailor the Heavy-ball method to network optimization problems. In such problems, a collection of decision-makers collaborate to find the decision vector that minimizes the total system cost. We derive the optimal step-sizes for the Heavy-ball method in this scenario, and show how the optimal convergence times depend on the individual cost functions and the structure of the underlying interaction graph. We present three engineering applications where our algorithm significantly outperform the tailor-made state-of-the-art algorithms.

In the second part of the thesis, we consider the Alternating Direction Method of Multipliers (ADMM), an alternative powerful method for solving structured optimization problems. The method has recently attracted a large interest from several engineering communities. Despite its popularity, its optimal parameters have been unknown. The third contribution of this thesis is to derive optimal parameters for the ADMM algorithm when applied to quadratic programming problems. Our derivations quantify how the Hessian of the cost functions and constraint matrices affect the convergence times. By exploiting this information, we develop a preconditioning technique that allows to accelerate the performance even further. Numerical studies of model-predictive control problems illustrate significant performance benefits of a well-tuned ADMM algorithm. The fourth and final contribution of the thesis is to extend our results on optimal scaling and parameter tuning of the ADMM method to a distributed setting. We derive optimal algorithm parameters and suggest heuristic methods that can be executed by individual agents using local information. The resulting algorithm is applied to distributed averaging problem and shown to yield substantial performance improvements over the state-of-the-art algorithms.

Sammanfattning

Många nya tillämpningar inom områden som multiagentsystem, reglerteknik, kommunikationsteori och maskininlärning innefattar beslut som ska fattas på bästa möjliga sätt. Matematiska kan detta formuleras som optimeringsproblem. I vissa fall är de resulterande problemen mycket stora med många beslutsvariabler. I andra fall måste problemen lösas distribuerat av flera olika beslutsfattare som var och en har begränsade beräkningsresurser. Det visar sig ofta att de traditionella och generella optimeringslösarna är olämpliga för dessa nya problem. Genom att utnyttja de givna problemstrukturerna kan man istället formulera beräkningsmässigt mycket mer effektiva algoritmer för de specifika optimeringsproblemen.

I denna avhandling studeras ett antal olika tekniker för att förbättra prestandan hos optimeringsalgoritmer för storskaliga problem. Först studeras heavy-ball-metoden som en beräkningstekniskt enkel teknik för att öka konvergenshastigheten hos gradientmetoden. Heavy-ball-metoden introducerar minne i gradientmetoden genom att ta tidigare iterationer i beaktande när nästa iterat beräknas. Det har visats att heavy-ball-metoden har betydande fördelar jämfört med gradientmetoden i fråga om lokal konvergens för två gånger kontinuerligt deriverbara målfunktioner. Metodens globala konvergensegenskaper har dock varit okända under lång tid. Här presenteras en global konvergensanalys för heavy-ball-metoden applicerad på problem med Lipschitzkontinuerliga gradienter och starkt konvexa kostnadsfunktioner. Vidare introduceras en familj av gradient-baserade flerstegsmetoder för nätverksoptimeringsproblem. Algoritmerna bygger på att problemet distribueras till ett antal beslutsfattare som var för sig utför en typ av heavy-ball-iterationer. Algoritmernas prestanda kan ytterligare förbättras genom rätt val av parametrar. Tre tillämpningar där de nya algoritmerna uppvisar betydliga prestandaförbättringar jämfört med gradientmetoden presenteras i denna avhandling.

Slutligen studeras ett tredje alternativ för att lösa storskaliga optimeringsproblem med viss given struktur. Metoden Alternating Direction Method of Multipliers (ADMM) är en teknik som ökat i popularitet inom många olika ingenjörsområden. Prestandan hos ADMM beror kritiskt på valet av ett antal parametrar. Det bästa valet för ett givet problem har hittills varit okänt. I denna avhandling studeras valet av optimala parametrar för ADMM då den används för att lösa centraliserade och distribuerade kvadratiska optimeringsproblem. För centraliserade problem spelar hessianens och bivillkorsmatrisernas spektralgap en avgörande roll medan kommunikationsgrafens spektralegenskaper är avgörande för distribuerade problem. Följaktligen kan prestandan hos ADMM förbättras genom skalning av ursprungsproblemet. Numeriska exempel visar fördelarna hos en optimalt skalad och inställd ADMM-algoritm jämfört med andra tillgängliga metoder.

To my mother, Narges, and my father, Khalegh.

Acknowledgements

I am grateful to my main advisor Mikael Johansson. He provided the opportunity for me to pursue a PhD and walked me through the way with a remarkable patience. I also want to thank my co-advisor Carlo Fischione for the nice discussions and providing the opportunity to teach Wireless Sensor Network course with him.

I am indebted to several cheerful friends and brilliant collaborators during my study in KTH. Dear Iman, Pablo, Olaf, André, PG, Jose, Burak, and Hamid thank you all. I really enjoyed and learned a lot from working with you.

I want to thank all the past and present colleagues at the Automatic Control department for the fun times, great discussions, and hang outs all over the world. Specially, Mikael's group people: Arda, Burak, Antonio G., Demia, Sadegh, Hamid, Themis, Jie, Liqun, Zhenhua, and Jeff, thank you for all the vibrant technical meetings and more importantly for all the hilarious fun activities that we did in past 5 years. You are awesome!

Special thanks goes to Anneli, Hanna, Karin, Kristina, and Gerd for taking care of all the things that make the department run so smoothly.

I want to thank Christian L. for his kind helps with writing the Swedish summary of thesis and for providing latex templates, and Afrooz, Antonio A., Hamid, Sadegh, Sindri, Sebastian, Themis, and Zhenhua for proof reading the thesis.

I want to thank the Swedish Research Council (VR) and The Swedish Foundation for Strategic Research (SSF), for the financial support of this work.

I want to express my sincere gratitude to all my great friends in Stockholm Burak, Esther, Alireza, PG, Alessandra, Pan, Damiano, Pablo, Annemarie, Chithrupa, André, Kathia, Salome, Edurne, Ane, Pili, Rafa, Farhad, Jalil, Assad, Behdad, Afrooz, Iman, Hammed, Maryam, Sadegh, Zahra, Kaveh, Keivan, Gabriel, Marco, Martin, Hossein, and Forough, for providing meaning to life outside research.

I would like to extend my deepest gratitude to my parents: Maman Narges, and Baba khalegh, lovely sisters: Elham and Samaneh, parents-in-law: Maman Akram and Baba Mansour, brothers and sister-in-law: Kamran, Ehsan, and Somayeh, for always believing in me, for all the support and for their unconditional love. My aunt Khale Saeideh and her beloved family: Agha Jafari, Ali, Sepideh, Hassan, Amirhossein, Sheida, Agha Raziei, and Parsa deserve special thanks for their kindness and limitless helps during my 7 years study in Tehran. Last, but not least, my utmost appreciation goes out to my amazing wife Elaheh, without your love, support, and encouragement I could not have done it!

Euhanna Ghadimi
Stockholm, March 2015.

Contents

Chapter 1
Introduction

IN the age of connectivity, billions of cell-phones[1], tablets, cars, smart appliances, wireless sensors, and other devices are beginning to form an “intelligent ambient”. In the mid 80’s, when the Internet was first introduced, it would have been hard to anticipate that less than 30 years later, networked devices would play such a prominent part in our everyday lives.

Still, it is likely that we have only seen the beginning of this networked society. In emerging road transportation networks, it is envisioned that groups of autonomous vehicles interact with each other and operator management centers. By accessing critical traffic information from the infrastructure, vehicles will be able to compute efficient routes, and even form platoons to minimize fuel consumption while ensuring safety and traffic constraints. As another example, in the near future, electric cars, smart household appliances, and new power meters will cooperate in large-scale “smart grids” to help customers to control their power bills and emissions, while allowing power system operators to safely and efficiently integrate large amount of renewable energy production.

One thing that the above examples have in common is that peers with limited communication and computation capabilities have to act collectively to perform a complex task. In other words, it is the clever interactions among the interconnected peers that make the overall system appear intelligent.

One way of engineering these interconnected systems is to develop them by engineering intuition in a trial-and-error fashion. This approach has been used frequently in the past, and produced several impressive systems in computer networking, wireless sensor networks [1], multi-agent systems [2], etc. However, it is likely that it has produced many more failed attempts, where system interactions have proven too complex to manage in an ad-hoc manner.

To be able to exploit the full potential of modern networked systems, we need systematic techniques for designing mechanisms that coordinate connected peers. Ideally, these should ensure that the peers converge quickly to the optimal operating point and do so in an energy-efficient manner with a minimal information exchange. In several emerging applications, it

[1]There are almost as many cell-phone subscriptions (6.8 billion) as there are people on this earth (seven billion) and it took a little more than 20 years for that to happen. In 2013, there were some 96 cell-phone service subscriptions for every 100 people in the world. source: International Telecommunication Union (ITU), the United Nations specialized agency for information and communication technologies.

is also desirable to have formal guarantees that the final implementation behaves correctly and safely, and that the system respects end-user privacy. We argue that such formal guarantees can only be given if we base our design based on systematic and scientifically sound techniques.

There exists a great deal of interest in developing novel mathematical and computational tools for fundamental understanding and engineering design of interactions between the connected peers in emerging networks. This thesis is part of these exertions and aims to contribute by designing optimization techniques for advanced engineering applications.

1.1 Engineering interconnected systems by optimization

Optimization theory provides an attractive framework for solving numerous decision problems. It provides a methodology to formalize the objective of an engineering problem and the operational constraints in mathematical terms and then look for the best solution. Using mathematical notation, an optimization problem can be formulated as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X}. \end{array} \tag{1.1}$$

Here, the vector $x \in \mathbb{R}^n$ is the optimization variable (representing the decision parameter that we optimize over), the function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is the objective function (describing the loss, or cost of operating our system at x), and $\mathcal{X}$ is the set of constraints that our decision vector should satisfy.

Once we formulate the optimization problem, then obviously we would be interested in solving it. It usually can done by an iterative process called optimization algorithm.

Classical optimization algorithms typically run in a central computer where the objective function and the constraints set are known and described by closed-form expressions. Distributed optimization algorithms, on the other hand, decompose the optimization problem into multiple pieces assigned to disjoint processors or agents that collaboratively solve the overall problem. Given the limited capabilities of the individual agents, simple computation and collaboration mechanisms are often required for agents to carry out the local computations and interact with neighbors. One example of a distributed optimization problem is illustrated in Figure1.1.

Alternative theoretical frameworks such as control and game theory are also suitable means to deal with decentralized decision making problems. In control theory, one studies the behavior of dynamical systems with inputs and outputs and how to modify their behavior by feedback. The objective in a control problem is typically to keep the state of the system at rest despite uncertainties, or to shape the dynamic response of a system, despite uncertainties.

Game theory, on the other hand, is about reconciling different interests in a competitive environment. The majority of game theory considers non-collaborative decision-making and addresses how individual strategies can influence the state of a competition which is often called a game.

In emerging engineering applications, we believe that optimization, control, and game theory should go hand in hand in order to address different aspects of decision making.

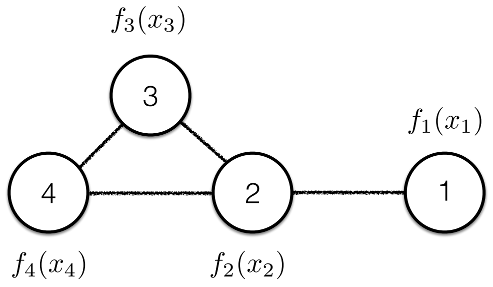


Figure 1.1: An example of a distributed optimization problem. A network of 4 agents collaboratively solve an optimization problem of form $f(x) = \text{minimize} \sum_i f_i(x)$. Each node i is endowed with a local cost function f_i and a local variable x_i. An example of f_i and x_i are a function penalizing a mobile agent for deviating from its original position and the current position of the agent, respectively. The agents do not have access to each other cost functions and only can communicate to a subset of entire agents. A line connecting two agents indicates that they can communicate with each other. The constraints $x_i = x_j$ for i, j being connected by a line indicates that the neighboring agents should meet each other at a common position.

In the current thesis, we focus on optimization theory to formulate and solve collaborative engineering problems.

1.2 Convex optimization

This thesis is about convex optimization, an important subset of mathematical optimization. A convex optimization problem has an objective function that satisfies

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \tag{1.2}$$

for all $x, y \in \mathbb{R}^n$ and $\theta \in [0, 1]$. In addition, in a convex optimization problem, the constraint set is convex. That is for all $x, y \in \mathcal{X}$,

$$\theta x + (1-\theta)y \in \mathcal{X}, \tag{1.3}$$

for any $\theta \in [0, 1]$. Figure1.2 depicts a convex function and a convex set.

Convex optimization problems have several distinct advantages. First, every locally optimal point is also globally optimal. When we have found a locally optimal point, we can safely terminate our algorithm knowing that we have found the optimal solution. Convex optimization problems also have a strong and useful duality theory. Associated to every (primal) optimization problem is another dual problem. For convex problems, under mild assumptions of constraint qualification [4], the optimal value of the primal and dual problem agree. This is known as strong duality. Moreover, when strong duality holds, the Karush-Kuhn-Tucker conditions provide a necessary and sufficient characterization of primal-dual

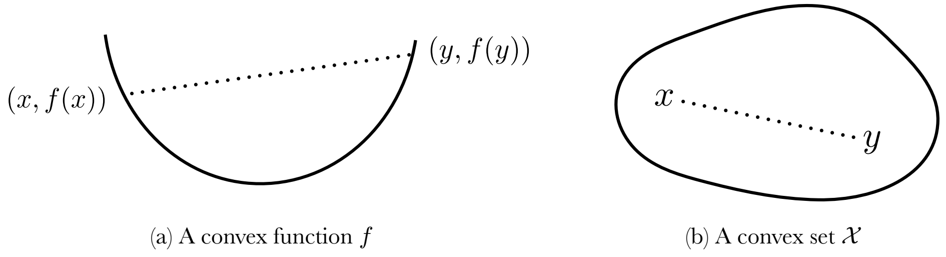


Figure 1.2: Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function defined on $\mathbb{R}^n$. Then we say f is convex if for any two points $x, y \in \mathbb{R}^n$, the line segment between $(x, f(x))$ and $(y, f(y))$ lies above f. We say a set $\mathcal{X}$ is convex if for any two points $x, y \in \mathcal{X}$, the line segment connecting x and y lies in $\mathcal{X}$.

optimal points. These conditions can be used to develop well-founded stopping criteria for iterative algorithms, to bound how far from optimal a given iterate is, and to design efficient algorithms that exploit the structure in both primal and dual problems.

By exploiting properties of convex problems, several efficient solution techniques have been developed in recent decades. One group of such techniques are the interior-point methods [3] that solve a wide range of convex problems including linear programs and quadratic programs to a specified accuracy within a number of operations that does not exceed a polynomial function of the problem dimension.

A large number of engineering problems can be formulated as convex problems and many others can be well approximated by convex problems (see e.g., [4, 5] for detailed discussions on convex optimization methods and convexifying techniques).

1.2.1 Engineering applications that use convex optimization

Given the benefits of convex problems, several engineering communities have recently applied convex optimization techniques to solve their problems of interest:

Multi-agent systems involve a collection of mobile agents equipped with processing and communication units performing collective tasks. Convex optimization theory provides an attractive framework to formulate many problems including distributed estimation and control for robotic networks [6], formation control and coordination of autonomous agents [7], and decentralized rendezvous problems in multi-agents [8].

Wireless sensor networks consist of small sensor nodes with limited sensing, processing, and communication capabilities that are usually deployed in some fields of interest to perform monitoring, detection, or surveillance tasks. Modern wireless sensor network scenarios in which convex optimization techniques are brought into action include: sensor node position estimation [9], designing protocols for reliable packet transfer for industrial process control [10] [11], and deadline-constrained reliable forwarding [12].

Communication networks have been actively developed during past decades. Convex optimization has served as an important tool for researchers in this field. Some examples include: distributed cross-layer congestion control in data networks [13, 14, 15], resource allocation in wireless networks [16, 17], coordinated transmission and power management for wireless interference networks [18, 19, 20], energy-efficient mobile radio-access technologies [21], and quality of service and fairness in cellular networks [22].

Networked control systems is an attractive area that has emerged as recent advances in communication technologies embraced the traditional control techniques [23, 24]. Convex optimization methods have contributed a major role in this topic including distributed model predictive control [25, 26], fuel-efficient heavy-duty vehicle platooning [27], distributed reconfiguration for sensor and actuators [28], and cyber-security and resilience against faults and attacks [29].

Machine learning deals with designing algorithms that can learn from data. Such algorithms operate by building a model based on a restricted set of available data and then utilizing the model to perform decision and prediction tasks. Convex optimization has played a major role in developing modern machine learning algorithms. The classical examples include the convex optimization methods applied in: support vector machine [30], image denoising [31], matrix completion [32], and compressed sensing [33] problems. Moreover, recent advances in first-order convex optimization methods has crafted lots of powerful machine learning related techniques such as composite methods [34, 35, 36], incremental gradient methods [37], dual averaging method [38, 39], to name a few.

1.2.2 Convex optimization methods

The gradient descent method is among the earliest methods for solving optimization problems. To find a minimum of a function using gradient descent, one takes steps in the direction of the negative gradient of the function at the current point. In each iteration $k \in \mathbb{N}_0$ the gradient descent method updates its iterate through

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \nabla f(x^{(k)}),$$

where $x^{(k)} \in \mathbb{R}^n$ is the current point, $\alpha^{(k)} \in \mathbb{R}_{++}$ is a positive step-size, and $\nabla f(x^{(k)})$ is the gradient of the function at the current point. The main drawback of the gradient descent algorithm is that for general convex functions it converges slowly toward the optimal solution. More effective optimization algorithms have been invented that require higher order information. One example of such methods is the Newton's method that needs Hessian of the cost function in addition to its gradient at the current point in order to compute the next iteration point. In cases in which higher order functional information is available or can be easily evaluated, the Newton's method can converge significantly faster than the gradient descent method [4].

In the past decades, versatile convex optimization solvers such as interior-point methods have been developed to solve convex optimization problems in polynomial time [40, 3]. For some variations of convex problems, including linear programs, these solvers can handle problem instances with thousands of constraints and variables in a few seconds. In generic nonlinear convex programs, however, such methods can be computationally prohibitive for large sizes of problem data.

1.3 New solution methods for modern engineering applications

In this section, we consider problems that traditional convex solvers are not able to cope with. In particular, we consider two types of engineering applications that motivate the content of this thesis. The first example is the so called distributed or network optimization problems. As discussed earlier, many problems in multi-agent and networking applications involve groups of decision makers with limited resources that collaboratively perform an overall task. Any solution method for these applications should allow for distributed implementation. Moreover, it should impose low computational overheads for individual decision makers.

In a second type of modern applications, one has to deal with a gigantic chunk of data to perform statistical analysis or data mining tasks formulated as optimization problems. Examples of such huge-scale applications includes weather prediction, finite element methods, and the analysis of data extracted from Internet or telecommunication networks. In these problems, one often deals with terabytes of data that needs to be processed. Even loading the entire data into memory, in such applications, can be an issue[2]. An optimization solver for such applications has to involve easy to perform operations due to the huge size of problems in hand.

Motivated by these examples, we investigate new accelerated solution methods that take into account two design principles of modern optimization algorithms: (i) simplicity in a sense that they should be applicable for the large scale problems with lots of parameters; (ii) decomposability in a sense that it should be possible to solve the optimization problem based on a "divide and conquer" paradigm. The problem is split into several pieces and each piece is often assigned to different parties that collaboratively solve the global problem.

In this thesis, we study the efficiency of convex optimization methods for modern applications. The efficiency of an optimization algorithm is characterized by its convergence time, that is, the time it takes to reach to the solution of optimization problem of interest. By optimizing their tunable parameters such as constant step-sizes and constraints scales, one can improve the convergence time of optimization algorithms. In particular, we are interested in the following two solution techniques that efficiently solve nowadays engineering problems.

[2]Since the 1980s, the world's technological per-capita capacity to compute and store information has roughly doubled every 24 and 40 months, respectively [41]. This is, however, far behind the rate in which we generate data. As of 2012, every day 2.5 exabytes (2.5×10^{18}) of data were created; so much that 90% of the data in the world at 2012 had been created in the previous two years alone [42]. About 75% of data is coming from sources such as text, voice and video. And as mobile phone penetration is forecast to grow from about 61% of the global population in 2013 to nearly 70% by 2017, those numbers can only grow [43].

1.3.1 Making the most of first-order methods

Many truly large-scale convex optimization problems can be handled by decomposition techniques that exploit the problem structure in the primal or dual space to distribute the computations on multiple processors. The decomposition techniques are particularly attractive when one can isolate subproblems that are easy to solve, and when these can be effectively coordinated using a simple algorithm such as the gradient method. However, in many cases, it is the slow convergence of the gradient method that constitutes the bottleneck in decomposition methods. By developing computationally cheap techniques that accelerate the convergence of the gradient method, it is possible to speed up decomposition techniques and deal with even larger problem sizes.

One of the simplest way to accelerate the gradient method is to consider multi-step first-order methods. The idea is to design algorithms that generate new iterates as a linear combination of past iterates and past gradient evaluations. Different multi-step methods use a different number of past iterates and past gradients, and weight them together in different ways. It turns out that it is possible to invent accelerated methods that only take just a few past iterates into account when computing the next ones. These methods are particularly efficient since their memory requirement is comparable with the vanilla gradient method but bring a huge performance improvement often in the order of magnitudes. In Chapters 3 and 4 we study accelerated gradient methods and provide theoretical performance bounds for some of these algorithms as well as machinery to implement such methods in distributed optimization.

1.3.2 Adding robustness to the picture

A disadvantage of a gradient based method is that its stability is sensitive to the choice of the algorithm parameters, even to the point where poor parameters can lead to algorithm divergence [44].

The Alternating Direction Method of Multipliers (ADMM) is a powerful algorithm for solving structured convex optimization problems that rectifies this issue. A key feature of the ADMM algorithm is that it converges for all values of algorithm parameters. Moreover, it provides a structured way of decomposing very large problems into smaller sub-problems that can be solved efficiently.

The origins of ADMM can be traced back to the alternating direction implicit (ADI) techniques for solving elliptic and parabolic partial difference equations. In the 70's, see [45] and references therein, ADMM was first introduced for solving optimization problems and enjoyed much attention in the following years. However, the main advantage of applying ADMM in solving distributed optimization problems remained largely untapped. Nevertheless, the technique has again raised to prominence in the last few years[3] as there are many applications, e.g., in financial or biological data analysis, that are too large to be handled by generic optimization solvers.

[3]A search for "Alternating direction method of multipliers" as of January 2015 in google scholar returned about 3000 hits.

Despite the superior stability of the ADMM method, its convergence speed is sensitive to the choice of algorithm parameters. In Chapters 5 and 6 we provide better understanding of the convergence properties of the ADMM method and develop optimal parameter selection rules for a number of problem classes.

1.4 Outline and contributions

This section provides a brief outline of the thesis contributions and lists the publications that the thesis is built upon. A more thorough description and the related work are presented in each chapter.

1.4.1 Chapter 2

In this chapter, the fundamental definitions and algorithms used in the thesis are presented. In particular, we discuss basic notions for fixed point iterations, convex optimization, graph theory, and distributed optimization.

1.4.2 Chapter 3

In this chapter, we present the performance analysis of accelerated first-order methods. The acceleration is obtained by adding extra memory taps to the basic gradient iterates resulting in so called multi-step methods. In particular, we present the global convergence of the celebrated Heavy-ball method for two classes of continuously differentiable convex cost functions. Two variations of the Heavy-ball method with constant and time-varying step-sizes and their convergence rate analysis is presneted in this chapter. As an artifact, we also discuss the convergence rate of the Nesterov method with constant step-sizes for the class of convex cost functions with Lipschitz continuous gradients. In all of these scenarios, we derive sufficient parameters bounds to globally stabilize the corresponding iterates. Numerical examples illustrate our contributions. The chapter is partially based on the following publication.

E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson. Global convergence of the Heavy-ball method for convex optimization. *Mathematical Programming*. 2014. Submitted.

A preliminary version of this work was presented in:

E. Ghadimi, H. R. Feyzmahdavian, M. Johansson. Global convergence of the heavy-ball method for convex optimization. *To appear in European Control Conference*. 2015.

1.4.3 Chapter 4

In this chapter, we devise the Heavy-ball based algorithms for the network optimization applications. In particular, we consider the class of twice continuously differentiable strongly convex cost functions and linear equality constraints. These problems arise in applications such as distributed power network state-estimation and distributed averaging. In this class of problems, a number of decision-makers collaborate with neighbors in a graph to minimize a cost function over a combination of shared variables represented by the linear equality constraints. Furthermore, the sparsity pattern of the linear constraints are induced by the structure of the underlying graph.

We develop distributed multi-step method applied on the primal and dual of the original problem and derive corresponding optimal algorithm parameters. In both cases, we show that the method has linear convergence rate and present the corresponding convergence factors.

In the chapter, we also perform a robustness analysis in which the effects of perturbations in input parameters is studied on the convergence of the algorithm. This study is practically important if we notice that in many applications, algorithm parameters such as Lipschitz constants or strong convexity parameters are estimated with some bounds that are not usually tight. Finally, we apply the developed algorithms to three applications: networked resource allocation, consensus, and network flow control. In each case, we compare the performance of new algorithms to the state-of-the-art methods.The following publications contributed to this chapter.

E. Ghadimi, I. Shames, M. Johansson. Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*. vol.61, no.21, pp.5417-5429, 2013.

E. Ghadimi, M. Johansson, I. Shames. Accelerated gradient methods for networked optimization. In *Proceedings of American Control Conference (ACC)*. 2011.

1.4.4 Chapter 5

Chapter 5 presents the convergence properties of the ADMM method for quadratic problems. We show that the method converges linearly for two classes of quadratic optimization problems: ℓ_2-regularized quadratic minimization and quadratic programming with linear inequality constraints. For each problem classes, we optimize the convergence behavior of corresponding ADMM algorithm. First, we derive the optimal step-size parameter and the corresponding factor as explicit expressions. Second, we study over-relaxation technique and demonstrate how to jointly pick the step-size parameter and the over-relaxation constant to even-further decrease the convergence factor of the ADMM method. The final technique to improve the convergence speed is to precondition the constraint matrices. We formulate semi-definite programs to achieve such scaling and show its benefits. A model predictive control application validates our theoretical findings. This chapter is based on the following publication.

E. Ghadimi, A. Teixeira, I. Shames, M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. *IEEE Transactions on Automatic Control*. vol.60, no.3, pp.644-658, 2015.

1.4.5 Chapter 6

The aim of Chapter 6 is to address the best achievable performance of the ADMM method for a class of distributed quadratic programming problems that appears in network optimization. The decision makers in these applications have private and share equality constraints between each other. By analyzing these equality-constrained QP problems, we are able to characterize the optimal step-size, over-relaxation and constraint preconditioning for the associated ADMM iterations.

Specifically, since the ADMM iterations for the problems in this chapter are linear, the convergence behavior depends on the spectrum of the transition matrix. We prove that the convergence of the ADMM iterates is linear for the problem of interest. The convergence factor, however, equals to the largest magnitude of non-unity eigenvalue of the transition matrix. We derive the explicit equations describing the minimal convergence factor and corresponding optimal step-size and over-relaxation parameters. Moreover, given that the optimal step-size and relaxation parameter are chosen, we propose methods to further improve the convergence factor by optimal scaling (preconditioning).

We note that derived performance bounds in this chapter correspond to the exact fixed-point representation of the original ADMM iterates and not their worst-case surrogates. This fact, as opposed to Chapter 5, provides exact performance bounds of the ADMM algorithm which has several theoretical merits.

As a case study, we specialize the results of the chapter for the distributed averaging problem. Numerical results show that our optimized ADMM based algorithms significantly outperform several state-of-the-art distributed averaging algorithms. The following publications contribute to this chapter.

A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, M. Johansson. Optimal scaling of the ADMM algorithm for distributed quadratic programming. *IEEE Transactions on Signal Processing*. 2014. Submitted.

E. Ghadimi, A. Teixeira, M. Rabbat, and M. Johansson. The ADMM algorithm for distributed averaging: Convergence rates and optimal parameter selection. In *Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers*. 2014.

1.4.6 Chapter 7

In this chapter, we summarize the thesis by discussing the main results. We further discuss possible directions to be taken in order to extend the work started with this thesis.

1.4.7 Other publications

The following publications are not explicitly covered in the thesis. However, they certainly influenced the contents.

E. Ghadimi, O. Landsiedel, P. Soldati, S. Duquennoy, M. Johansson. Opportunistic routing in low duty-cycled wireless sensor networks. *ACM Transactions on Sensor Networks*. vol.10, no.4, pp.67:1-39, 2014.

A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, M. Johansson. Optimal scaling of the ADMM algorithm for distributed quadratic programming. In *Proceeding of IEEE Conference on Decision and Control (CDC)*. 2013.

E. Ghadimi, O. Landsiedel, P. Soldati, M. Johansson. A metric for opportunistic routing in duty cycled wireless sensor networks. In *Proceedings of the 9th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. 2012.

O. Landsiedel, E. Ghadimi, S. Duquennoy, M. Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of the 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2012.

E. Ghadimi, A. Teixeira, I. Shames, M. Johansson. On the optimal step-size selection for the alternating direction method of multipliers. In *Proceeding of IFAC Workshop on Estimation and Control of Networked Systems (NECSYS)*. 2012.

E. Ghadimi, A. Khonsari, A. Diyanat, M. Farmani, N. Yazdani. An analytical model of delay in multi-hop wireless ad hoc networks. *Wireless Networks*. vol.17, no.7, pp.1679-1697, 2011.

E. Ghadimi, P. Soldati, F. Österlind, H. Zhang, M. Johansson. Hidden terminal-aware contention resolution with an optimal distribution. In *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*. 2011.

Chapter 2
Preliminaries

IN this chapter, we briefly review the mathematical background of the thesis.The outline of the chapter is as follows. We start with the basic definitions of fixed-point iterations in Section 2.1 and then present the type of convex optimization problems considered in the thesis in Section 2.2. Section 2.3 presents the graph theoretic concepts used throughout the thesis. Section 2.4 introduces the notion of network optimization and provides several related applications to be discussed in the thesis. In Section 2.5 we discuss different decomposition techniques that are used to solve network optimization problems in the thesis. Finally, Section 2.6 summarizes the concepts presented in this chapter.

2.1 Fixed-point iterations

Consider a sequence $\{x^{(k)}\}$ converging to a fixed-point $x^\star \in \mathbb{R}^n$. The convergence factor of $\{x^{(k)}\}$ is defined as

$$\zeta \triangleq \limsup_{k\to\infty} \frac{\|x^{(k+1)} - x^\star\|}{\|x^{(k)} - x^\star\|}. \tag{2.1}$$

The sequence $\{x^{(k)}\}$ is said to converge at Q-sublinear rate if $\zeta = 1$, at Q-linear rate if $\zeta \in (0,1)$, and at Q-superlinear rate if $\zeta = 0$. Moreover, we say that convergence rate is R-linear if there is a nonnegative scalar sequence $\{\nu^{(k)}\}$ such that for all $k \geq 1$, $\|x^{(k)} - x^\star\| \leq \nu^{(k)}$ and $\{\nu^{(k)}\}$ converges Q-linearly to 0 [46] [1]. In this thesis, we often omit the letters Q and R while referring to the convergence rate.

To clarify the distinctions between the linear and sublinear convergence rates, note that a linear rate is usually given in terms of an exponential function of the iteration count, i.e., $\|x^{(k)} - x^\star\| \leq \sigma\zeta^k$ with $\zeta \in (0,1)$ and $\sigma \in \mathbb{R}_+$ such that $\|x^{(0)} - x^\star\| \leq \sigma$. A sublinear rate, on the other hand, is described in terms of a power function of the iteration count. For example, we may have $\|x^{(k)} - x^\star\| \leq \sigma/k \triangleq \mathcal{O}(1/k)$. This rate is much slower than the linear rate. For instance, in order to reach to ε-vicinity of the optimal solution, i.e., to find $i \geq 1$ such that $\|x^{(i)} - x^\star\| \leq \varepsilon$, one has to perform roughly $i \simeq \ln(1/\varepsilon)$ and $i \simeq 1/\varepsilon$ number of iterations under linear and sublinear rate $\mathcal{O}(1/k)$, respectively.

[1]The letters Q and R stand for quotient and root, respectively.

We define the ε-solution time π_ε as the smallest iteration count to ensure that $\|x^{(k)} - x^\star\| \leq \varepsilon$ holds for all $k \geq \pi_\varepsilon$, in the worst case of all initial points $x^{(0)}$ for which $\|x^{(0)} - x^\star\| \leq \sigma$. For linearly converging sequences with $\zeta \in (0, 1)$ the ε-solution time is given by

$$\pi_\varepsilon \triangleq \left\lceil \frac{\log(\sigma) - \log(\varepsilon)}{-\log(\zeta)} \right\rceil.$$

If the 0-solution time is finite for all $x^{(0)}$, we say that the sequence converges in finite time. As for linearly converging sequences $\zeta < 1$, the ε-solution time π_ε is improved by decreasing ζ.

Consider the following linear iterative process

$$x^{(k+1)} = Tx^{(k)}, \tag{2.2}$$

where $x^{(k)} \in \mathbb{R}^n$ and $T \in \mathcal{S}^n$. Assume T has $m < n$ eigenvalues at 1 and let $V \in \mathbb{R}^{n\times m}$ be a matrix whose columns span the 1-eigenspace of T so that $TV = V$[2].

Next we determine the properties of T such that, for any given starting point $x^{(0)}$, the iteration in (2.2) converges to a fixed-point that is the projection of the $x^{(0)}$ into the 1-eigenspace of T, i.e.

$$x^\star \triangleq \lim_{k\to\infty} x^{(k)} = \lim_{k\to\infty} T^k x^{(0)} = \Pi_{\mathrm{Im}(V)} x^{(0)}. \tag{2.3}$$

Proposition 2.1

The iterations (2.2) converge to a fixed-point in $\mathrm{Im}(V)$ if and only if

$$V^\top T = V^\top, \quad TV = V, \quad r\left(T - \Pi_{\mathrm{Im}(V)}\right) < 1, \tag{2.4}$$

where $r(\cdot)$ denotes the spectral radius of a matrix.

Proof. The result is an extension of [47, Theorem 1] for the case of 1-eigenspace of T with dimension $m > 1$. First, we consider the sufficiency. Since $TV = V$ we have

$$\begin{aligned}
T^k - V(V^\top V)^{-1}V^\top &= T^k(I - V(V^\top V)^{-1}V^\top) \\
&\overset{(a)}{=} T^k(I - V(V^\top V)^{-1}V^\top)^k \\
&= \left(T\left(I - V(V^\top V)^{-1}V^\top\right)\right)^k \\
&= (T - V(V^\top V)^{-1}V^\top)^k,
\end{aligned}$$

where (a) uses the fact that $I - V(V^\top V)^{-1}V^\top$ is a projection matrix. Now applying the condition $r(T - V(V^\top V)^{-1}V^\top) < 1$ leads to the convergence result.

For the necessary part, note that $\lim_{k\to\infty} T^k$ exists if and only if there exist a nonsingular matrix S such that

$$T = S\begin{bmatrix} I_\eta & 0 \\ 0 & \Delta \end{bmatrix} S^{-1},$$

[2]Since $T \in \mathcal{S}^n$ we also have $V^\top T = V^\top T^\top = (TV)^\top = V^\top$.

where I_η is η-dimensional identity matrix ($0 \leq \eta \leq n$) and $\Delta \in \mathbb{R}^{n-\eta \times n-\eta}$ is a convergent matrix; i.e., $r(\Delta) < 1$. The former equality can be achieved via Jordan canonical forms (see [48]). Let $u_1, u_2, \dots, u_n$ be columns of S and $v_1^\top, v_2^\top, \dots, v_n^\top$ be rows of S^{-1}. Then we have

$$\begin{aligned}\lim_{k\to\infty} T^k &= \lim_{k\to\infty} \left(S \begin{bmatrix} I_m & 0 \\ 0 & \Delta \end{bmatrix} S^{-1} \right)^k = \lim_{k\to\infty} S \begin{bmatrix} I_m & 0 \\ 0 & \Delta^k \end{bmatrix} S^{-1} = S \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix} S^{-1} \\ &= \sum_{i=1}^{m} u_i v_i^\top .\end{aligned} \tag{2.5}$$

Since $u_i v_i^\top$ is a rank one matrix and the summation $\sum_{i=1}^n u_i v_i^\top = SS^{-1} = I$ is of rank n, the matrix $\sum_{i=1}^\eta u_i v_i^\top$ must have rank η. Comparing (2.3) and (2.5) reveals that $\eta = m$ and $\sum_{i=1}^m u_i v_i^\top = V(V^\top V)^{-1} V^\top$. Equivalently, it indicates that u_i and $v_i^\top$ for $i = 1, \dots m$ are pairs of right and left eigenvectors of T corresponding to 1-eigenvalue. Moreover, it follows that

$$r\left(T - V(V^\top V)^{-1} V^\top\right) = r\left(S \begin{bmatrix} 0 & 0 \\ 0 & \Delta \end{bmatrix} S^{-1} \right) = r(\Delta) < 1.$$

which is precisely (2.4). The proof is complete. □

Proposition 2.1 shows that when $T \in \mathcal{S}^n$, the fixed-point iteration (2.2) is guaranteed to converge to a point given by (2.3) if all the non-unitary eigenvalues of T have magnitudes strictly smaller than 1. From (2.2) one sees that

$$\begin{aligned} x^{(k+1)} - x^\star &= Tx^{(k)} - \Pi_{\mathrm{Im}(V)} x^{(0)} \overset{(a)}{=} \left(T - \Pi_{\mathrm{Im}(V)}\right) x^{(k)} \\ &= \left(T - \Pi_{\mathrm{Im}(V)}\right) (x^{(k)} - x^\star), \end{aligned}$$

where (a) holds due to $V^\top T = V^\top$. Hence, the convergence factor of (2.2) is the modulus of the largest non-unit eigenvalue of the symmetric matrix T.

One approach for improving the convergence properties of (2.2) is to also account for past iterates when computing the next ones. The approach is called relaxation and performs the following iterations

$$x^{(k+1)} = \alpha T x^{(k)} + (1 - \alpha) x^{(k)},$$

where $\alpha \in (0, 1]$ is called the relaxation parameter. Note that setting $\alpha = 1$ yields the original linear iterates (2.2). In Chapters 4, 5 and 6, we will present different techniques to minimize the quantity of convergence factor with respect to some design parameters including the relaxation parameter.

2.2 Convex optimization

This section explains basic definitions of convex optimization and related algorithms. The complete road map of these topics can be found in [4, 49].

2.2.1 Basic definitions

We start with defining the convex sets and convex functions.

Definition 2.1 A set $\mathcal{X} \subseteq \mathbb{R}^n$ is convex if for all $x, y \in \mathcal{X}$,

$$\theta x + (1-\theta)y \in \mathcal{X},$$

for any scalar $\theta \in [0, 1]$.

Definition 2.2 A function $f(x) : \mathbb{R}^n \to \mathbb{R}$ defined on the convex domain $\mathcal{X} \subseteq \mathbb{R}^n$ is convex if for all $x, y \in \mathcal{X}$,

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y),$$

for any scalar $\theta \in [0, 1]$.

Note that if the preceding inequality holds with strict inequality then we say f is strictly convex. Also if $-f$ is convex, then we say f is concave.

A generic optimization problem is usually formulated as

$$\underset{x \in \mathcal{X}}{\text{minimize}} \;\; f(x). \tag{2.6}$$

If the feasible set $\mathcal{X}$ is convex and f is a convex (concave) objective function then the minimization (maximization) problem in hand is called the convex optimization problem. One nice property of convex problems is that any local minimum point of a convex optimization problem is also a global minimum point, i.e., a solution point of the problem (see e.g., [5, Proposition 2.1.2]).

In this thesis, besides the convexity assumption of the optimization problem, we require that the objective f is a continuously differentiable convex function. Moreover, the objective function f may fulfill extra smoothness properties defined as the following.

Definition 2.3 We say that $f : \mathbb{R}^n \to \mathbb{R}$ belongs to the class $\mathcal{F}_L^{1,1}$, if it is convex, continuously differentiable, and its gradient is Lipschitz continuous with constant L, i.e.,

$$0 \leq f(y) - f(x) - \langle \nabla f(x),\, y - x\rangle \leq \frac{L}{2}\|x-y\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

In addition, if f is also strongly convex with modulus $\mu > 0$, i.e.,

$$f(x) + \langle \nabla f(x),\, y - x\rangle + \frac{\mu}{2}\|x-y\|^2 \leq f(y), \quad \forall x, y \in \mathbb{R}^n,$$

then, we say that f belongs to $\mathcal{S}_{\mu,L}^{1,1}$[3].

[3]The symbols $\mathcal{F}_L^{1,1}$ and $\mathcal{S}_{\mu,L}^{1,1}$ are adopted from [50]. Essentially, having a convex function $f \in \mathcal{F}_L^{k,p}$ means that f is k times continuously differentiable and that its p-th derivative is Lipschitz continuous with the constant L. A similar description holds for $f \in \mathcal{S}_{\mu,L}^{k,p}$ by additionally noting that f is also strongly convex with constant μ.

Aforementioned conditions for an optimization problem might appear restrictive. But surprisingly many real-world engineering problems fulfill these functional properties [4] and there exists many powerful and efficient schemes to solve these problems. The next sections review two important classes of optimization methods suitable for solving large-scale and distributed optimization problems. We refer the interested readers to [4, 50, 45] for a complete description of different solution methods to convex optimization problems.

2.2.2 First-order methods

In this thesis we consider convex optimization algorithms that only utilize first-order information i.e., the gradient (sub-gradient) of the objective functions. The first-order methods are among the earliest algorithms developed to solve optimization problems. Their simplicity and efficiency makes them attractive to various engineering communities.

Our baseline first-order method, in the thesis, is the gradient descent:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)}), \tag{2.7}$$

where α is a positive step-size parameter. Let $x^\star$ be an optimal point of an unconstrained convex optimization problem and $f^\star = f(x^\star)$. If $f \in \mathcal{F}_L^{1,1}$, then $f(x^{(k)}) - f^\star$ associated with the sequence $\{x^{(k)}\}$ in (2.7) converges at rate $\mathcal{O}(1/k)$ where k is the number of performed iterates (a similar result for constrained convex optimization problems with $f \in \mathcal{F}_L^{1,1}$was shown in [51]).

On the other hand, if $f \in \mathcal{S}_{\mu,L}^{1,1}$, then the sequence $\{x^{(k)}\}$ generated by the gradient descent method converges linearly, i.e., there exists $q \in [0, 1)$ such that

$$\|x^{(k)} - x^\star\| \leq q^k \|x^{(0)} - x^\star\|, \quad k \in \mathbb{N}_0.$$

Recall from Section 2.1 that the scalar q is called the convergence factor. The optimal gradient step-size parameter and the associated convergence factor for $f \in \mathcal{S}_{\mu,L}^{1,1}$is reported as (see [52])

$$\alpha = \frac{2}{L+\mu}, \quad q = \frac{L-\mu}{L+\mu}. \tag{2.8}$$

The convergence of the gradient iterates can be accelerated by accounting for the history of iterates when computing the ones to come. Methods in which the next iterate depends not only on the current iterate but also on the preceding ones are called multi-step methods. The simplest multi-step extension of gradient descent is Polyak's Heavy-ball method [52]:

$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)}) + \beta \left(x^{(k)} - x^{(k-1)} \right), \tag{2.9}$$

for constant parameters $\alpha, \beta \in \mathbb{R}_{++}$. For the class of twice continuously differentiable strongly convex functions with Lipschitz continuous gradient, Polyak used a local analysis based on bounds on the norm of the Hessian of the objective function to derive optimal step-size parameters. He showed that the optimal convergence factor of the Heavy-ball iterates and the associated step-size parameters are

$$\alpha = \left(\frac{2}{\sqrt{L}+\sqrt{\mu}} \right)^2, \quad \beta = \left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}} \right)^2, \quad q = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}, \tag{2.10}$$

where μ and L are the lower and the upper bounds on the Hessian of the objective function[4]. This convergence factor is always smaller than the one associated with the gradient iterates. Note that this convergence analysis holds globally if the Hessians are constant, i.e., QPs with positive definite Hessians. For general cases of $f \in \mathcal{F}_L^{1,1}$ and $f \in \mathcal{S}_{\mu,L}^{1,1}$, however, Polyak's analysis only holds locally.

In contrast, Nesterov's fast gradient method [50] is a first-order method with better global convergence guarantees than the basic gradient method for objectives in $\mathcal{F}_L^{1,1}$ and $\mathcal{S}_{\mu,L}^{1,1}$ classes. In its simplest form, Nesterov's algorithm with constant step-sizes takes the form

$$\begin{aligned} y^{(k+1)} &= x^{(k)} - \alpha \nabla f(x^{(k)}), \\ x^{(k+1)} &= y^{(k+1)} + \beta(y^{(k+1)} - y^{(k)}), \end{aligned} \tag{2.11}$$

with $\alpha > 0$ and $\beta > 0$. When $f \in \mathcal{S}_{\mu,L}^{1,1}$, Nesterov [50] proved a global linear convergence rate towards the optimal point for the iterates produced by (2.11) with the following step-sizes and convergence factor

$$\alpha = \frac{1}{L}, \quad \beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}, \quad q = 1 - \sqrt{\frac{\mu}{L}}. \tag{2.12}$$

This factor is smaller than that of the gradient, but larger than that of the Heavy-ball method. A better local convergence factor of Nesterov's method for twice continuously differentiable strongly convex functions with Lipschitz continuous gradient is achievable with (see e.g., [53])

$$\alpha^\star = \frac{4}{3L + \mu}, \quad \beta^\star = \frac{1 - \sqrt{\mu\alpha^\star}}{1 + \sqrt{\mu\alpha^\star}}, \quad q = 1 - 2\sqrt{\frac{\mu}{3L + \mu}}. \tag{2.13}$$

This convergence factor is better than the one of gradient method but still worse than the one of the Heavy-ball method.

2.2.3 Alternating Direction Method of Multipliers

This section presents the background to the celebrated ADMM method for solving structured and large-scale problems. These concepts will be used later in Chapters 5 and 6 to optimize the performance of the ADMM algorithm (See [45] for a detailed review of the technique). The ADMM algorithm solves problems of the form

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c, \end{array} \tag{2.14}$$

where f and g are convex functions, $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$ and $c \in \mathbb{R}^p$.

[4]Here f is assumed to belong to the class $\mathcal{S}_{\mu,L}^{2,1}$ which is a stronger assumption than $f \in \mathcal{S}_{\mu,L}^{1,1}$.

Relevant examples that appear in this form are, e.g., regularized estimation, where f is the estimator loss and g is the regularization term, and various network optimization problems, e.g., [54, 45]. The method is based on the augmented Lagrangian

$$L_\rho(x,z,y) = f(x) + g(z) + \frac{\rho}{2}\|Ax + Bz - c\|^2 + y^\top(Ax + Bz - c),$$

with the penalty parameter $\rho \in \mathbb{R}_+$ and Lagrange multipliers $y \in \mathbb{R}^p$, and performs sequential minimization of the x and z variables followed by a dual variable update:

$$\begin{aligned} x^{(k+1)} &= \underset{x}{\operatorname{argmin}}\, L_\rho(x, z^{(k)}, y^{(k)}), \\ z^{(k+1)} &= \underset{z}{\operatorname{argmin}}\, L_\rho(x^{(k+1)}, z, y^{(k)}), \\ y^{(k+1)} &= y^{(k)} + \rho(Ax^{(k+1)} + Bz^{(k+1)} - c), \end{aligned} \tag{2.15}$$

for some arbitrary $x^{(0)} \in \mathbb{R}^n$, $z^{(0)} \in \mathbb{R}^m$, and $y^{(0)} \in \mathbb{R}^p$. It is often convenient to express the iterations in terms of the scaled dual variable $u = y/\rho$:

$$\begin{aligned} x^{(k+1)} &= \underset{x}{\operatorname{argmin}} \left\{ f(x) + \frac{\rho}{2}\|Ax + Bz^{(k)} - c + u^{(k)}\|^2 \right\}, \\ z^{(k+1)} &= \underset{z}{\operatorname{argmin}} \left\{ g(z) + \frac{\rho}{2}\|Ax^{(k+1)} + Bz - c + u^{(k)}\|^2 \right\}, \\ u^{(k+1)} &= u^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c. \end{aligned} \tag{2.16}$$

ADMM is particularly useful when the x- and z-minimizations can be carried out efficiently, for example when they admit closed-form expressions. Examples of such problems include linear and quadratic programming, basis pursuit, ℓ_1-regularized minimization, and model fitting problems to name a few.

One advantage of the ADMM method is that there is only a single algorithm parameter, ρ, and under rather mild conditions, the method can be shown to converge for all values of the parameter; see [45, 55] and references therein. This contrasts the gradient method whose iterates diverge if the step-size parameter is chosen too large. However, ρ has a direct impact on the convergence factor of the algorithm, and inadequate tuning of this parameter can render the method slow. The convergence of ADMM is often characterized in terms of the residuals

$$r^{(k+1)} = Ax^{(k+1)} + Bz^{(k+1)} - c, \tag{2.17}$$

$$s^{(k+1)} = \rho A^\top B(z^{(k+1)} - z^{(k)}), \tag{2.18}$$

termed the primal and dual residuals, respectively [45]. One approach for improving the convergence properties of the algorithm is to use the relaxation technique introduced earlier in this chapter. In particular, the relaxation technique in the ADMM context works by replacing $Ax^{(k+1)}$ with $h^{(k+1)} = \alpha^{(k)}Ax^{(k+1)} - (1-\alpha^{(k)})(Bz^{(k)} - c)$ in the z- and u-updates [45], yielding

$$\begin{aligned} z^{(k+1)} &= \underset{z}{\operatorname{argmin}} \left\{ g(z) + \frac{\rho}{2}\left\| h^{(k+1)} + Bz - c + u^{(k)} \right\|^2 \right\}, \\ u^{(k+1)} &= u^{(k)} + h^{(k+1)} + Bz^{(k+1)} - c. \end{aligned} \tag{2.19}$$

The parameter $\alpha^{(k)} \in (0,2)$ is called the relaxation parameter. Note that letting $\alpha^{(k)} = 1$ for all k recovers the original ADMM iterations (2.16).

Empirical studies show that over-relaxation, i.e., letting $\alpha^{(k)} > 1$, is often advantageous and the guideline $\alpha^{(k)} \in [1.5, 1.8]$ has been proposed [56]. An alternative way of accelerating the ADMM iterates is to assign different quantities to the penalty parameter and the dual step-size parameter. While certain accelerations in the algorithm are reported in the literature the stability of the algorithm is also restricted by this design choice [57]. In Chapters 5 and 6 we present different techniques to tune the ADMM algorithm parameters for two classes of quadratic programing.

2.3 Graphs

In the thesis, we often deal with optimization problems to be solved in networked systems. Graph theory provides a powerful framework to model connected peers in the network and the physical correlations among them. In this section, we briefly introduce graph theoretical concepts and refer the interested reader to [58, 59] for thorough discussions on this topic.

An undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ consists of a vertex (node) set $\mathcal{V}$ and an edge set $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$. The pair $\{i,j\} \in \mathcal{E}$ represents an edge between vertices $i, j \in \mathcal{V}$. Sometimes it is convenient to consider a given ordering of the edges of $\mathcal{G}$, in which case the k-th edge is denoted by $e_k \in \mathcal{E}$. We consider graphs with only finitely many vertices. A path in $\mathcal{G}$ from $i \in \mathcal{V}$ to $j \in \mathcal{V}$ is a sequence $i, \dots, j$ of distinct vertices such that there exists an edge between each successive two vertices k, l in the sequence; i.e., $\{k,l\} \in \mathcal{E}$. A graph is called connected if there exists a path between each pair of the nodes in the graph. Let $\mathcal{N}_i \triangleq \{j \neq i | \{i,j\} \in \mathcal{E}\}$ be the neighbor set of node $i \in \mathcal{V}$. The sparsity pattern induced by $\mathcal{G}$ is defined as

$$\mathcal{A} \triangleq \{S \in \mathcal{S}^{|\mathcal{V}|} | S_{ij} = 0 \,\text{if}\, i \neq j \,\text{and}\, \{i,j\} \notin \mathcal{E}\}.$$

The adjacency matrix $A \in \mathcal{A}$ of a graph $\mathcal{G}$ is a binary matrix (with 0 and 1 entries) given by

$$A_{ij} = \begin{cases} 1 & \text{if}\, i \neq j \,\text{and}\, \{i,j\} \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

The corresponding diagonal degree matrix D is given by $D_{ii} = \sum_{j \in \mathcal{N}_i} A_{ij}$.

Another useful matrix in the context of graph theory is the incidence matrix. Assign arbitrary orientations to the edges $\{i,j\} \in \mathcal{E}$ then the oriented incidence matrix and the (unsigned) incidence matrix are defined as $\bar{B} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|} \triangleq \hat{B} - \mathring{B}$ and $B \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|} \triangleq \hat{B} + \mathring{B}$, respectively. Moreover, $\mathring{B}_{kj} = 1$ if j is the head of $e_k \in \mathcal{E}$ and $\mathring{B}_{kj} = 0$ otherwise. Similarly, $\hat{B} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|}$ is defined such that $\hat{B}_{kj} = 1$ if j is the tail of $e_k \in \mathcal{E}$ and $\hat{B}_{kj} = 0$ otherwise. Note that with this definition, $\bar{B}$ is not unique with respect to $\mathcal{G}$. All oriented incidence matrices of $\mathcal{G}$ differ only by negating some set of columns. According to their construction, the following relations between adjacency matrix $A \in \mathcal{A}$, degree matrix D and incidence matrix B holds

$$\mathring{B}^\top \hat{B} + \hat{B}^\top \mathring{B} = A, \quad \mathring{B}^\top \mathring{B} + \hat{B}^\top \hat{B} = D.$$

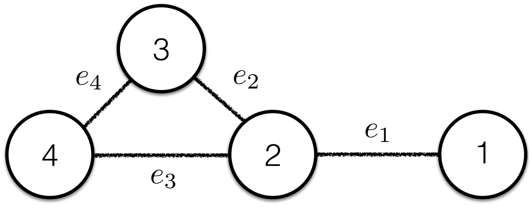


Figure 2.1: An example graph $\mathcal{G}$ with $|\mathcal{V}| = 4$ and $|\mathcal{E}| = 4$.

The Laplacian matrix $\mathcal{L}$ of a graph $\mathcal{G}$ is another important matrix which is defined as $\mathcal{L} \triangleq D - A = \bar{B}^\top \bar{B}$. Aforementioned matrices have several interesting properties from which one can say a great deal about the graph in hand [60, 59]. This is, however, out of the scope of the current thesis.

Example 2.1 An example graph $\mathcal{G}$ is shown in Figure.2.1. The associated adjacency, incidence, and Laplacian matrices are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}.$$

In Chapters 4 and 6, we use asymmetric weights on the links in order to accelerate the convergence rate of certain fixed-point iterates. Therefore, along with the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, we introduce an associated weighted directed graph $\bar{\mathcal{G}}(\mathcal{V}, \bar{\mathcal{E}}, \mathcal{W})$. The edge set $\bar{\mathcal{E}}$ of $\bar{\mathcal{G}}$ contains two directed edges (i, j) and (j, i) for each undirected edge $\{i, j\} \in \mathcal{E}$, such that for each $e_k = \{i, j\}$ with $1 \leq k \leq |\mathcal{E}|$, we have $\bar{e}_k = (i, j)$ and $\bar{e}_{k+|\mathcal{E}|} = (j, i)$. The edge weights $\mathcal{W} = \{W_{(i,j)}\}_{(i,j)\in\bar{\mathcal{E}}}$ comprise matrix-valued weights $W_{(i,j)} \in \mathcal{S}_+^{n_w}$ with some appropriate dimension n_w for each directed edge $(i, j) \in \bar{\mathcal{E}}$. Moreover, the edge-weight matrix is defined as $W \triangleq \text{diag}\left(\{W_{\bar{e}_k}\}_{k=1}^{|\bar{\mathcal{E}}|}\right)$.

For $n_w = 1$ and symmetric weights $W_{\{i,j\}} = W_{(i,j)} = W_{(j,i)} \in \mathbb{R}_+$, $\bar{\mathcal{G}}$ is equivalent to a weighted undirected graph whose adjacency matrix $A \in \mathcal{A}$ is defined as $A_{ij} = W_{\{i,j\}}$ for $\{i, j\} \in \mathcal{E}$ and $A_{ii} = 0$. The diagonal degree matrix D is defined as before $D_{ii} = \sum_{j\in\mathcal{N}_i} W_{\{i,j\}}$.

2.4 Network optimization

Network optimization refers to collaborative optimization by a network of decision-makers. Each decision-maker i is endowed with a loss function f_i, has control of one decision-variable x_i, and collaborates with the others to solve

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & Ax = b, \end{array} \tag{2.20}$$

for given matrices A and b with appropriate sizes. We will assume that b lies in the range space of A, i.e., that there exists at least one decision vector x that satisfies the constraints. The physical information exchange between decision-makers is represented by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Specifically, we will assume that decision-maker i only communicate with neighboring nodes, and that the edges can be interpreted as bidirectional communication links. That is if $\{i, j\} \in \mathcal{E}$ then both decision makers i and j can communicate with one another. Since we are interested in finding a global optimal solution to (2.20), each decision maker should be able to coordinate with other ones. Therefore, we make the following assumption [61].

Assumption 2.1 The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is connected.

In what follows, we introduce several network optimization problems that are considered later in the thesis.

2.4.1 Collaborative minimization over a global shared variable

Consider a network of agents, each endowed with a local convex loss function $f_i(x)$, that collaborate to find the decision vector x that results in the minimal total loss, i.e.,

$$\underset{x \in \mathbb{R}^{n_x}}{\text{minimize}} \quad \textstyle\sum_{i \in \mathcal{V}} f_i(x). \tag{2.21}$$

The interactions among agents are described by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

One natural way of distributing the computation of (2.21) among the agents is by introducing local copies $x_i \in \mathbb{R}^{n_x}$ of the global decision vector at each node $i \in \mathcal{V}$. Then, the original problem (2.21) can be re-written as an equality constrained optimization problem with decision variables $\{x_i\}_{i \in \mathcal{V}}$ and separable objective:

$$\begin{array}{ll} \underset{\{x_i\}}{\text{minimize}} & \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{subject to} & x_i = x_j, \quad \forall\, i, j \in \mathcal{V}. \end{array} \tag{2.22}$$

The equality constraints ensure that the local decision vectors x_i of all agents agree at optimum. Note that when the communication graph is connected (Assumption 2.1) all equality constraints in (2.22) that do not correspond to neighboring nodes in $\mathcal{G}$ can be removed without altering the optimal solution. The remaining inequality constraints can be accounted for in different ways that will be discussed in details in chapters 4 and 6. Moreover, we devise different distributed optimization algorithms that accelerate solving the network-wide agreement problem (2.21).

2.4.2 Optimal resource allocation

When the decision-makers are only constrained by a global resource budget, (2.20) reduces to

$$\begin{array}{ll} \text{minimize} & \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{subject to} & \sum_{i \in \mathcal{V}} x_i = x_{\text{tot}}. \end{array} \tag{2.23}$$

When the functions $f_i : \mathbb{R} \to \mathbb{R}$ are strictly convex and twice differentiable with bounded second derivatives then the problem has a unique optimal solution that can be find in an efficient distributed manner.

A distributed algorithm for this problem was developed in [62] and interpreted as a weighted gradient method in [63]. Later, in Chapter 4 we design a distributed algorithm that solves (2.23) in a faster way.

2.4.3 Internet congestion control

Our final network optimization application is devoted to the area of Internet congestion control, where Network Utility Maximization (NUM) has emerged as a powerful framework for studying various important resource allocation problems, see, e.g., [64, 15, 13, 14]. The vast majority of the work in this area is based on the distributed approach introduced in [15]. Here, the optimal bandwidth sharing among S flows in a data network is posed as the optimal solution to the convex optimization problem

$$\begin{array}{ll} \underset{x}{\text{maximize}} & \sum_s u_s(x_s) \\ \text{subject to} & x_s \in [m_s, M_s] \\ & Rx \leq c. \end{array} \tag{2.24}$$

Here, x_s is the communication rate of flow s, and $u_s(x_s)$ is a strictly concave and increasing function that describes the utility that source s has if communicate at rate x_s. The communication rate is subject to upper and lower bounds. Finally, $R \in \{0, 1\}^{L \times S}$ is a routing matrix whose entries $R_{\ell s}$ are 1 if flow s traverses link ℓ and are 0 otherwise. In this way, Rx is the total traffic on links, which cannot exceed the link capacities $c \in \mathbb{R}^n$.

In this thesis, we consider the cases where the utility functions are twice continuously differentiable, the routing matrix has full row-rank and all the link constraints hold with equality at optimum. Hence, we can replace $Rx \leq c$ in (2.24) with $Rx = c$ and, consequently, this problem falls under the umbrella of the network optimization problem (2.20). In Chapter 4 we develop techniques that lead into fast distributed algorithms to solve (2.24).

2.5 Decomposition techniques

Decomposition techniques refer to methods by which a central optimization problem is split into several subproblems that are solved by separate processors or agents. Even though the subproblems can be solved independently they usually need to coordinate in order to reach to a global optimal solution.

During past years, fueled by extraordinary advances in the fields of telecommunication and wireless networking, several decomposition techniques such as primal, dual, primal-dual decomposition have been developed that efficiently solve many engineering problems. See [65, 66] and [67, Chapter 6] for a survey on decomposition methods and related engineering problems. In this section, we briefly introduce primal and dual decomposition methods for a class of problems that will be discussed later in this thesis.

In particular, consider the following constrained convex optimization problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) = \sum_{i=1}^{N} f_i(x(i)) \\ \text{subject to} & Ax = b, \end{array} \tag{2.25}$$

with variable $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ and $f : \mathbb{R}^n \to \mathbb{R}$ is convex. Let us assume that x is separable into N sub-vectors $x = (x(1), \ldots, x(N))$ with dimensions $x(i) \in \mathbb{R}^{n_i}$. Moreover, let $\mathcal{I}_i$ denote the set of indices of x belonging to the sub-vector i.

Partitioning A, accordingly, into N blocks $A = (A(1), \ldots, A(N))$, the identity $Ax = \sum_{i=1}^{N} A(i)x(i)$ holds for the linear equality constraint in the original optimization problem (2.25). We will use this setting to explain the decomposition methods for solving the convex problem (2.25).

2.5.1 Dual decomposition

In dual decomposition, the original problem is solved by subproblems using Lagrangian relaxation. The Lagrangian of our optimization problem (2.25) can be written as

$$L(x, y) = \left(\sum_{i=1}^{N} f_i(x(i)) + y^\top \left(A(i)x(i) - \frac{1}{N}b\right)\right) = \sum_{i=1}^{N} L_i(x(i), y),$$

where y is called the Lagrange multiplier (see [4] for the background). Clearly, the Lagrangian is separable in x. This indicates that the problem can be solved separately in N parallel sub-problems. To solve the problem, one forms the dual function and dual problem pair

$$d(y) = \inf_x L(x, y), \quad \underset{y\in\mathbb{R}^m}{\text{maximize}}\ d(y).$$

Assuming that strong duality holds, the optimal values of the primal and dual problems coincide. One then can recover a primal optimal variable $x^\star$ from the optimal dual solution $y^\star$ as

$$x^\star = \underset{x}{\text{argmin}}\ L(x, y^\star).$$

In the dual decomposition method, the dual problem is solved by employing gradient ascent algorithm (see e.g., [4]). Assuming that f is differentiable then d is differentiable and the gradient $\nabla d(y)$ can be evaluated for each value of x. The dual ascent method then reads

$$\begin{aligned} x(i)^{(k+1)} &= \underset{x(i)}{\text{argmin}}\ L_i(x(i), y^{(k)}) \\ y^{(k+1)} &= y^{(k)} + \alpha^{(k)}\left(\sum_{i=1}^{N} A(i)x(i)^{(k)} - \frac{1}{N}b\right), \end{aligned} \tag{2.26}$$

where $\alpha^{(k)} \in \mathbb{R}_+$ is a step-size. Provided that $\alpha^{(k)}$ is picked appropriately, $(x^{(k)}, y^{(k)})$ converges to an optimal primal-dual pair.

The first step (2.26) is an x-minimization step, which is carried out in parallel for each $i = 1, ..., N$. The second step is the dual variable update. The method is called dual decomposition because it mixes parallel decomposed steps with a dual variable update step.

Note that in the dual update step (2.26), the equality constraint contributions $A(i)x(i)^{(k)}$ are collected by a master node in order to carry out the summation term in the dual update. Once the dual update $y^{(k+1)}$ took place, it must be disseminated to the local processors $1, \ldots, N$ to compute local $x(i)$-minimization steps.

Another main point that has to be considered while using dual decomposition is that even though the final primal-dual optimality is guaranteed (if several aforementioned assumptions hold) the intermediate x-updates may not be even feasible. This is not a problem in the primal decomposition method that is discussed next.

2.5.2 Primal decomposition

In primal decomposition, the subproblems are formulated using the original (primal) variables. The main advantage of primal decomposition compared to dual decomposition is that at each intermediate updates of resulting algorithms the primal feasibility is maintained. There are various interpretations and solution methods associated with primal decomposition in the literature [8, 66]. Here we present a version of scaled gradient method applicable to our network optimization problem (2.25).

One natural way of maintaining the primal feasibility in the literature is to employ gradient-descent updates accompanied by Euclidean projection onto the constraint set. Computing the Euclidean projection onto the constraint of (2.25) typically requires the full decision vector x, which is not available to the decision-makers in our problem setting.

A primal decomposition technique, explored e.g., in [63, 68], is to consider weighted gradient methods which use a linear combination of the information available to nodes to ensure that iterates remain feasible.

In general, the weighted gradient method takes the form

$$x^{(k+1)} = x^{(k)} - \alpha W \nabla f(x^{(k)}), \tag{2.27}$$

where $W \in \mathcal{S}^n$ is a scaling matrix. For our problem (2.25), to be able to use this technique, W should satisfy the following conditions:

1. the locality of information exchange between the decision makers should be preserved. Assuming that graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ represents the physical communication between the decision makers $1, \ldots, N$ in (2.25) then if $i \neq j$ and $\{i, j\} \notin \mathcal{E}$ then for all $k \in \mathcal{I}_i$ and $l \in \mathcal{I}_j$ we should have $W_{k,l} = 0$.

2. provided that the initial point $x^{(0)}$ is feasible, the iterates generated by the weighted gradient should remain feasible. It is easily verifiable that if $AW = 0$ then the identity $Ax^{(k+1)} = Ax^{(k)}$ is guaranteed which results in the primal feasibility of each iterate in (2.27).

3. the fixed-points of (2.27) should satisfy the optimality conditions of the original problem (2.25). From the optimality conditions of the Lagrangian of (2.25) we have that pair $(x^\star, y^\star)$ are optimal primal-dual variables if

$$Ax^\star = b, \quad \nabla f(x^\star) = -A^\top y^\star.$$

The first condition always holds by the per-iterate feasibility of the weighted gradient method. Replacing the second condition in (2.27) with $x^{(k)} = x^\star$ yields

$$x^{(k+1)} = x^\star + \alpha W A^\top y^\star,$$

indicating that if $WA^\top = \mathbf{0}$ then the fixed-point identity holds.

To sum up, if W has the same sparsity pattern as the information graph $\mathcal{G}$, and

$$AW = \mathbf{0}, \quad WA^\top = \mathbf{0}, \tag{2.28}$$

then one can derive the following distributed weighted gradient method that solves (2.25):

$$x(i)^{(k+1)} = x(i)^{(k)} - \alpha \sum_{j \in \{i \cup \mathcal{N}_i\}} W_{(i,j)} \nabla f_j(x(j)^{(k)}),$$

where $W_{(i,j)} \in \mathbb{R}^{n_i \times n_j}$ and $\nabla f_i \in \mathbb{R}^{n_i}$ are the block weight and gradient matrices storing corresponding sub-vector elements.

As the reader may recognize although primal decomposition method enjoys the feasible x-updates, its applicability is subject to the restrictive assumptions (2.28). In Chapter 4, we study examples where such a technique is applicable.

2.6 Summary

In this chapter we reviewed essential definitions and concepts related to the thesis. We walked through the fixed-point iterations and defined the performance metrics such as convergence rate and factors. We then introduced basic results in convex optimization and reviewed first-order methods as well as the ADMM method to solve large-scale convex optimization problems. We also presented required backgrounds to graph theory and termed the network optimization problems. Finally, the decomposition techniques to solve network optimization problems were discussed.

Chapter 3
First-order methods: convergence bounds and accelerations

THE aim of this chapter is to study some open problems in global convergence of first-order methods. In particular, We provide a global convergence analysis for the Heavy-ball method on convex optimization problems with Lipschitz-continuous gradient, with and without the additional assumption of strong convexity. We show that if the parameters of the Heavy-ball method are chosen within certain ranges, the running average of the iterates converge to the optimal point at the rate $\mathcal{O}(1/k)$ when the objective function has Lipschitz continuous gradient. A similar result is shown for the Nesterov's method with constant step-sizes. Moreover, for the same class of problems, we are able to show that the individual iterates themselves converge at rate $\mathcal{O}(1/k)$ if the Heavy-ball method uses (appropriately chosen) time-varying step-sizes. Finally, if the cost function is also strongly convex, we show that the Heavy-ball iterates converge globally at a linear rate.

The rest of the chapter is organized as follows. Section 3.1 reviews the related work in first-order convex optimization algorithms. When the objective functions have Lipschitz continuous gradients, global convergence proofs for the Heavy-ball algorithm and Nesterov's method with constant step-sizes are presented in Section 3.2 and Section 3.3, respectively. For objective functions that are also strongly convex, we provide global convergence proofs for the Heavy-ball iterates in Section 3.4. Section 3.5 concludes the chapter with a brief statement of the results.

3.1 Related work

First-order convex optimization methods have a rich history dating back to 1950's [69, 70, 71]. Recently, these methods have attracted significant interest, both in terms of new theory [72, 73, 74] and in terms of applications in numerous areas such as signal processing [51], machine learning [75] and control [76]. One reason for this renewed interest is that first-order methods have a small per-iteration cost and are attractive in large-scale and distributed settings. But the development has also been fueled by the advance of accelerated methods with optimal convergence rates [77] and re-discovery of methods that are not only order-optimal, but also have optimal convergence times for smooth convex problems [78]. In spite of all this progress, some very basic questions about the achievable convergence speed of first-order convex optimization methods are still open [74].

The basic first-order method is the gradient descent algorithm. For unconstrained convex optimization problems with objective functions that have Lipschitz-continuous gradient, the method produces iterates that are guaranteed to converge to the optimum at the rate $\mathcal{O}(1/k)$, where k is the number of iterations. When the objective function is also strongly convex, the iterates are guaranteed to converge at a linear rate [52].

In the early 1980's, Nemirovski and Yudin [79] proved that no first-order method can converge at a rate faster than $\mathcal{O}(1/k^2)$ on convex optimization problems with Lipschitz-continuous gradient. This created a gap between the guaranteed convergence rate of the gradient method and what could potentially be achieved. This gap was closed by Nesterov, who presented an accelerated first-order method that converges as $\mathcal{O}(1/k^2)$ [77]. Later, the method was generalized to also attain linear convergence rate for strongly convex objective functions, resulting in the first truly order-optimal first-order method for convex optimization [50]. The accelerated first-order methods combine gradient information at the current and the past iterate, as well as the iterates themselves [50]. For strongly convex problems, Nesterov's method can be tuned to yield a better convergence factor than the gradient iteration, but it is not known how small the convergence factor can be made.

When the objective function is twice differentiable, strongly convex, and has Lipschitz continuous gradient, Polyak [78] showed that the Heavy-ball iterates converge locally at linear rate and have better convergence factor than both the gradient and Nesterov's accelerated gradient method[1]. The Heavy-ball method uses previous iterates when computing the next, but in contrast to Nesterov's method it only uses the gradient at the current iterate. Extensions of the Heavy-ball method to constrained and distributed optimization problems have confirmed its performance benefits over the standard gradient-based methods [44, 80, 81].

On the other hand, when the objective function is not necessary convex but has Lipschitz continuous gradient Zavriev et al. [82] provided sufficient conditions for the Heavy-ball trajectories to converge to a stationary point. However, there are virtually no results on the global rate of convergence of the Heavy-ball method for convex problems. Recently, Lessard et al [83] showed by an example that starting far enough from the stationary point, the Heavy-ball method does not necessarily converge on strongly convex objective functions even if one chooses step-size parameters according to Polyak's original stability criterion. In general, when the objective is not quadratic, it is not clear whether the Heavy-ball method performs better than Nesterov's method, or even than the basic gradient descent.

[1]Polyak's convergence analysis for the Heavy-ball iterates holds globally when the objective function is of quadratic form with positive definite Hessian.

3.2 Global analysis of Heavy-ball algorithm for the class $\mathcal{F}_L^{1,1}$

In this section, we consider the Heavy-ball iterates (2.9) for the objective functions $f \in \mathcal{F}_L^{1,1}$. Our first result shows that the method is indeed guaranteed to converge globally and estimates the convergence rate of the Cesáro averages of the iterates.

Theorem 3.1

Assume that $f \in \mathcal{F}_L^{1,1}$ and that

$$\beta \in [0,1), \quad \alpha \in \left(0, \frac{2(1-\beta)}{L}\right). \tag{3.1}$$

Then, the sequence $\{x^{(k)}\}$ generated by Heavy-ball iteration (2.9) satisfies

$$f(\overline{x}^{(T)}) - f^\star \le \begin{cases} \frac{\|x^{(0)}-x^\star\|^2}{2(T+1)}\left(\frac{L\beta}{1-\beta} + \frac{1-\beta}{\alpha}\right), & \text{if } \alpha \in \big(0, \frac{1-\beta}{L}\big], \\ \frac{\|x^{(0)}-x^\star\|^2}{2(T+1)(2(1-\beta)-\alpha L)}\left(L\beta + \frac{(1-\beta)^2}{\alpha}\right), & \text{if } \alpha \in \big[\frac{1-\beta}{L}, \frac{2(1-\beta)}{L}\big), \end{cases} \tag{3.2}$$

where $\overline{x}^{(T)}$ is the Cesáro average of the iterates

$$\overline{x}^{(T)} \triangleq \frac{1}{T+1}\sum_{k=0}^{T} x^{(k)}.$$

Proof. Assume that $\beta \in [0,1)$, and let

$$p^{(k)} = \frac{\beta}{1-\beta}(x^{(k)} - x^{(k-1)}), \quad k \in \mathbb{N}_0. \tag{3.3}$$

Then

$$x^{(k+1)} + p^{(k+1)} = \frac{1}{1-\beta}x^{(k+1)} - \frac{\beta}{1-\beta}x^{(k)} \overset{(2.9)}{=} x^{(k)} + p^{(k)} - \frac{\alpha}{1-\beta}\nabla f(x^{(k)}),$$

which implies that

$$\begin{aligned} \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 =& \|x^{(k)} + p^{(k)} - x^\star\|^2 + \left(\frac{\alpha}{1-\beta}\right)^2 \|\nabla f(x^{(k)})\|^2 \\ & - \frac{2\alpha}{1-\beta}\langle x^{(k)} + p^{(k)} - x^\star, \nabla f(x^{(k)})\rangle \\ \overset{(3.3)}{=}& \|x^{(k)} + p^{(k)} - x^\star\|^2 - \frac{2\alpha}{1-\beta}\langle x^{(k)} - x^\star, \nabla f(x^{(k)})\rangle \\ & - \frac{2\alpha\beta}{(1-\beta)^2}\langle x^{(k)} - x^{(k-1)}, \nabla f(x^{(k)})\rangle + \left(\frac{\alpha}{1-\beta}\right)^2 \|\nabla f(x^{(k)})\|^2. \end{aligned} \tag{3.4}$$

Since $f \in \mathcal{F}_L^{1,1}$, it follows from [50, Theorem 2.1.5] that

$$\begin{aligned} \frac{1}{L}\|\nabla f(x^{(k)})\|^2 &\le \langle x^{(k)} - x^\star, \nabla f(x^{(k)})\rangle, \\ f(x^{(k)}) - f^\star + \frac{1}{2L}\|\nabla f(x^{(k)})\|^2 &\le \langle x^{(k)} - x^\star, \nabla f(x^{(k)})\rangle, \\ f(x^{(k)}) - f(x^{(k-1)}) &\le \langle x^{(k)} - x^{(k-1)}, \nabla f(x^{(k)}))\rangle. \end{aligned} \tag{3.5}$$

Substituting the above inequalities into (3.4) yields

$$\begin{aligned} \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 \le& \|x^{(k)} + p^{(k)} - x^\star\|^2 - \frac{2\alpha(1-\lambda)}{L(1-\beta)}\|\nabla f(x^{(k)})\|^2 \\ &- \frac{2\alpha\lambda}{1-\beta}\big(f(x^{(k)}) - f^\star\big) - \frac{\alpha\lambda}{L(1-\beta)}\|\nabla f(x^{(k)})\|^2 \\ &- \frac{2\alpha\beta}{(1-\beta)^2}\big(f(x^{(k)}) - f(x^{(k-1)})\big) + \left(\frac{\alpha}{1-\beta}\right)^2 \|\nabla f(x^{(k)})\|^2, \end{aligned}$$

where $\lambda \in (0, 1]$ is a parameter which we will use to balance the weights between the first two inequities in (3.5). Collecting the terms in the preceding inequality, we find

$$\begin{aligned} \frac{2\alpha}{(1-\beta)}\big(\lambda + \frac{\beta}{1-\beta}\big)\big(f(x^{(k)}) - f^\star\big) + \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 \\ \le \frac{2\alpha\beta}{(1-\beta)^2}\big(f(x^{(k-1)}) - f^\star\big) + \|x^{(k)} + p^{(k)} - x^\star\|^2 \\ + \left(\frac{\alpha}{1-\beta}\right)\left(\frac{\alpha}{1-\beta} - \frac{2-\lambda}{L}\right)\|\nabla f(x^{(k)})\|^2. \end{aligned} \tag{3.6}$$

Note that when $\alpha \in (0, (2-\lambda)(1-\beta)/L]$, the last term of (3.6) becomes non-positive and, therefore, can be eliminated from the right-hand-side. Summing (3.6) over $k = 0, \ldots, T$ gives

$$\begin{aligned} \frac{2\alpha\lambda}{(1-\beta)}\sum_{k=0}^{T}\big(f(x^{(k)}) - f^\star\big) + \sum_{k=0}^{T}\left(\frac{2\alpha\beta}{(1-\beta)^2}\big(f(x^{(k)}) - f^\star\big) + \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2\right) \\ \le \sum_{k=0}^{T}\left(\frac{2\alpha\beta}{(1-\beta)^2}\big(f(x^{(k-1)}) - f^\star\big) + \|x^{(k)} + p^{(k)} - x^\star\|^2\right), \end{aligned}$$

which implies that

$$\frac{2\alpha\lambda}{(1-\beta)}\sum_{k=0}^{T}\big(f(x^{(k)}) - f^\star\big) \le \frac{2\alpha\beta}{(1-\beta)^2}\big(f(x^{(0)}) - f^\star\big) + \|x^{(0)} - x^\star\|^2.$$

Note that as f is convex, we have

$$(T+1)f(\overline{x}^{(T)}) \le \sum_{k=0}^{T} f(x^{(k)}). \tag{3.7}$$

It now follows that

$$f(\overline{x}^{(T)}) - f^\star \leq \frac{1}{T+1}\left(\frac{\beta}{\lambda(1-\beta)}(f(x^{(0)}) - f^\star) + \frac{1-\beta}{2\alpha\lambda}\|x^{(0)} - x^\star\|^2\right). \tag{3.8}$$

Additionally, according to [50, Lemma 1.2.3], $f(x^{(0)}) - f^\star \leq (L/2)\|x^{(0)} - x^\star\|^2$. The proof is completed by replacing this upper bound in (3.8) and setting $\lambda = 1$ for $\alpha \in \big(0, (1-\beta)/L\big]$ and $\lambda = 2 - (\alpha L)/(1-\beta)$ for $\alpha \in \big[(1-\beta)/L, 2(1-\beta)/L\big)$. □

A few remarks regarding the results of Theorem 3.1 are in order. first, a similar convergence rate can be proved for the minimum function values within T number of Heavy-ball iterates. More precisely, the sequence $\{x^{(k)}\}$ generated by (2.9) satisfies

$$\min_{0\leq k\leq T} f(x^{(k)}) - f^\star \leq \mathcal{O}\left(\frac{\|x^{(0)} - x^\star\|^2}{T}\right),$$

for all $T \in \mathbb{N}_0$. Second, for any fixed $\bar{\alpha} \in (0, 1/L]$, one can verify that the $\beta \in (0,1)$ that minimize the convergence factor (3.2) is $\beta^\star = 1 - \sqrt{\bar{\alpha}L}$ which yields the convergence factor

$$f(\overline{x}^{(T)}) - f^\star \leq \frac{1}{2(T+1)}\left(\frac{2\sqrt{\bar{\alpha}L} - \bar{\alpha}L}{\bar{\alpha}}\right)\|x^{(0)} - x^\star\|^2.$$

This convergence factor is always smaller than the one for the gradient descent method obtained by setting $\beta = 0$ in (3.2), i.e.,

$$f(\overline{x}^{(T)}) - f^\star \leq \frac{1}{2\bar{\alpha}(T+1)}\|x^{(0)} - x^\star\|^2.$$

Finally, setting $\bar{\alpha} = 1/L$ in the preceding upper bounds, we see that the factors coincide and equal the best convergence factor of the gradient descent method reported in [51].

Next, we show that our analysis can be strengthened when we use (appropriately chosen) time-varying step-sizes in the Heavy-ball method. In this case, the individual iterates $x^{(k)}$ (and not just their running average) converge with rate $\mathcal{O}(1/k)$.

Theorem 3.2

Assume that $f \in \mathcal{F}_L^{1,1}$ and that

$$\beta^{(k)} = \frac{k}{k+2}, \quad \alpha^{(k)} = \frac{\alpha^{(0)}}{k+2}, \quad k \in \mathbb{N}, \tag{3.9}$$

where $\alpha^{(0)} \in (0, 1/L]$. Then, the sequence $\{x^{(k)}\}$ generated by Heavy-ball iteration (2.9) satisfies

$$f(x^{(T)}) - f^\star \leq \frac{\|x^{(0)} - x^\star\|^2}{2\alpha^{(0)}(T+1)}. \tag{3.10}$$

Proof. The proof is similar to that of Theorem 3.1, so we will be somewhat terse. For $k \in \mathbb{N}_0$, let $p^{(k)} = k(x^{(k)} - x^{(k-1)})$. It is easy to verify that

$$x^{(k+1)} + p^{(k+1)} = x^{(k)} + p^{(k)} - \alpha^{(0)} \nabla f(x^{(k)}),$$

which together with the inequalities in (3.4) implies that

$$\begin{aligned} 2\alpha^{(0)}(k+1)\big(f(x^{(k)}) - f^\star\big) + \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 \\ \le 2\alpha^{(0)} k\big(f(x^{(k-1)}) - f^\star\big) + \|x^{(k)} + p^{(k)} - x^\star\|^2. \end{aligned}$$

Summing this inequality over $k = 0, \dots, T$ gives

$$2\alpha^{(0)}(T+1)\big(f(x^{(T)}) - f^\star\big) + \|x^{(T+1)} + p^{(T+1)} - x^\star\|^2 \le \|x^{(0)} - x^\star\|^2.$$

The proof is complete. $\square$

To illustrate our results, we evaluate the gradient method and the two variations of the Heavy-ball method on a numerical example. In this example, the objective function is the Moreau proximal envelope of the function $f(x) = (1/c)\|x\|$:

$$f(x) = \begin{cases} \dfrac{1}{c}\|x\| - \dfrac{1}{2c^2} & \|x\| \ge \dfrac{1}{c}, \\ \dfrac{1}{2}\|x\|^2 & \|x\| \le \dfrac{1}{c}, \end{cases} \tag{3.11}$$

with $c = 5$ and $x \in \mathbb{R}^{50}$. One can verify that $f(x) \in \mathcal{F}_L^{1,1}$, i.e., it is convex and continuously differentiable with Lipschitz constant $L = 1$ [84]. First-order methods designed to find the minimum of this cost function are expected to pertain very poor convergence behavior [74]. For the Heavy-ball algorithm with constant step-sizes (2.9) we chose $\beta = 0.5$ and $\alpha = 1/L$, for the variant with time varying step-sizes (3.9) we used $\alpha^{(0)} = 1/L$ whereas the gradient algorithm was implemented with the step-size $\alpha = 1/L$. Figure 3.1 shows the progress of the objective values towards the optimal solution. The plot shows that both Heavy-ball iterates outperform the gradient algorithm and suggests that $\mathcal{O}(1/k)$ is a quite accurate convergence rate estimate for the Heavy-ball and the gradient method.

3.3 Convergence of Nesterov's method with constant step-sizes for the class $\mathcal{F}_L^{1,1}$

For objective functions $f \in \mathcal{S}_{\mu,L}^{1,1}$, it is possible to use constant step-sizes in Nesterov's method and still guarantee a linear rate of convergence [50]. For the objective functions on the class $\mathcal{F}_L^{1,1}$, however, to the best of our knowledge no convergence result exists for Nesterov's method with fixed step-sizes. Using a similar analysis as in the previous section, we can derive the following convergence rate bound.

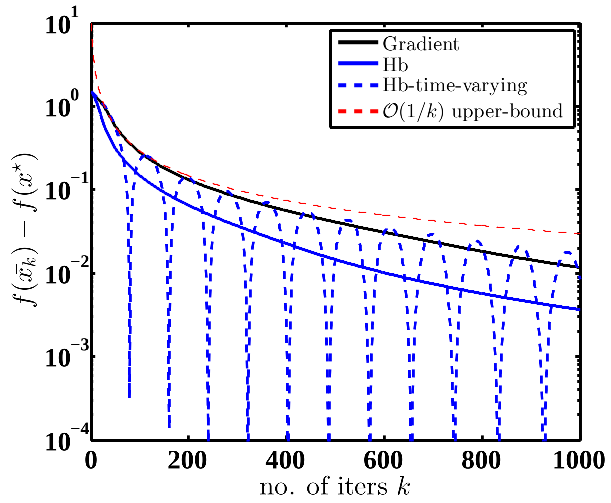


Figure 3.1: Comparison of the progress of the objective values evaluated at the Cesáro average of the iterates of the gradient descent and Heavy-ball methods, and of the primal variable itself for Heavy-ball iterates with time-varying step-sizes. Included for reference is also an $\mathcal{O}(1/k)$ upper bound.

Theorem 3.3

Assume that $f \in \mathcal{F}_L^{1,1}$ and that $\beta \in [0,1)$. Then the sequence $\{x^{(k)}\}$ generated by Nesterov's iteration (2.11) satisfies

$$f(\overline{x}^{(T)}) - f^\star \leq \frac{1}{T+1}\left(\frac{\beta}{1-\beta}(f(x^{(0)}) - f^\star) + \frac{L(1-\beta)}{2}\|x^{(0)} - x^\star\|^2\right) \qquad (3.12)$$

where $\overline{x}^{(T)}$ is the Cesáro average of the iterates:

$$\overline{x}^{(T)} \triangleq \frac{1}{T+1}\sum_{k=0}^{T} x^{(k)}.$$

Proof. Assume that $\beta \in [0,1)$, and let

$$p^{(k)} = \frac{\beta}{1-\beta}\left(x^{(k)} - x^{(k-1)} + \frac{1}{L}\nabla f(x^{(k-1)})\right), \quad k \in \mathbb{N}_0. \qquad (3.13)$$

Considering (2.11) and substituting the y-th iterates in the x-th iterates

$$\begin{aligned} x^{(k+1)} + p^{(k+1)} &= \frac{1}{1-\beta}x^{(k+1)} + \frac{\beta}{1-\beta}(\frac{1}{L}\nabla f(x^{(k)}) - x^{(k)}) \\ &\overset{(2.11)}{=} x^{(k)} + p^{(k)} - \frac{1}{L(1-\beta)}\nabla f(x^{(k)}), \end{aligned}$$

which implies that

$$\begin{aligned}
\|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 =& \|x^{(k)} + p^{(k)} - x^\star\|^2 + \frac{1}{L^2(1-\beta)^2}\|\nabla f(x^{(k)})\|^2 \\
&- \frac{2}{L(1-\beta)}\langle x^{(k)} + p^{(k)} - x^\star, \nabla f(x^{(k)})\rangle \\
\overset{(3.13)}{=}& \|x^{(k)} + p^{(k)} - x^\star\|^2 - \frac{2}{L(1-\beta)}\langle x^{(k)} - x^\star, \nabla f(x^{(k)})\rangle \\
&- \frac{2\beta}{L^2(1-\beta)^2}\langle \nabla f(x^{(k-1)}),\ \nabla f(x^{(k)})\rangle + \frac{1}{L^2(1-\beta)^2}\|\nabla f(x^{(k)})\|^2 \\
&- \frac{2\beta}{L(1-\beta)^2}\langle x^{(k)} - x^{(k-1)}, \nabla f(x^{(k)})\rangle \\
\overset{(3.5)}{\leq}& \|x^{(k)} + p^{(k)} - x^\star\|^2 - \frac{2}{L(1-\beta)}(f(x^{(k)}) - f^\star) \\
&- \frac{1}{L^2(1-\beta)}\|\nabla f(x^{(k)})\|^2 - \frac{2\beta}{L(1-\beta)^2}(f(x^{(k)}) - f(x^{(k-1)})) \\
&- \frac{\beta}{L^2(1-\beta)^2}\|\nabla f(x^{(k)}) - \nabla f(x^{(k-1)})\|^2 \\
&- \frac{2\beta}{L^2(1-\beta)^2}\langle \nabla f(x^{(k-1)}),\ \nabla f(x^{(k)})\rangle \\
&+ \frac{1}{L^2(1-\beta)^2}\|\nabla f(x^{(k)})\|^2.
\end{aligned}$$

After rearrangement of terms, we thus have

$$\begin{aligned}
\frac{2}{L(1-\beta)^2}(f(x^{(k)}) - f^\star) + \|x^{(k+1)} + p^{(k+1)} - x^\star\|^2 \leq& \frac{2\beta}{L(1-\beta)^2}(f(x^{(k-1)}) - f^\star) \\
&+ \|x^{(k)} + p^{(k)} - x^\star\|^2 - \frac{\beta}{L^2(1-\beta)^2}\|\nabla f(x^{(k-1)})\|^2.
\end{aligned} \tag{3.14}$$

Multiplying the sides of (3.14) in $L/2$ and summing over $k = 0, \ldots, T$ gives

$$\begin{aligned}
\frac{1}{1-\beta}\sum_{k=0}^{T}\big(f(x^{(k)}) - f^\star\big) + \sum_{k=0}^{T}\left(\frac{\beta}{(1-\beta)^2}\big(f(x^{(k)}) - f^\star\big) + \frac{L}{2}\|x^{(k+1)} + p^{(k+1)} - x^\star\|^2\right) \\
\leq \sum_{k=0}^{T}\left(\frac{\beta}{(1-\beta)^2}\big(f(x^{(k-1)}) - f^\star\big) + \frac{L}{2}\|x^{(k)} + p^{(k)} - x^\star\|^2\right),
\end{aligned}$$

which implies that

$$\frac{1}{1-\beta}\sum_{k=0}^{T}\big(f(x^{(k)}) - f^\star\big) \leq \frac{\beta}{(1-\beta)^2}\big(f(x^{(0)}) - f^\star\big) + \frac{L}{2}\|x^{(0)} - x^\star\|^2.$$

Using the convexity inequality (3.7) concludes the proof. □

Recently, Allen-Zou and Orrechia [85] demonstrated that another fast gradient method due to Nesterov [34] converges with constant step-sizes for all $f \in \mathcal{F}_L^{1,1}$. That method generates iterates in the following manner

$$\begin{aligned} y^{(k+1)} &= x^{(k)} - \frac{1}{L}\nabla f(x^{(k)}), \\ z^{(k+1)} &= \underset{z\in\mathbb{R}^n}{\arg\min}\{V_x(z) + \langle \alpha\nabla f(x^{(k)}),\, z - z^{(k)}\rangle\}, \\ x^{(k+1)} &= \tau z^{(k+1)} + (1-\tau)y^{(k+1)}, \end{aligned} \tag{3.15}$$

where $\alpha \in \mathbb{R}_+$, $\tau \in [0,1]$, and $V_x(\cdot)$ is the Bergman divergence function [85]. Similar to Theorem 3.3, it has been shown in [85] that the Cesáro average of the iterates generated by (3.15) converges to the optimum at a rate of $\mathcal{O}(1/k)$. Note that while both iterations (2.11) and (3.15) enjoy the same global rate of convergence, the two schemes are remarkably different computationally. In particular, (3.15) requires two gradient computations per iteration, as opposed to one gradient computation needed in (2.11).

3.4 Global analysis of Heavy-ball algorithm for the class $\mathcal{S}_{\mu,L}^{1,1}$

In this section, we focus on objective functions in the class $\mathcal{S}_{\mu,L}^{1,1}$and derive a global linear rate of convergence for the Heavy-ball algorithm. In our convergence analysis, we will use the following simple lemma on convergence of sequences.

Lemma 3.1
Let $\{A^{(k)}\}_{k\geq 0}$ and $\{B^{(k)}\}_{k\geq 0}$ be nonnegative sequences of real numbers satisfying

$$A^{(k+1)} + bB^{(k+1)} \leq a_1A^{(k)} + a_2A^{(k-1)} + cB^{(k)}, \quad k = 0, 1 \ldots, \tag{3.16}$$

with constants $a_1, a_2, b \in \mathbb{R}_+$ and $c \in \mathbb{R}$. Moreover, assume

$$A^{(-1)} = A^{(0)}, \quad a_1 + a_2 < 1, \quad c < b.$$

Then

$$A^{(k)} \leq q^k((q - a_1 + 1)A^{(0)} + cB^{(0)}), \quad q = \max\left\{\frac{c}{b}, \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}\right\} \in [0, 1). \tag{3.17}$$

Proof. It is easy to check that (3.17) holds for $k = 0$. Let $\gamma \geq 0$ and $k = t$. From (3.16) we have

$$\begin{aligned} A^{(t+1)} + \gamma A^{(t)} + bB^{(t+1)} &\leq (a_1+\gamma)A^{(t)} + a_2A^{(t-1)} + cB^{(t)} \\ &= (a_1+\gamma)(A^{(t)} + \frac{a_2}{a_1+\gamma}A^{(t-1)} + \frac{c}{a_1+\gamma}B^{(t)}) \\ &\leq (a_1+\gamma)(A^{(t)} + \gamma A^{(t-1)} + bB^{(t)}), \end{aligned} \tag{3.18}$$

where the last inequality holds if and only if

$$\frac{a_2}{a_1+\gamma} \leq \gamma, \quad \frac{c}{a_1+\gamma} \leq b. \tag{3.19}$$

The first term in (3.19) along with $\gamma \geq 0$ is equivalent to have $\gamma \geq (-a_1 + \sqrt{a_1^2 + 4a_2})/2$. The second condition in (3.19) can be rewritten as $\gamma \geq c/b - a_1$. Thus, (3.19) holds if

$$\gamma = \max\left\{ \frac{-a_1 + \sqrt{a_1^2 + 4a_2}}{2}, \frac{c}{b} - a_1, 0 \right\}. \tag{3.20}$$

Denoting $q \triangleq a_1 + \gamma < 1$, it follows from (3.18) that

$$A^{(t+1)} + \gamma A^{(t)} + bB^{(t+1)} \leq q(A^{(t)} + \gamma A^{(t-1)} + cB^{(t)}) \leq \cdots \leq q^{t+1}((1+\gamma)A_0 + cB_0).$$

Since $A^{(t)}$ and $B^{(t+1)}$ are nonnegative, (3.17) holds. The proof is complete.

$\square$

We are now ready for the main result in this section.

Theorem 3.4

Assume that $f \in \mathcal{S}^{1,1}_{\mu,L}$ and that

$$\alpha \in (0, \frac{2}{L}), \quad 0 \leq \beta < \frac{1}{2}\left(\frac{\mu\alpha}{2} + \sqrt{\frac{\mu^2\alpha^2}{4} + 4(1 - \frac{\alpha L}{2})} \right). \tag{3.21}$$

Then, the Heavy-ball method (2.9) converges linearly to a unique optimizer $x^\star$. In particular,

$$f(x^{(k)}) - f^\star \leq q^k (f(x^{(0)}) - f^\star). \tag{3.22}$$

where $q \in [0, 1)$.

Proof. Consider the following identity that holds for the heavy-ball iterates (2.9)

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\|^2 =& \alpha^2 \|\nabla f(x^{(k)})\|^2 + \beta^2 \|x^{(k)} - x^{(k-1)}\|^2 \\ & - 2\alpha\beta \langle \nabla f(x^{(k)}), x^{(k)} - x^{(k-1)} \rangle. \end{aligned} \tag{3.23}$$

Since f also belongs to $\mathcal{F}^{1,1}_L$, by [50, Theorem 2.1.5] and (2.9) we have

$$\begin{aligned} f(x^{(k+1)}) - f^\star \leq & f(x^{(k)}) - f^\star - \alpha(1 - \frac{\alpha L}{2}) \|\nabla f(x^{(k)})\|^2 + \frac{L\beta^2}{2} \|x^{(k)} - x^{(k-1)}\|^2 \\ & + \beta(1 - \alpha L) \langle \nabla f(x^{(k)}), x^{(k)} - x^{(k-1)} \rangle \end{aligned} \tag{3.24}$$

Let $\theta \in (0,1)$, multiply both sides of (3.23) by $L\theta/(2-2\theta)$, and add the resulting identity to (3.24) to obtain

$$\begin{aligned} f(x^{(k+1)}) - f^\star + \frac{L\theta}{2(1-\theta)}\|x^{(k+1)} - x^{(k)}\|^2 \leq{}& f(x^{(k)}) - f^\star \\ &+ \alpha\left(\frac{L}{2(1-\theta)}\alpha - 1\right)\|\nabla f(x^{(k)})\|^2 + \frac{L\beta^2}{2(1-\theta)}\|x^{(k)} - x^{(k-1)}\|^2 \\ &+ \beta(1 - \frac{\alpha L}{1-\theta})\langle \nabla f(x^{(k)}), x^{(k)} - x^{(k-1)}\rangle \end{aligned} \tag{3.25}$$

Assume that $(1-\theta)/L \leq \alpha < 2(1-\theta)/L$. Then, since $f \in \mathcal{S}^{1,1}_{\mu,L}$, by [50, Theorem 2.1.10] and Definition 2.3, we have

$$\begin{aligned} f(x^{(k+1)}) - f^\star + \frac{L\theta}{2(1-\theta)}\|x^{(k+1)} - x^{(k)}\|^2 & \\ \leq{}& f(x^{(k)}) - f^\star + 2\alpha\mu\left(\frac{L}{2(1-\theta)}\alpha - 1\right)(f(x^{(k)}) - f^\star) \\ &+ \frac{L\beta^2}{2(1-\theta)}\|x^{(k)} - x^{(k-1)}\|^2 \\ &+ \beta(1 - \frac{\alpha L}{1-\theta})(f(x^{(k)}) - f(x^{(k-1)})) \\ &+ \frac{\beta\mu}{2}(1 - \frac{\alpha L}{1-\theta})\|x^{(k)} - x^{(k-1)}\|^2. \end{aligned}$$

Collecting terms yields

$$\begin{aligned} f(x^{(k+1)}) - f^\star + b\|x^{(k+1)} - x^{(k)}\|^2 \leq{}& a_1(f(x^{(k)}) - f^\star) \\ &+ a_2(f(x^{(k-1)}) - f^\star) + c\|x^{(k)} - x^{(k-1)}\|^2, \end{aligned} \tag{3.26}$$

with

$$\begin{aligned} a_1 &\triangleq 1 - 2\alpha\mu(1 - \frac{\alpha L}{2(1-\theta)}) - \beta(\frac{\alpha L}{1-\theta} - 1), \quad a_2 \triangleq \beta(\frac{\alpha L}{1-\theta} - 1), \\ b &\triangleq \frac{L\theta}{2(1-\theta)}, \quad c \triangleq \frac{\beta}{2}\left(\mu(1 - \frac{\alpha L}{1-\theta}) + \frac{L\beta}{1-\theta}\right), \end{aligned}$$

which is on the form of Lemma 3.1 if we identify $A^{(k)}$ with $f(x^{(k)}) - f^\star$ and $B^{(k)}$ with $\|x^{(k)} - x^\star\|^2$. It is easy to verify that if $\theta \in (0,1)$ and $(1-\theta)/L \leq \alpha < 2(1-\theta)/L$ one has

$$b > 0, \quad a_1 + a_2 < 1.$$

Moreover, provided that

$$0 \leq \beta < \frac{1}{2}\left(\frac{\mu}{L}(\alpha L + \theta - 1) + \sqrt{\frac{\mu^2}{L^2}(\alpha L + \theta - 1)^2 + 4\theta}\right),$$

it holds that $c < b$ and consequently one can apply Lemma 3.1 with constants a_1, a_2, b, and c to conclude the linear convergence (3.22). Defining $\lambda \triangleq 1 - \theta$ the stability criteria reads

$$\lambda \in (0,1), \quad \frac{\lambda}{L} \leq \alpha < \frac{2\lambda}{L}, \quad 0 \leq \beta < \frac{1}{2}\left(\frac{\mu}{L}(\alpha L - \lambda) + \sqrt{\frac{\mu^2}{L^2}(\alpha L - \lambda)^2 + 4(1-\lambda)}\right).$$

The first two conditions can be rewritten as

$$\alpha \in (0, \frac{2}{L}), \quad \lambda \in \left(\frac{\alpha L}{2}, \min(\alpha L, 1)\right).$$

Substituting $\lambda = \alpha L/2$ in the upper stability bound on β completes the proof. $\square$

This result extends earlier theoretical results for $\mathcal{S}^{2,1}_{L,\mu}$ to $\mathcal{S}^{1,1}_{\mu,L}$ and demonstrates that the Heavy-ball method has the same rate of convergence as the gradient method and Nesterov's fast gradient method for this class of objective functions. A few comments regarding our stability criteria (3.21) are in order.

First, we observe that (3.21) guarantees stability for a wider range of parameters than the stability criteria (3.1) for $f \in \mathcal{F}^{1,1}_L$, and wider ranges of parameters than the stability analysis of the Heavy-ball method for non-convex cost functions presented in [82]. In particular, when α tends to $2\lambda/L$, our stability criterion allows β to be as large as μ/L, whereas the stability condition (3.1) requires that β tends to zero when α reaches $2/L$; see Figure 3.2.

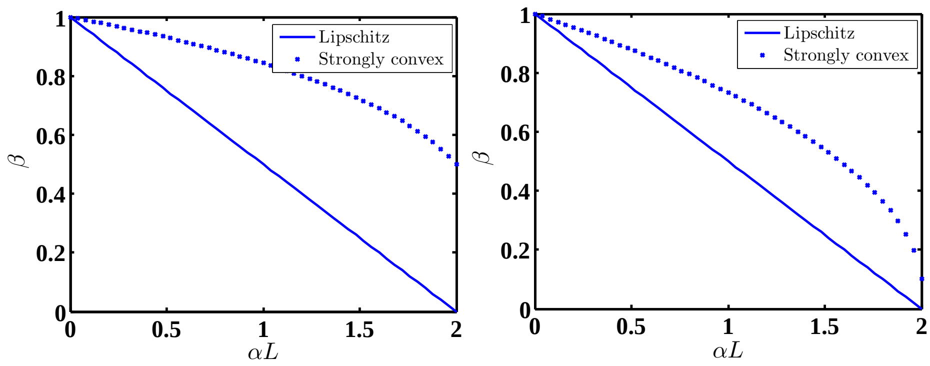


Figure 3.2: The set of parameters (α, β) which guarantee convergence of the Heavy-ball algorithm for $f \in \mathcal{F}^{1,1}_L$(Theorem. 3.1) and $f \in \mathcal{S}^{1,1}_{\mu,L}$(Theorem. 3.4). The left figure uses $L = 2$, $\mu = 1$ and in the right figure $L = 10$, $\mu = 1$.

Second, by comparing (3.21) with α and β that guarantee stability for $f \in \mathcal{S}^{1,1}_{\mu,L}$ [52]:

$$\beta \in [0, 1), \quad \alpha \in \left(0, \frac{2(1+\beta)}{L}\right), \tag{3.27}$$

our stability criteria may appear restrictive at first. However, motivated by [83], we consider a counter example where the local stability criteria (3.27) for twice differentiable strongly

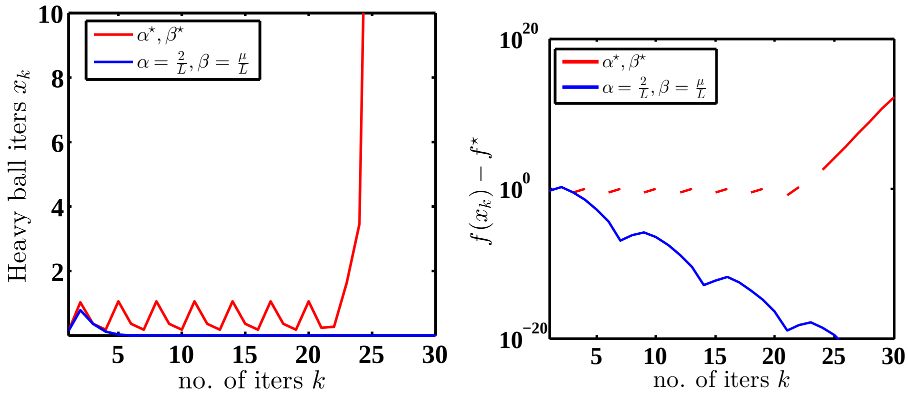


Figure 3.3: Heavy-ball iterates with optimal step-sizes for $f \in \mathcal{S}_{\mu,L}^{2,1}$ do not converge for the example in (3.28). However, parameters that satisfy our new global stability criteria ensure convergence of the iterates.

convex functions do not hold globally for the class $\mathcal{S}_{\mu,L}^{1,1}$. In particular, let us consider

$$\nabla f(x) = \begin{cases} 50x + 45 & x < -1, \\ 5x & -1 \le x < 0, \\ 50x & x \ge 0. \end{cases} \tag{3.28}$$

It is easy to check that ∇f is continuous and $f \in \mathcal{S}_{\mu,L}^{1,1}$with $\mu = 1$ and $L = 50$. According to our numerical tests, for an initial condition in the interval $x^{(0)} < -0.8$ or $x^{(0)} > 0.15$, the Heavy-ball method with optimal parameters $\alpha^\star = 4/(\sqrt{L} + \sqrt{\mu})^2$ and $\beta^\star = (\sqrt{L} - \sqrt{\mu})^2/(\sqrt{L} + \sqrt{\mu})^2$ produces non-converging sequences. However, Figure 3.3 shows that using the maximum value of α permitted by our global analysis results in iterates that converge to the optimum.

Finally, note that Lemma 3.1 also provides an estimate of the convergence factor of the iterates. In particular, after a few simplifications one can find that for

$$\alpha \in (0, \frac{1}{L}], \quad \beta = \sqrt{(1 - \alpha\mu)(1 - \alpha L)},$$

and $\theta = 1 - \alpha L$ in (3.26), the convergence factor of the Heavy-ball method (2.9) is given by $q = 1 - \alpha\mu$. Note that this factor coincides with the best known convergence factor for the gradient method on $\mathcal{S}_{\mu,L}^{1,1}$ [52, Theorem 2, Chapter 1].

Here, we illustrate an instance of problems in the class $\mathcal{S}_{\mu,L}^{1,1}$and compare the performance of first-order methods. The objective function is the regularized logistic regression

$$f(x) = \log(1 + \exp(-a^\top x)) + (1/2)a^\top x + (\mu/2)\|x\|^2,$$

with random parameters $a \in \mathbb{R}^{100}$ and $\mu \in \mathbb{R}_+$. The problem is widely used in machine learning applications [75]. Figure 3.4 compares the per-iterate progress of the gradient descent, Nesterov's and Heavy-ball algorithms. For the gradient descent method, we set the optimal step-size $\alpha = 2/(L+\mu)$; for the Nesterov's algorithm we set $\alpha = 1/L$ and $\beta = (\sqrt{L}-\sqrt{\mu})/(\sqrt{L}+\sqrt{\mu})$ according to [50], and finally for the Heavy-ball method we picked $\alpha = 2/L$ and $\beta = \mu/L$.

Supported by the numerical simulations we envisage that the convergence factor could be strengthened even further. This is indeed left as a future work.

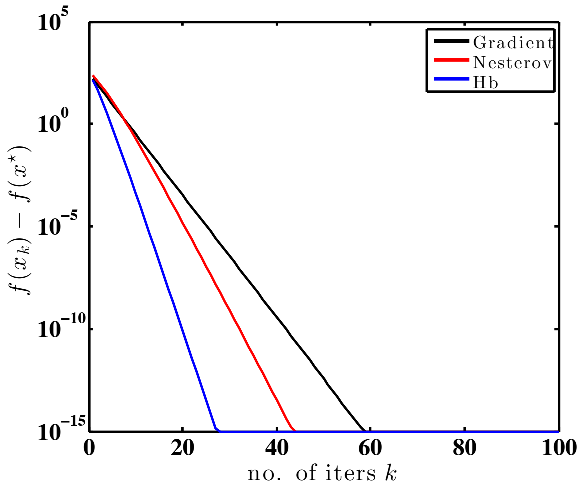


Figure 3.4: Comparison of the progress of the objective values for the gradient descent, Nesterov's and Heavy-ball methods. For this particular example, both Nesterov's and Heavy-ball algorithms out-perform the gradient method.

3.5 Summary

Global stability of the first-order methods have been established for two important classes of convex optimization problems. Specifically, we have shown that when the objective function is convex and has a Lipschitz-continuous gradient, then the Cesáro-averages of the iterates generated by the Heavy-ball and Nesterov's methods with constant step-sizes converge to an optimum at a rate no slower than $\mathcal{O}(1/k)$, where k is the number of iterations. When the objective function is also strongly convex, we established that the Heavy-ball iterates converge linearly to the unique optimum. Numerical examples confirmed the theoretical findings.

Chapter 4
Multi-step methods for network optimization

DISTRIBUTED first-order methods are investigated in this chapter. First, we develop a multi-step weighted gradient method that maintains a network-wide constraint on the decision variables throughout the iterations. The accelerated algorithm is based on the Heavy-ball method extended to the networked setting. We derive optimal algorithm parameters, show that the method has linear convergence rate, and quantify the improvement in convergence factor over the gradient method. Our analysis shows that the method is particularly advantageous when the eigenvalues of the Hessian of the objective function and/or the eigenvalues of the graph Laplacian of the underlying network have a large spread. Second, we investigate how similar techniques can be used to accelerate dual decomposition across a network of decision-makers. In particular, given smoothness parameters of the objective function, we present closed-form expressions for the optimal parameters of an accelerated gradient method for the dual. Third, we quantify how the convergence properties of the algorithm are affected when the algorithm is tuned using misestimated problem parameters. This robustness analysis shows that the accelerated algorithm endures parameter violations well and in most cases outperforms its non-accelerated counterpart. Finally, we apply the developed algorithms to three case studies: networked resource allocation, distributed averaging, and Internet congestion control. In each application we demonstrate superior performance compared to alternatives from the literature.

The chapter is organized as follows. In Section 4.1, we review the related literature to distributed first-order methods. The problem formulation along with our assumptions are presented in Section 4.2. Section 4.3 proposes a multi-step weighted gradient algorithm, establishes conditions for its convergence, and derives optimal step-size parameters. Section 4.4 develops a technique for accelerating the dual problem based on parameters for the (smooth) primal. Section 4.5 presents a robustness analysis of the multi-step algorithm in the presence of uncertainty. Section 4.6 applies the proposed techniques to three engineering problems: resource allocation, consensus, and network flow control. Numerical results and performance comparisons are presented for each case study. Section 4.7 summarizes the content of the chapter.

4.1 Related work

Distributed optimization has recently attracted significant attention from several research communities. Examples include the work on network utility maximization for resource allocation in communication networks [64], distributed coordination of multi-agent systems [86], collaborative estimation in wireless sensor networks [87], distributed machine learning [45], and many others. The majority of these works apply gradient or sub-gradient methods to the dual formulation of the decision problem. Although gradient methods are easy to implement and require modest computations, they suffer from slow convergence. In some cases, such as the development of distributed power control algorithms for cellular phones [88], one can replace gradient methods by fixed-point iterations and achieve improved convergence rates. For other problems, such as average consensus [89], a number of heuristic methods have been proposed that improve the convergence time of the standard method [90, 8]. However, we are not interested in tailoring techniques to individual problems; our aim is to develop general-purpose schemes that retain the simplicity of the gradient method but improve convergence times.

Even if the optimization problem is convex and the sub-gradient method is guaranteed to converge to an optimal solution, the rate of convergence is very modest. The convergence rate of the gradient method is improved if the objective function is differentiable with Lipschitz-continuous gradient, and even more so if the function is also strongly convex [50]. When the objective and constraint functions are smooth, several techniques exist that allow for even shorter solution times. One such technique is higher-order methods, such as Newton's method [67], which use both the gradient and the Hessian of the objective function. Although distributed Newton methods have recently been developed for special problem classes (e.g., [91, 92]), they impose large communication overhead for collecting global Hessian information. Another way to obtain faster convergence is to use multi-step methods [52, 67] that have been introduced in the previous chapters. Recall that these methods rely only on gradient information but use a history of the past iterates when computing the future ones. We explore the latter approach for distributed optimization.

4.2 Assumptions and problem formulation

This chapter is concerned with collaborative optimization by a network of decision-makers of the form (2.20), which is restated here for convenience

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & Ax = b, \end{array} \tag{4.1}$$

where x, A, and b are of dimensions $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, and $\mathbb{R}^m$, respectively. Recall from Section 2.4 that we termed (4.1) as networked optimization problem and that the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the physical information exchange between the decision-makers.

The formulation (4.1) is more general that it might first appear. For example, problems with coupled objectives can be put of the form (4.1) by first introducing new variables, which act as local copies of the shared variable, and then constraining these copies to be equal (see, e.g., [66] and Appendix 4.A). We will use this technique in Section 4.6.2. To use such modeling techniques in their fullest generality (e.g., to allow cost functions to be coupled through several distinct decision variables), one would need to allow the cost functions to be multivariable. While the results in this chapter are generalizable to multivariable cost functions, we have chosen to present the scalar case for ease of notation.

Most acceleration techniques in the literature (e.g., [50, 93, 94]) require that the cost functions are smooth and convex. Similarly, we will make the following assumptions:

Assumption 4.1 Each cost function f_i in (4.1) is convex and twice continuously differentiable with

$$l_i \leq \nabla^2 f_i(x_i) \leq u_i, \quad \forall i \in \mathcal{V}, \tag{4.2}$$

for some positive real constants l_i and u_i such that $0 < l_i \leq u_i$.

Some remarks are in order. Let

$$\mu \triangleq \min_{i \in \mathcal{V}} \; l_i, \quad L \triangleq \max_{i \in \mathcal{V}} \; u_i, \tag{4.3}$$

and define $f(x) \triangleq \sum_{i \in \mathcal{V}} f_i(x_i)$. Then, Assumption 4.1 ensures that $f(x) \in \mathcal{S}^{1,1}_{\mu,L}$, i.e., it is strongly convex with modulus μ and its gradient is Lipschitz-continuous with constant μ. Moreover, the Hessian of f satisfies

$$\mu I \preceq \nabla^2 f(x) \preceq LI, \quad \forall x. \tag{4.4}$$

See, *e.g*, [50, Lemma 1.2.2 and Theorem 2.1.11] for details. Furthermore, Assumption 4.1 guarantees that (4.1) is a convex optimization problem whose unique optimizer $x^\star$ satisfies

$$Ax^\star = b, \qquad \nabla f(x^\star) = A^\top y^\star, \tag{4.5}$$

where $y^\star \in \mathbb{R}^m$ is the vector of optimal Lagrange multipliers for the linear constraints.

4.3 A multi-step weighted gradient method

In the absence of constraints, (4.1) is trivial to solve since the objective function is separable and each decision-maker could simply minimize its cost independently of the others. Hence, it is the existence of constraints that makes (4.1) challenging. Recall from Chapter 2 that in the optimization literature, there are essentially two ways to deal with constraints: the primal and the dual decomposition. In the primal decomposition, one projects the iterates onto the constraint set to maintain feasibility at all iterations; such a method will be developed in this section. In the dual decomposition, on the other hand, we eliminate couplings between decision-makers and solve the associated dual problem; we will consider such techniques in Section 4.4.

Computing the Euclidean projection onto the constraint of (4.1) typically requires the full decision vector x, which is not available to the decision-makers in our setting. An alternative, explored e.g., in [63], is to consider weighted gradient methods which use a linear combination of the information available to nodes to ensure that iterates remain feasible. For our problem (4.1), the weighted gradient method takes the form

$$x^{(k+1)} = x^{(k)} - \alpha W \nabla f(x^{(k)}). \tag{4.6}$$

Here, $W \in \mathbb{R}^{n \times n}$ is a weight matrix that should satisfy the following three conditions: (i) the locality of information exchange between the decision makers should be preserved; (ii) provided that the initial point $x^{(0)}$ is feasible, the iterates generated by (4.6) should remain feasible; and (iii), the fixed-points of (4.6) should satisfy the optimality conditions (4.5).

To ensure condition (i), W should have the same sparsity pattern as the information graph $\mathcal{G}$, i.e., $W_{ij} = 0$ if $i \neq j$ and $\{i, j\} \notin \mathcal{E}$. In this way, the iterations (4.6) read

$$x_i{}^{(k+1)} = x_i{}^{(k)} - \alpha \sum_{j \in i \cup \mathcal{N}_i} W_{ij} \nabla f_j(x_j^{(k)}),$$

and can be executed by individual decision-makers based on the information that they have access to. Conditions (ii) and (iii) translate into the following requirements (see [63] for details)

$$AW = \mathbf{0}, \qquad\qquad WA^\top = \mathbf{0}. \tag{4.7}$$

The next example describes one particular problem instance.

Example 4.1 When the decision-makers are only constrained by a global resource budget, (4.1) reduces to

$$\begin{array}{ll} \text{minimize} & \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{subject to} & \sum_{i \in \mathcal{V}} x_i = x_{\text{tot}}. \end{array}$$

A distributed algorithm for this problem was developed in [62] and interpreted as a weighted gradient method in [63]. In our notation, $A = \mathbf{1}^\top$ and $b = x_{\text{tot}}$ so in addition to the sparsity pattern, W should also satisfy $\mathbf{1}^\top W = \mathbf{0}^\top$ and $W\mathbf{1} = \mathbf{0}$. In [63], it was shown that the symmetric W that satisfies these constraints and guarantees the smallest convergence factor of the weighted gradient iterations can be found by solving a convex optimization problem. In addition, [63] proposed several heuristics for constructing W in a distributed manner.

A few comments are in order. First, not all constraint matrices A admit a weight matrix W that satisfies the above constraints, hence not all problems of the form (4.1) are amendable to a distributed solution using a weighted gradient method. Second, to find a feasible initial point $x^{(0)}$, one typically needs to find a solution for the linear system of equations $Ax = b$ in a distributed way. This generally requires a separate distributed mechanism; see, e.g., [62] and [95, Chapter 2].

4.3.1 A multi-step weighted gradient method and its convergence

Consider the following multi-step weighted gradient iteration

$$x^{(k+1)} = x^{(k)} - \alpha W \nabla f(x^{(k)}) + \beta \left(x^{(k)} - x^{(k-1)} \right). \tag{4.8}$$

Under the sparsity constraint on W detailed above, these iterations can be implemented by individual decision-makers. Moreover, (4.7) ensures that if $Ax^{(0)} = Ax^{(1)} = b$ then every iterate produced by (4.8) will also satisfy the linear constraints. The next theorem characterizes the convergence of the iterations (4.8) and derives optimal step-size parameters α and β.

Theorem 4.1
Consider the optimization problem (4.1) under Assumption 4.1, and let $x^\star$ denote its unique optimizer. Assume that W has $m < n$ eigenvalues at 0 and satisfies $AW = \mathbf{0}$ and $WA^\top = \mathbf{0}$. Let $H = \nabla^2 f(x^\star)$ and $\lambda_1(WH) \le \lambda_2(WH) \le \cdots \le \lambda_n(WH)$ be the (ordered) eigenvalues of WH so that $\underline{\lambda} = \lambda_{m+1}(WH)$ is the smallest non-zero eigenvalue of WH and $\overline{\lambda} = \lambda_n(WH)$ is the largest. Then, if

$$0 \le \beta < 1, \qquad\qquad 0 < \alpha < \frac{2}{L}\frac{(1+\beta)}{\lambda_n(W)},$$

the iterates (4.8) converge to $x^\star$ at linear rate

$$\|x^{(k+1)} - x^\star\| \le q\|x^{(k)} - x^\star\| \qquad \forall k \ge 0,$$

with $q = \max\left\{\sqrt{\beta}, |1+\beta-\alpha\underline{\lambda}| - \sqrt{\beta}, |1+\beta-\alpha\overline{\lambda}| - \sqrt{\beta}\right\}$. Moreover, the minimal value of q is

$$q^\star = \frac{\sqrt{\overline{\lambda}} - \sqrt{\underline{\lambda}}}{\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}},$$

obtained for step-sizes $\alpha = \alpha^\star$ and $\beta = \beta^\star$ where

$$\alpha^\star = \left(\frac{2}{\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}} \right)^2, \quad \beta^\star = \left(\frac{\sqrt{\overline{\lambda}} - \sqrt{\underline{\lambda}}}{\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}} \right)^2. \tag{4.9}$$

Proof. See Appendix 4.C for this and all other proofs of this chapter. □

It is interesting to investigate when (4.8) significantly improves over the single-step algorithm. In [63], it is shown that the best convergence factor of the weighted gradient iteration (4.6) is

$$q_0^\star = \frac{\overline{\lambda} - \underline{\lambda}}{\overline{\lambda} + \underline{\lambda}}.$$

One can verify that $q^\star \leq q_0^\star$, i.e., the optimally tuned multi-step method is never slower than the single-step method. Moreover, the improvement in convergence factor depends on the quantity $\kappa = \overline{\lambda}/\underline{\lambda}$: when κ is large, the speed-up is roughly proportional to $\sqrt{\kappa}$. In the networked setting, there are two reasons for a large value of κ. One is simply that the Hessian of the objective function is ill-conditioned, so that the ratio L/μ is large. The other is that the matrix W is ill-conditioned, i.e., that $\lambda_n(W)/\lambda_{m+1}(W)$ is large. As we will see in the examples, the graph Laplacian is often a valid choice for W. Thus, there is a direct connection between the topology of the underlying information graph and the convergence rate (improvements) of the multi-step weighted gradient method. We will discuss this connection in detail in Section 4.6.

In many applications, $H = \nabla^2 f(x^\star)$ is not known, but the bounds such as (4.4) are known. In such cases, the next result can be useful.

Proposition 4.1

Let $\underline{\lambda}_W = \mu\lambda_{m+1}(W)$ and $\overline{\lambda}_W = L\lambda_n(W)$. Then $\underline{\lambda}_W \leq \underline{\lambda}$ and $\overline{\lambda}_W \geq \overline{\lambda}$. Moreover, the step-sizes

$$\alpha = \left(\frac{2}{\sqrt{\overline{\lambda}_W} + \sqrt{\underline{\lambda}_W}} \right)^2, \qquad \beta = \left(\frac{\sqrt{\overline{\lambda}_W} - \sqrt{\underline{\lambda}_W}}{\sqrt{\overline{\lambda}_W} + \sqrt{\underline{\lambda}_W}} \right)^2,$$

ensure the linear convergence of (4.8) with the convergence factor

$$\tilde{q} = \frac{\sqrt{\overline{\lambda}_W} - \sqrt{\underline{\lambda}_W}}{\sqrt{\overline{\lambda}_W} + \sqrt{\underline{\lambda}_W}}.$$

4.3.2 Optimal weight selection for the multi-step method

The results in the previous subsection provide optimal step-size parameters α and β for a given weight matrix W. However, the expressions for the associated convergence factors depend on the eigenvalues of WH and optimizing the entries in W jointly with the step-size parameters can yield even further speed-ups. We make the following observation.

Proposition 4.2

Under the hypotheses of Proposition 4.1,

(i) If H is known, then minimizing the convergence factor $q^\star$ is equivalent to minimizing $\overline{\lambda}/\underline{\lambda}$.

(ii) If H is not known, while μ and L in (4.4) are, then the weight matrix that minimizes $\tilde{q}$ is the one with minimal value of $\overline{\lambda}_W/\underline{\lambda}_W$.

The next result shows how the optimal weight selection for both scenarios can be found via convex optimization.

Proposition 4.3
The weight matrix for (4.8) that minimizes $\overline{\lambda}/\underline{\lambda}$ can be found by solving the convex optimization problem

$$\begin{array}{ll} \underset{W,t}{\text{minimize}} & t \\ \text{subject to} & I_{n-m} \preceq P^\top H^{1/2} W H^{1/2} P \preceq t I_{n-m} \\ & W \in \mathcal{A},\ W \succeq \mathbf{0},\ H^{1/2} W H^{1/2} V = \mathbf{0}, \end{array} \tag{4.10}$$

where $\mathcal{A} \in \mathcal{S}^n$ is the sparsity pattern induced by $\mathcal{G}$, $V = H^{-1/2} A^\top$, and $P \in \mathbb{R}^{n \times n-m}$ is a matrix of orthonormal vectors spanning the null space of $V^\top$.

Remark 4.1 When H is not known but the bounds μ and L in (4.4) are, Proposition 4.2 suggests that one should look for a weight matrix W that minimizes $\overline{\lambda}_W/\underline{\lambda}_W$. This can be done by using the formulation in Proposition 4.3 by setting $H = I$.

Remark 4.2 In addition to the basic conditions that admissible weight matrices have to satisfy, Proposition 4.3 also requires that W be positive semi-definite. Without this additional requirement, (4.10) is non-convex and generically has no tractable solution (see [96], [97]). Designing distributed algorithms for constructing weight matrices that guarantee optimal or close-to-optimal convergence factors is an open question and a subject for future investigations.

4.4 A multi-step dual ascent method

When the constraint matrix A is such that the weight optimization problem (4.10) does not admit a solution, we have no systematic approach to find a weight matrix W that guarantees convergence of the weighted multi-step gradient iterations. An alternative approach for solving (4.1) can then be to use Lagrange relaxation, i.e., to introduce Lagrange multipliers $y \in \mathbb{R}^m$ for the equality constraints and solve the dual problem. The dual function associated with (4.1) is

$$d(y) \triangleq \inf_x \sum_i \left\{ f_i(x_i) + \left(\sum_{r=1}^m y_r A_{ri} \right) x_i \right\} - \sum_{r=1}^m y_r b_r. \tag{4.11}$$

Since the dual function is separable in x, it can be evaluated in parallel. For a given Lagrange multiplier vector y, each decision-maker then needs to compute

$$x_i^\star(y) = \arg\min_z f_i(z) + \left(\sum_{r=1}^m y_r A_{ri} \right) z. \tag{4.12}$$

The dual problem is to maximize $d(y)$ with respect to y, i.e.,

$$\underset{y}{\text{minimize}} \quad -d(y) = f_\star(-A^\top y) + b^\top y,$$

where $f_\star(z) \triangleq \sup_x \; z^\top x - f(x)$ is the conjugate function. Recall that if f is strongly convex then $f_\star$ and hence $-d(\cdot)$ are convex and continuously differentiable [98]. Moreover, $\nabla d(y) = Ax^\star(y) - b$. In light of our earlier discussion, it is natural to attempt to solve the dual problem using a multi-step iteration of the form

$$y_r^{(k+1)} \quad = \quad y_r^{(k)} + \alpha \left(\sum_i A_{ri} x_i^\star(y^{(k)}) - b_r \right) + \beta \left(y_r^{(k)} - y_r^{(k-1)} \right). \tag{4.13}$$

To be able to execute the multi-step dual ascent iterations (4.12) and (4.13) in a distributed manner, decision-maker i needs to be able to collect the Lagrange multipliers y_r for all r such that $A_{ri} \neq 0$, and the decision-maker in charge of updating y_r needs to be able to collect all $x_i^\star(y)$ for all i with $A_{ri} \neq 0$. This is certainly not always possible, but we will give two examples that satisfy these requirements in Section 4.6.

To find the optimal step-sizes and estimate the convergence factors of the iterations, we need to be able to bound the strong convexity modulus of $d(y)$ and the Lipschitz constant of its gradient. The following observation is in order:

Lemma 4.1

Consider the optimization problem (4.1) with associated dual function (4.11). Let f be a continuously differentiable and closed convex function. Then,

(i) If f is strongly convex with modulus μ, then $-\nabla d$ is Lipschitz continuous with constant $\lambda_n(AA^\top)/\mu$.

(ii) If ∇f is Lipschitz continuous with constant L, then $-d$ is strongly convex with modulus $\lambda_1(AA^\top)/L$.

These dual bounds can be used to derive the following result:

Theorem 4.2

For the optimization problem (4.1) under Assumption 4.1, the multi-step dual ascent iterations (4.13) converge to $y^\star$ at linear rate with the guaranteed convergence factor

$$q^\star = \frac{\sqrt{L\lambda_n(AA^\top)} - \sqrt{\mu\lambda_1(AA^\top)}}{\sqrt{L\lambda_n(AA^\top)} + \sqrt{\mu\lambda_1(AA^\top)}},$$

obtained for step-sizes:

$$\alpha^\star = \left(\frac{2}{\sqrt{L\lambda_n(AA^\top)} + \sqrt{\mu\lambda_1(AA^\top)}} \right)^2, \qquad \beta^\star = \left(\frac{\sqrt{L\lambda_n(AA^\top)} - \sqrt{\mu\lambda_1(AA^\top)}}{\sqrt{L\lambda_n(AA^\top)} + \sqrt{\mu\lambda_1(AA^\top)}} \right)^2.$$

The advantage of Theorem 4.2 is that it provides step-size parameters with guaranteed convergence factor using readily available data of the primal problem. How close to optimal these results are depends on how tight the bounds in Lemma 4.1 are. If the bounds are tight, then the step-sizes in Theorem 4.2 are truly optimal. The next example shows that a certain degree of conservatism may be present, even for quadratic problems.

Example 4.2 Consider the quadratic minimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \end{array}$$

where $Q \succ 0$, $A \in \mathbb{R}^{n\times n}$ is full row-ranked, and $b \in \mathbb{R}^n$. This implies that the objective function is strongly convex with modulus $\lambda_1(Q)$ and that its gradient is Lipschitz-continuous with constant $\lambda_n(Q)$. Hence, according to Lemma 4.1, $-d$ is strongly convex with modulus $\lambda_1(AA^\top)/\lambda_n(Q)$ and its gradient is Lipschitz continuous with constant $\lambda_n(AA^\top)/\lambda_1(Q)$. However, direct calculations reveal that

$$d(y) = -\frac{1}{2}y^\top AQ^{-1}A^\top y - y^\top b,$$

from which we see that $-d$ has convexity modulus $\lambda_1(AQ^{-1}A^\top)$ and that its gradient is Lipschitz continuous with constant $\lambda_n(AQ^{-1}A^\top)$. By [99, p. 225], these bounds are tighter than those offered by Lemma 4.1. Specifically, for congruent matrices Q^{-1} and $AQ^{-1}A^\top$ there exist nonnegative real numbers θ_k such that $\lambda_1(AA^\top) \leq \theta_k \leq \lambda_n(AA^\top)$ and $\theta_k\lambda_k(Q^{-1}) = \lambda_k(AQ^{-1}A^\top)$. For $k = 1, n$ we obtain

$$\frac{\lambda_1(AA^\top)}{\lambda_n(Q)} \leq \lambda_1(AQ^{-1}A^\top), \quad \lambda_n(AQ^{-1}A^\top) \leq \frac{\lambda_n(AA^\top)}{\lambda_1(Q)}.$$

For some important classes of problems, the bounds are, however, tight. One such example is the average consensus application considered in Section 4.6.

4.5 Robustness analysis

The proposed multi-step methods have significantly improved convergence factors compared to the gradient iterations, and particularly so when the Hessian of the cost function and/or the graph Laplacian of the network is ill-conditioned. The results of Theorem 4.1 and Proposition 4.1 specify sufficient conditions for the convergence of multi-step iterations in terms of the design parameters α, β, and W. However, these parameters are determined based on upper and lower bounds on the Hessian and the largest and smallest non-zero eigenvalue of W. In many applications, W and H might not be perfectly known, and $\overline{\lambda}$ and $\underline{\lambda}$ have to be estimated based on available data. It is therefore important to analyze the sensitivity of the multi-step methods to errors in these parameters to assess if the performance benefits prevail when the step-sizes are calculated using (slightly) misestimated $\overline{\lambda}$ and $\underline{\lambda}$. Such an analysis will be performed in this section.

Let $\underset{\sim}{\lambda}$ and $\widetilde{\lambda}$ denote the estimates of $\underline{\lambda}$ and $\overline{\lambda}$ available when tuning the step-sizes. We are interested in quantifying how the convergence and the convergence factors of the gradient and the multi-step methods are affected when $\underset{\sim}{\lambda}$ and $\widetilde{\lambda}$ are used in the step-size formulas we derived earlier. Theorem 4.1 provides some useful observations for the multi-step method. The corresponding results for the weighted gradient method are summarized in the following lemma:

Lemma 4.2

Consider the weighted gradient iterations (4.6) and let $\overline{\lambda}$ and $\underline{\lambda}$ denote the largest and smallest non-zero eigenvalue of WH, respectively. Then, for fixed step-size $0 < \alpha < 2/\overline{\lambda}$, (4.6) converges to $x^\star$ at linear rate with convergence factor

$$q_G = \max\left\{|1-\alpha\underline{\lambda}|,\ |1-\alpha\overline{\lambda}|\right\}.$$

Furthermore, the minimal value $q_G^\star = (\overline{\lambda}-\underline{\lambda})/(\overline{\lambda}+\underline{\lambda})$ is obtained for the step-size $\alpha = 2/(\overline{\lambda}+\underline{\lambda})$.

Combining this lemma with our previous results from Theorem 4.1 yields the following observation.

Proposition 4.4

Let $\underset{\sim}{\lambda}$ and $\widetilde{\lambda}$ be estimates of $\underline{\lambda}$ and $\overline{\lambda}$, respectively, and assume that $0 < \underset{\sim}{\lambda} < \widetilde{\lambda}$. Then, for all values of $\underset{\sim}{\lambda}$ and $\widetilde{\lambda}$ such that $\overline{\lambda} < \widetilde{\lambda} + \underset{\sim}{\lambda}$, both the weighted gradient iteration (4.6) with step-size

$$\widetilde{\alpha} = 2/(\widetilde{\lambda}+\underset{\sim}{\lambda}), \tag{4.14}$$

and the multi-step method variant (4.8) with

$$\widetilde{\alpha} = \left(\frac{2}{\sqrt{\widetilde{\lambda}}+\sqrt{\underset{\sim}{\lambda}}}\right)^2,\ \widetilde{\beta} = \left(\frac{\sqrt{\widetilde{\lambda}}-\sqrt{\underset{\sim}{\lambda}}}{\sqrt{\widetilde{\lambda}}+\sqrt{\underset{\sim}{\lambda}}}\right)^2, \tag{4.15}$$

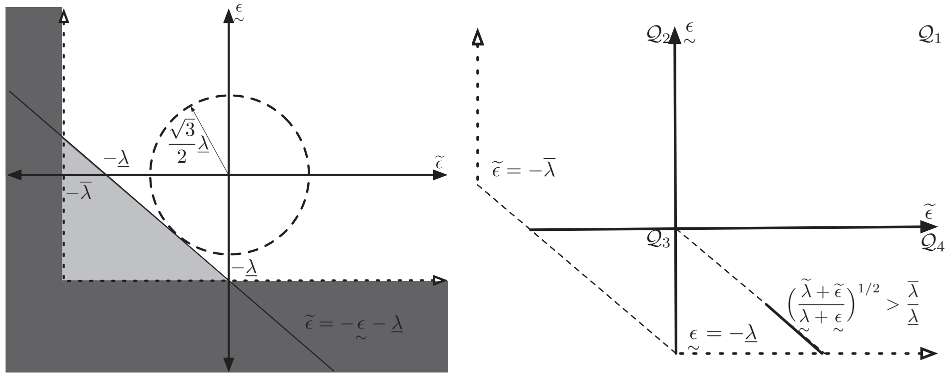
converge to the optimizer $x^\star$ of (4.1).

In practice, one should expect an overestimated $\widetilde{\lambda}$, in which case both methods converge. However, convergence can be guaranteed for a much wider range of perturbations. Figure 4.1(a) considers perturbations of the form $\underset{\sim}{\lambda} = \underline{\lambda} + \underset{\sim}{\varepsilon}$ and $\widetilde{\lambda} = \overline{\lambda} + \widetilde{\varepsilon}$. The white area is the locus of perturbations for which convergence is guaranteed, while the dark area represents inadmissible perturbations which render either $\underset{\sim}{\lambda}$ or $\widetilde{\lambda}$ negative. Note that both algorithms are robust to a continuous departure from the true values of $\underline{\lambda}$ and $\overline{\lambda}$, since there is a ball with radius $\sqrt{3}\underline{\lambda}/2$ around the origin for which both methods are guaranteed to converge. Next, we compare the convergence *factors* of the two methods when the step-sizes are tuned based on inaccurate parameters. The following lemma is then useful.

Lemma 4.3

Let $\underset{\sim}{\lambda}$ and $\widetilde{\lambda}$ satisfy $0 < \overline{\lambda} < \underset{\sim}{\lambda} + \widetilde{\lambda}$. The convergence factor of the weighted gradient method (4.6) with step-size (4.14) is given by

$$\widetilde{q}_G = \begin{cases} 2\overline{\lambda}/(\underset{\sim}{\lambda}+\widetilde{\lambda}) - 1 & \text{if } \underset{\sim}{\lambda}+\widetilde{\lambda} \le \underline{\lambda}+\overline{\lambda} \\ 1 - 2\underline{\lambda}/(\underset{\sim}{\lambda}+\widetilde{\lambda}) & \text{otherwise,} \end{cases} \tag{4.16}$$


$\underset{\sim}{\epsilon}$
$\frac{\sqrt{3}}{2}\underline{\lambda}$
$-\underline{\lambda}$
$-\overline{\lambda}$
$\widetilde{\epsilon}$
$-\underline{\lambda}$
$\widetilde{\epsilon} = -\underset{\sim}{\epsilon} - \underline{\lambda}$
$\mathcal{Q}_2$
$\underset{\sim}{\epsilon}$
$\mathcal{Q}_1$
$\widetilde{\epsilon} = -\overline{\lambda}$
$\widetilde{\epsilon}$
$\mathcal{Q}_3$
$\mathcal{Q}_4$
$\left(\frac{\widetilde{\lambda}+\widetilde{\epsilon}}{\underset{\sim}{\lambda}+\underset{\sim}{\epsilon}}\right)^{1/2} > \frac{\overline{\lambda}}{\underline{\lambda}}$
$\underset{\sim}{\epsilon} = -\underline{\lambda}$

(a) Stability regions.

(b) Different perturbation regions.

Figure 4.1: (a) Perturbations in the white and gray area correspond to the stable and unstable regions of multi-step algorithm respectively. (b) Multi-step algorithm outperforms gradient iterations in $(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{C}\backslash\mathcal{Q}_4$. For symmetric errors in $\mathcal{Q}_4$ (along the line $\underset{\sim}{\varepsilon} = -\widetilde{\varepsilon}$) gradient might outperform multi-step algorithm.

while the multi-step weighted gradient method (4.8) with step-sizes (4.15) has convergence factor

$$\widetilde{q} = \max\left\{\sqrt{\widetilde{\beta}}, |1+\widetilde{\beta}-\widetilde{\alpha}\underline{\lambda}| - \sqrt{\widetilde{\beta}}, |1+\widetilde{\beta}-\widetilde{\alpha}\overline{\lambda}| - \sqrt{\widetilde{\beta}}\right\}. \tag{4.17}$$

The convergence factor expressions derived in Lemma 4.3 allow us to come to the following conclusions:

Proposition 4.5

Let $\underset{\sim}{\lambda} = \underline{\lambda} + \underset{\sim}{\varepsilon}$, $\widetilde{\lambda} = \overline{\lambda} + \widetilde{\varepsilon}$ and define the set of perturbations, $\mathcal{C}$, under which the methods converge

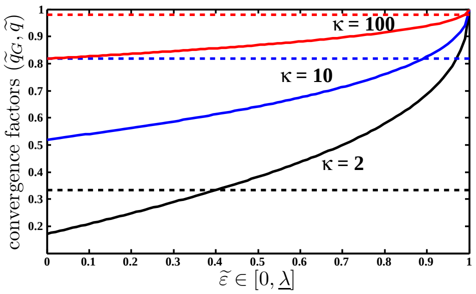
$$\mathcal{C} = \{(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \mid \underset{\sim}{\varepsilon} \geq -\underline{\lambda},\ \widetilde{\varepsilon} \geq -\overline{\lambda},\ \underset{\sim}{\varepsilon} + \widetilde{\varepsilon} \geq -\underline{\lambda}\},$$

and the fourth quadrant in the perturbation space $\mathcal{Q}_4 = \{(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \mid \underset{\sim}{\varepsilon} < 0 \cap \widetilde{\varepsilon} > 0\}$. Then, for all $(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{C}\backslash\mathcal{Q}_4$, we have $\widetilde{q} \leq \widetilde{q}_G$. However, there exists $(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{Q}_4$ for which the scaled gradient has a smaller convergence factor than the multi-step variant. In particular, for

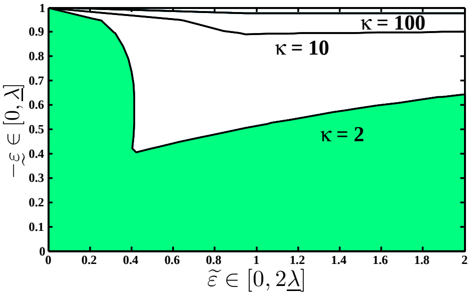
$$(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{Q}_4 \text{ and } (\overline{\lambda} + \widetilde{\varepsilon})/(\underline{\lambda} + \underset{\sim}{\varepsilon}) \geq (\overline{\lambda}/\underline{\lambda})^2, \tag{4.18}$$

the multi-step iterations (4.8) converge slower than (4.6).

Figure 4.1b illustrates the different perturbations considered in Proposition 4.5. While the multi-step method has superior convergence factors for many perturbations, the



(a) Symmetric perturbations in $\mathcal{Q}_4$.



(b) General perturbation in $\mathcal{Q}_4$.

Figure 4.2: (a) Convergence factor of multi-step and gradient algorithms under the condition described by (4.18). Solid lines belong to $\widetilde{q}$ while the dashed lines depict $\widetilde{q}_G$. (b) Level curves of $\widetilde{q} - \widetilde{q}_G$ around the origin for $(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{Q}_4$.

troublesome region $\mathcal{Q}_4$ is probably common in engineering applications since it represents perturbations where the smallest eigenvalue is underestimated while the largest eigenvalue is overestimated. To shed more light on the convergence properties in this region, we perform a numerical study on a quadratic function with $\underline{\lambda} = 1$ and $\overline{\lambda}$ varying from 2 to 100. We first consider symmetric perturbations $\underset{\sim}{\varepsilon} = -\widetilde{\varepsilon}$, in which case the convergence factor of the gradient method is $\widetilde{q}_G = 1 - 2/(1 + \overline{\lambda}/\underline{\lambda})$ while the convergence factor of the multi-step method is $\widetilde{q} = 1 - 2/\sqrt{1 + \widetilde{\lambda}/\underset{\sim}{\lambda}}$. The convergence factor of the gradient iterations is insensitive to symmetric perturbations, while the performance of the multi-step iterations degrades with the size of the perturbation and eventually becomes inferior to the gradient; see Figure 4.2a. To complement this study, we also sweep over $(\underset{\sim}{\varepsilon}, \widetilde{\varepsilon}) \in \mathcal{C} \cap \mathcal{Q}_4$ and compute the convergence factors of the two methods for problems with different $\overline{\lambda}$. Figure 4.2b indicates that when the condition number $\overline{\lambda}/\underline{\lambda}$ increases, the area where the gradient method is superior (the area above the contour line) shrinks. It also shows that when $\underset{\sim}{\lambda}$ tends to zero or $\widetilde{\lambda}$ is very large, the performance of the multi-step method is severely degraded.

4.6 Applications

In this section, we apply the developed techniques to three classes of engineering problems: resource-allocation under network-wide resource constraints, distributed averaging, and Internet congestion control. In all cases, we demonstrate that direct applications of our techniques yield algorithms with significantly faster convergence than state-of-the-art algorithms that have been tailor-made to the specific applications.

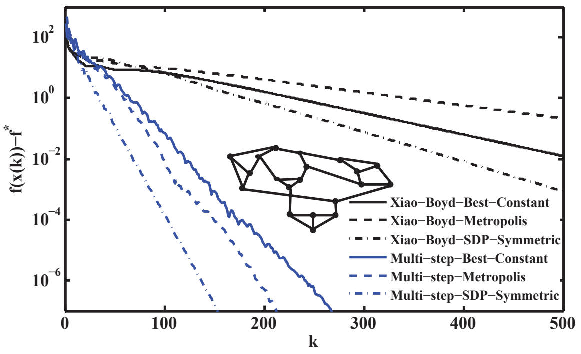


Figure 4.3: Convergence behavior for weighted and multi-step weighted gradient iterations to solve resource allocation problem. The considered network is depicted in the figure. Plot shows $f\left(x^{(k)}\right) - f^\star$ versus iteration number k.

4.6.1 Accelerated resource allocation

Our first application is the distributed resource allocation problem under a network-wide resource constraint [62, 63] discussed in Example 4.1. We compare the multi-step method developed in this chapter with the optimal and suboptimal tuning for the standard weighted gradient iterations proposed in [63]. Similarly to [63], we create problem instances by generating random networks and assigning cost functions of the form $f_i(x_i) = a_i(x_i - c_i)^2 + \log[1+\exp(x_i - d_i)]$ to nodes. The parameters a_i, b_i, c_i and d_i are drawn uniformly from intervals $(0, 2]$, $[-2, 2]$, $[-10, 10]$ and $[-10, 10]$, respectively. In [63], it was shown that the second derivatives of these functions are bounded by $l_i = a_i$ and $u_i = a_i + b_i^2/4$.

Figure 4.3 shows a representative problem instance along with the convergence behavior for weighted and multi-step weighted gradient iterations under several weight choices. The optimal weights for the weighted gradient method, denoted by Xiao-Boyd in the figure, are found by solving a semi-definite program derived in [63], and by Proposition 4.3, with $H = I$, for the multi-step variant. In addition, we evaluate the heuristic weights “best constant” and “metropolis” introduced in [63] (refer to Appendix 4.B for their definitions). In all cases, we observe significantly improved convergence factors for the multi-step method.

4.6.2 Distributed averaging

Our second application is devoted to the distributed averaging. Distributed algorithms for consensus seeking have been researched intensively for decades; see e.g., [89, 100, 101]. Here, each node i in the network initially holds a value c_i and coordinates with neighbors in the graph to find the network-wide average. Clearly, this average can be found by applying any distributed optimization technique to the problem

$$\underset{x}{\text{minimize}} \quad \textstyle\sum_{i\in\mathcal{V}} \frac{1}{2}(x - c_i)^2, \tag{4.19}$$

since the optimal solution to this problem is the network-wide average of the constants c_i. In particular, we will explore how the multi-step technique with our optimal parameter selection rule compares with the state-of-the art distributed averaging algorithms from the literature.

The basic consensus algorithms use iterations of the form

$$x_i{}^{(k+1)} = Q_{ii}x_i{}^{(k)} + \sum_{j\in\mathcal{N}_i} Q_{ij}x_j{}^{(k)}, \tag{4.20}$$

where Q_{ij} are scalar weights, and the node states are initialized by $x_i^{(0)} = c_i$. The paper [47] provides necessary and sufficient conditions on the matrix $Q = [Q_{ij}]$ to ensure that the iterations converge to the network-wide average of the initial values. Specifically, it is required that $\mathbf{1}^\top Q = \mathbf{1}^\top$, $Q\mathbf{1} = \mathbf{1}$, and $r(Q - (1/|\mathcal{V}|)\mathbf{1}\mathbf{1}^\top) < 1$ where $r(\cdot)$ denotes the spectral radius of a matrix. Although the convergence conditions do not require that Q is symmetric, techniques for minimizing the convergence factor often assume Q to be symmetric [47], [8].

Following the steps given in Section 4.4, we now develop a dual approach to derive iterations that solve (4.19). In particular, we can rewrite (4.19) in the form of collaborative minimization over a global shared variable (2.22). Converting to the vector notation, the constraints $x_i = x_j$ for all $\{i, j\} \in \mathcal{E}$ reads $\bar{B}x = 0$, where $\bar{B}$ is the oriented incidence matrix associated with $\mathcal{G}$. Applying Lagrange duality to the coupling constraint, we find the Lagrangian

$$L(x, y) = \frac{1}{2}(x - c)^\top(x - c) + y^\top\bar{B}x. \tag{4.21}$$

By the first-order optimality conditions for (4.21), we can define the dual problem

$$\text{minimize} \quad -g(y) = -\tfrac{1}{2}y^\top\bar{B}\bar{B}^\top y - y^\top\bar{B}c. \tag{4.22}$$

For given Lagrange multiplier y, the primal variable x will be updated by minimizing the Lagrangian (4.21). The multi-step dual ascent method, hence, suggests the accelerated iterations

$$\begin{aligned} y^{(k+1)} &= y^{(k)} - \alpha(\bar{B}\bar{B}^\top y^{(k)} - \bar{B}c) + \beta\left(y^{(k)} - y^{(k-1)}\right) \\ x^{(k+1)} &= c - \bar{B}^\top y^{(k+1)}. \end{aligned} \tag{4.23}$$

To understand the relationship between this method and alternative schemes in the literature, it is useful to try to eliminate the y-update and only consider the dynamics of the primal variables. To this end, multiplying $\bar{B}^\top$ on both sides of y-update in (4.23) yields

$$\bar{B}^\top y^{(k+1)} = \bar{B}^\top y^{(k)} - \alpha\bar{B}^\top(\bar{B}\bar{B}^\top y^{(k)} - \bar{B}c) + \beta\bar{B}^\top(y^{(k)} - y^{(k-1)}). \tag{4.24}$$

Applying the identity $\bar{B}^\top y^{(k)} = c - x^{(k)}$ and letting $W = \bar{B}^\top\bar{B}$ gives

$$x^{(k+1)} = x^{(k)} - \alpha W x^{(k)} + \beta(x^{(k)} - x^{(k-1)}). \tag{4.25}$$

Similarly, the gradient based counterpart of the consensus iterates is

$$x^{(k+1)} = x^{(k)} - \alpha W x^{(k)}, \tag{4.26}$$

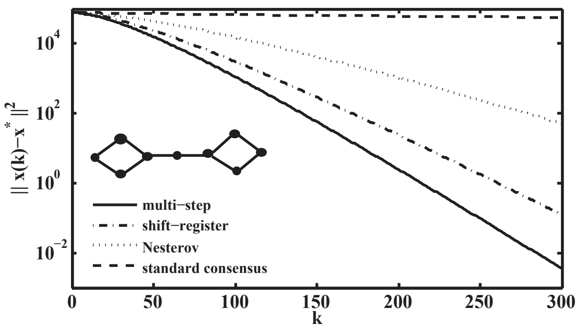


Figure 4.4: Comparison of standard, multi-step, shift-register, and Nesterov consensus algorithms using metropolis wights. Simulation on a dumbbell of 100 nodes: log scale of objective function $\|x^{(k)} - x^\star\|^2$ versus iteration number k. Algorithms start from common initial point $x^{(0)}$.

From the properties of the graph Laplacian, W is positive semidefinite with a simple eigenvalue at 0 and fulfills $W\mathbf{1} = \mathbf{0}$, $\mathbf{1}^\top W = \mathbf{0}^\top$. These iterations are of the same form as (4.20) but use a particular weight matrix. In a fair comparison between the multi-step iterations (4.25) and the basic consensus iterations, the weight matrices of the two approaches should not necessarily be the same, nor necessarily equal to the graph Laplacian. Rather, the weight matrix for the consensus iterations (4.20) should be optimized using the results from [47] and the weight matrix for the multi-step iteration should be computed by using Proposition 4.3.

In addition to the basic consensus iterations with optimal weights, we will also compare our multi-step iterations with two alternative acceleration schemes from the literature. The first one comes from the literature on accelerated consensus and uses shift registers [90], [102], [103]. The idea is to use relaxation technique to accelerate the linear consensus iterates. Recall from Chapter 2 that similarly to the multi-step method, the relaxation method uses a history of past iterates, stored in local registers, when computing the next. For the consensus iterations (4.20), the corresponding shift register iterations are

$$x^{(k+1)} = \eta Q x^{(k)} + (1-\eta) x^{(k-1)}. \tag{4.27}$$

The current approaches to consensus based on shift-registers assume that Q is given and design η to minimize the convergence factor of the iterations. The key results can be traced back to Golub and Varga [104] who determined the optimal η and the associated convergence factor to be

$$\eta^\star = \frac{2}{1+\sqrt{1-\lambda_{n-1}^2(Q)}}, \quad q_{SR}^\star = \sqrt{\frac{1-\sqrt{1-\lambda_{n-1}^2(Q)}}{1+\sqrt{1-\lambda_{n-1}^2(Q)}}}. \tag{4.28}$$

In our comparisons, the shift-register iterations will use the Q-matrix optimized for the basic consensus iterations and the associated $\eta^\star$ given above. The second acceleration technique that we will compare with is the order-optimal gradient iterations (2.11) developed by Nesterov [50]. Recall that while the technique has optimal convergence rate, it is not guaranteed to obtain the best convergence factor. When applying this technique to the consensus problem, following a similar approach as the multi-step iterations, we arrive at the iterations

$$x^{(k+1)} = (I - aW)\left(x^{(k)} + b(x^{(k)} - x^{(k-1)})\right), \tag{4.29}$$

with $W = \bar{B}^\top \bar{B}$, $a = \lambda_n^{-1}(W)$ and $b = (\sqrt{\lambda_n(W)} - \sqrt{\lambda_2(W)})/(\sqrt{\lambda_n(W)} + \sqrt{\lambda_2(W)})$, where $\lambda_2(\cdot)$ and $\lambda_n(\cdot)$ are the smallest and largest non-zero eigenvalues of their variables, respectively.

Figure 4.4 compares the multi-step iterations (4.25) developed in this chapter with (a) the basic consensus iterations (4.20) using a weight matrix determined using the metropolis scheme, (b) the shift-register acceleration (4.27) with the same weight matrix and the optimal η, and (c) the order-optimal method (4.29). The particular results shown are for a network of 100 nodes in a dumbbell topology. The simulations show that all three methods yield a significant improvement in convergence factors over the basic iterations, and that the multi-step method developed in this chapter outperforms the alternatives.

Several remarks are in order. First, since the Hessian of (4.19) is equal to identity matrix, the speed-up of the multi-step iterations is proportional to $\sqrt{\kappa} = \sqrt{\lambda_n(W)/\lambda_2(W)}$. When W equals $\mathcal{L}$, the Laplacian of the underlying graph, we can quantify the speed-ups for certain classes of graphs using spectral graph theory [60]. For example, the complete graph has $\lambda_2(\mathcal{L}) = \lambda_n(\mathcal{L})$ so $\kappa = 1$ and there is no real advantage of the multi-step iterations. On the other hand, for a ring network the eigenvalues of $\mathcal{L}$ are given by $1 - \cos(2\pi k)/|\mathcal{V}|$ for $k = 0, 1, \dots, |\mathcal{V}| - 1$. Therefore, κ grows quickly with the number of nodes, and the performance improvements of (4.25) over (4.26) could be substantial.

Our second remark pertains to the shift-register iterations. Since these iterations have the same form as (4.25), we can go beyond the current literature on shift-register consensus (which assumes Q to be given and optimizes η) and provide jointly optimal weight matrix and η-parameter:

Proposition 4.6

The weight matrix $Q^\star$ and constant $\eta^\star$ that minimizes the convergence factor of the shift-register consensus iterations (4.27) are $Q^\star = I - \theta^\star W^\star$, where $W^\star$ is computed in Proposition 4.3 with $H = I$ and $\theta^\star = \frac{2}{\lambda_2(W^\star)+\lambda_n(W^\star)}$ while $\eta^\star = 1 + \beta^\star$ and $\beta^\star$ is given in Theorem 4.1.

4.6.3 Internet congestion control

Our final application is related to the area of Internet congestion control, where Network Utility Maximization (NUM) has emerged as a powerful framework for studying various important resource allocation problems, see, e.g., [64, 15, 13, 14]. The vast majority of the work in this area is based on the dual decomposition approach introduced in [15]. Here, the optimal bandwidth sharing among S flows in a data network is posed as the optimizer of a convex optimization problem

$$\begin{array}{ll} \underset{x}{\text{maximize}} & \sum_s u_s(x_s) \\ \text{subject to} & Rx \leq c \\ & x_s \in [m_s, M_s]. \end{array} \tag{4.30}$$

Here, x_s is the communication rate of flow s, and $u_s(x_s)$ is a strictly concave and increasing function that describes the utility that source s has of communicating at rate x_s. The communication rate is subject to upper and lower bounds. Finally, $R \in \{0,1\}^{L\times S}$ is a routing matrix whose entries $R_{\ell s} = 1$ if flow s traverses link ℓ and $R_{\ell s} = 0$ otherwise. In this way, Rx is the total traffic on links that cannot exceed the link capacities $c \in \mathbb{R}^n$. We make the following assumptions.

Assumption 4.2 For the problem (4.30) it holds that

(i) Each $u_s(x_s)$ is twice continuously differentiable and satisfies $0 < \mu < -\nabla^2 u_s(x_s) < L$ for $x_s \in [m_s, M_s]$.

(ii) For every link ℓ, there exists a source s whose flow only traverses ℓ, i.e., $R_{\ell s} = 1$ and $R_{\ell' s} = 0$ for all $\ell' \neq \ell$.

While these assumptions appear restrictive, they are often postulated in the literature (e.g., [15, Assumptions C1-C4]). Note that under Assumption 4.2, the routing matrix has full row-rank and all the link constraints hold with equality at optimum. Hence, we can replace $Rx \leq c$ in (4.30) with $Rx = c$. Following the steps of the dual ascent method in Section 4.4, we have the following primal-dual iterations

$$x_s^{(k+1)} = \underset{z\in[m_s,M_s]}{\arg\max}\, u_s(z) - z\sum_\ell R_{\ell s} y_\ell^{(k)} \tag{4.31}$$

$$y_\ell^{(k+1)} = y_\ell^{(k)} + \alpha\left(\sum_\ell R_{\ell s} x_s^{(k+1)} - c_\ell\right). \tag{4.32}$$

Note that each source solves a localized minimization problem based on the sum of the Lagrange multipliers for the links that the flow traverses; this information can be effectively signaled back to the source explicitly or implicitly using the end-to-end acknowledgements. The Lagrange multipliers, on the other hand, are updated by individual links based on the difference between the total traffic imposed by the sources and the capacity of link. Clearly, this information is also locally available. It is possible to show that under the conditions of Assumption 4.2, the dual function is strongly concave, differentiable, and has a Lipschitz-continuous gradient [15]. Hence, by standard arguments, the updates (4.31) and (4.32) converge to a primal-dual optimal point $(x^\star, y^\star)$ for appropriately chosen step-size α. Our results from Section 4.4 indicate that substantially improved convergence factors could be obtained by the following class of multi-step updates of the Lagrange multipliers

$$y_\ell^{(k+1)} = y_\ell^{(k)} + \alpha \left(\sum_\ell R_{\ell s} x_s^{(k+1)} - c_\ell \right) + \beta (y_\ell^{(k)} - y_\ell^{(k-1)}). \tag{4.33}$$

To tune the step-sizes in an optimal way, we bring the techniques from Section 4.4 into action. To do so, we first bound the eigenvalues of $RR^\top$ using the following result:

Lemma 4.4

Let $R \in \{0,1\}^{L\times S}$ satisfy Assumption 4.2. Then

$$1 \le \lambda_1(RR^\top), \quad \lambda_n(RR^\top) \le \ell_{\max} s_{\max},$$

where $\ell_{\max} = \max_s \sum_\ell R_{\ell s}$ and $s_{\max} = \max_\ell \sum_s R_{\ell s}$.

The optimal step-size parameters and corresponding convergence factor now follow from Lemma 4.4 and Theorem 4.2:

Proposition 4.7

Consider the NUM problem (4.30) under Assumption 4.2. Then, for $0 \le \beta < 1$ and $0 < \alpha < 2(1+\beta)/(L\ell_{\max} s_{\max})$, the iterations (4.31) and (4.33) converge linearly to a primal-dual optimal pair. The step-sizes

$$\alpha = \left(\frac{2}{\sqrt{L\ell_{\max} s_{\max}} + \sqrt{\mu}} \right)^2, \ \beta = \left(\frac{\sqrt{L\ell_{\max} s_{\max}} - \sqrt{\mu}}{\sqrt{L\ell_{\max} s_{\max}} + \sqrt{\mu}} \right)^2,$$

ensure that the convergence factor of the dual iterates is

$$q_{\text{NUM}} = \frac{\sqrt{L\ell_{\max} s_{\max}} - \sqrt{\mu}}{\sqrt{L\ell_{\max} s_{\max}} + \sqrt{\mu}}.$$

Note that an upper bound on the Hessian of the dual function was also derived in [15]. However, strong concavity was not explored and the associated bounds were not derived.

We now comment on the steady behavior of accelerated link price algorithm (4.33). From the saturation of link traffics due to Assumption 4.2 as $k \to \infty$, close to the equilibrium, we have

$$\alpha \left(\sum_\ell R_{\ell s} x_s^{(k)} - c_\ell \right) \to 0.$$

Therefore, for large values of k, we approximate the link price updates (4.33) as

$$\begin{aligned} y_\ell^{(k+1)} &= y_\ell^{(k)} + \beta\big(y_\ell^{(k)} - y_\ell^{(k-1)}\big) \\ y_\ell^{(k+1)} - y_\ell^\star &= y_\ell^{(k)} - y_\ell^\star + \beta\Big(\big(y_\ell^{(k)} - y_\ell^\star\big) - \big(y_\ell^{(k-1)} - y_\ell^\star\big)\Big) \\ e_\ell^{(k+1)} &= e_\ell^{(k)} + \beta\big(e_\ell^{(k)} - e_\ell^{(k-1)}\big), \end{aligned} \tag{4.34}$$

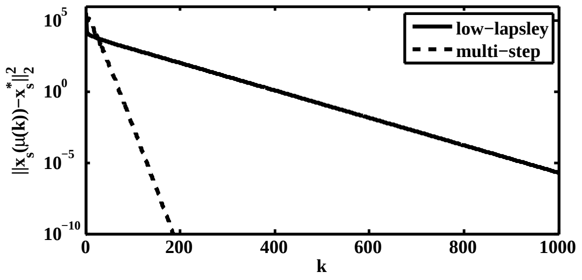


Figure 4.5: Convergence of Low-Lapsley [15] and multi-step. Plot shows log-scale of the Euclidian distance from optimal source rates $\|x_s^{(k)} - x_s^\star\|^2$ vs. the iteration count k.

where $y_\ell^\star$ is the optimal price of link ℓ and $e_\ell^{(k)} \triangleq y_\ell^{(k)} - y_\ell^\star$ is the discrepancy between the current price and the optimal price of link ℓ. It is easy to note that (4.34) corresponds to a Proportional-Derivative (PD) controller for driving the price of link ℓ to its optimal value. Hence, it is obvious that asymptotically (4.33) behaves like a PD controller.

To compare the gradient iterations with the multi-step congestion control mechanism, we present representative results from a network with 10 links and 20 flows which satisfies Assumption 4.2. The utility functions are of the form $-(M_s - x_s)^2/2$ with $m_s = 0$ and $M_s = 10^5$ for all sources. As shown in Figure 4.5, substantial speedups are obtained.

As a final remark, note that Lemma 4.4 underestimates λ_1 and overestimates λ_n, so we have no formal guarantee that the multi-step method will always outperform the gradient-based algorithm. However, in our experiments with a large number of randomly generated networks, the disadvantageous situation identified in Section 4.5 never occurred.

4.7 Summary

In this chapter, we studied accelerated gradient methods for distributed optimization problems. In particular, we considered problems with twice differentiable strongly convex cost functions subject to a set of network constraints. Given the bounds of the Hessian of the objective function and the Laplacian of the underlying communication graph, we derived primal and dual multi-step techniques that allow improving the convergence factors significantly compared to the standard gradient-based techniques.

We derived optimal parameters and convergence factors, and characterized the robustness of our tuning rules to errors that occur when critical problem parameters are not known but have to be estimated. Our multi-step techniques were applied to three classes of problems: distributed resource allocation under a network-wide resource constraint, distributed averaging, and Internet congestion control. We demonstrated, both analytically and through numerical simulations, that the approaches developed in this chapter significantly outperform alternatives from the literature.

Appendix

4.A Network optimization with coupled costs and higher dimensional variables

The proposed method in this chapter is indeed applicable for a class of problems of the general form (4.1). However, there exist several applications that can be cast as this form. Apart from the examples presented in the chapter, here we consider a form of coupled cost functions and convert it into the canonical form (4.1). Consider the following problem

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{V},j\in\mathcal{C}_i} f_i(x_i, x_j) \\ \text{subject to} & Ax = b, \end{array}$$

where $\mathcal{C}_i \subseteq \mathcal{N}_i$ is the set of neighbors $j \in \mathcal{N}_i$ such that f_i depends on their decision variable x_j. Moreover, $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{R}^n$ is a vector consisting of local decision variables $x_i \in \mathbb{R}$. One can recast the former problem into

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{V},j\in\mathcal{C}_i} f_i(x_i, x_{ij}) \\ \text{subject to} & Ax = b \\ & x_{ij} = x_j, \quad \forall i \in \mathcal{V},\ j \in \mathcal{C}_i, \end{array}$$

where x_{ij} is a local copy of the general variable x_j. Define $z_i \in \mathbb{R}^{|\mathcal{C}_i|}$ as the vector of local copies $\{x_{ij} | j \in \mathcal{C}_i\}$ at agent i. For each agent i, the last constraint in the aforementioned problem can be written as $E_i x = z_i$ where $E_i \in \mathbb{R}^{|\mathcal{C}_i|\times n}$, and is defined in the following way. Assume that the m-th component ($m < |\mathcal{C}_i|$) of z_i is x_{ik} then m-th row of E_i is all zeros except the k-th column that equals 1. Now, letting $z \triangleq [\{z_i^\top\}_{i=1}^{|\mathcal{V}|}]^\top$ and $E \triangleq [\{E_i^\top\}_{i=1}^{|\mathcal{V}|}]^\top$ be the stacked matrices of the newly defined variables, one can obtain the general problem formulation

$$\begin{array}{ll} \text{minimize} & \sum_i f_i(\overline{x}_i) \\ \text{subject to} & \underbrace{\begin{bmatrix} A & \mathbf{0} \\ E & -I \end{bmatrix}}_{\hat{A}} \underbrace{\begin{bmatrix} x \\ z \end{bmatrix}}_{\hat{x}} = \underbrace{\begin{bmatrix} b \\ \mathbf{0} \end{bmatrix}}_{\hat{b}}, \end{array}$$

where $\overline{x}_i \triangleq \begin{bmatrix} x_i & z_i^\top \end{bmatrix}^\top$ is the local decision vector. Given that b belongs to the column space of A, it can be observed that $\hat{A}\hat{x} = \hat{b}$ is also feasible. Therefore, we have reformulated an originally coupled problem into the canonical form (4.1) with decoupled cost functions. One might apply the results of either multi-step gradient technique or the dual ascent method to accelerate this problem.

Above arguments can be generalized to those problems with local variables of higher dimensions. Particularly, in the case of vector valued decision variables, one needs to augment the constraint matrix A and the weight matrix W in the following way. Consider $|\mathcal{V}|$ number of agents with local decision variables $x_i \in \mathbb{R}^{n_i}$ that aim at solving (4.1). Assume that the stack vector x lives in $\mathbb{R}^n$ with $n = \sum_{i \in \mathcal{V}} n_i$. One has to update the constraint matrix A, as $A = [A_1 \dots A_{|\mathcal{V}|}] \in \mathbb{R}^{m \times n}$ where $A_i \in \mathbb{R}^{m \times n_i}$ is the constraint matrix for agent i. The smoothness assumption of the cost function of each agent takes the form $l_i I_{n_i} \preceq \nabla^2 f_i(x_i) \preceq u_i I_{n_i}$, $\forall x_i$ for nonzero $l_i, u_i \in \mathbb{R}_+$, under which the strong convexity constant μ and Lipschitz-continuity constant L remain untapped as in (4.3).

The sparsity pattern of the wight matrix $W \in \mathbb{R}^{n \times n}$ is defined as $W_{ij} = \mathbf{0}_{n_i \times n_j}$ if $i \neq j$ and $\{i, j\} \notin \mathcal{E}$. The results of the rest of the chapter will hold with aforementioned extended dimensions.

4.B Heuristic weights for weighted gradient method

Consider the resource allocation problem in Example 4.1 and the weighting matrix W in the associated weighted gradient iterates (4.6) to solve it.

One heuristic matrix that satisfies these constraints on W is the Laplacian of the underlying graph. Recall from Chapter 2 that $\mathcal{L} = D - A$ with D and A being the degree and the adjacency matrices defined on $\mathcal{G}$, respectively. To ensure convergence of the iteration (4.6), the Laplacian has to be appropriately scaled $W = -\delta\mathcal{L}$ for some $\delta \in \mathbb{R}$. Note that the Laplacian weights relevant to a specific node can be determined using local topology information. But $W = -\delta\mathcal{L}$ might need some global information depending on δ. Specifically, we have

$$W_{ij} = \begin{cases} \delta & (i,j) \in \mathcal{E}, \\ -\delta d_i & i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{4.35}$$

where d_i is the degree of node i; i.e., we have $d_i = D_{ii}$. Following the notation in [63], we call these weights constant since they assign a constant weight to all edges and then adjust W_{ii} such that $W\mathbf{1} = \mathbf{0}$. Several heuristic weight choices are introduced in [63] that include: the max-degree weights where $\delta = -1/(\max_{i \in \mathcal{V}} d_i u_i)$; the best constant weights, for which $\delta = -2/(\lambda_2(\mathcal{L}) + \lambda_n(\mathcal{L}))$; and the locally determined Metropolis-Hasting weights

$$W_{ij} = \begin{cases} -\min\{1/(d_i u_i), 1/(d_j u_j)\} & \{i,j\} \in \mathcal{E}, \\ -\sum_{k \in \mathcal{N}_i} W_{ik} & i = j, \\ 0 & \text{otherwise}. \end{cases} \tag{4.36}$$

4.C Proofs

Here we present the proofs of the chapter.

4.C.1 Proof of Theorem 4.1

Let $x^\star$ be the optimizer of (4.1). The Taylor series expansion of $\nabla f\left(x^{(k)}\right)$ around $x^\star$ yields

$$\begin{aligned} W\nabla f\left(x^{(k)}\right) &\cong W(\nabla f(x^\star) + \nabla^2 f(x^\star)(x^{(k)} - x^\star)) \\ &= W\nabla^2 f(x^\star)(x^{(k)} - x^\star) \\ &\triangleq WH(x^{(k)} - x^\star), \end{aligned}$$

where the first identity holds because $W\nabla f(x^\star) = 0$ due to (4.5) and (4.7). Introducing $z(k) \triangleq [x^{(k)} - x^\star,\, x^{(k-1)} - x^\star]^\top$, we can thus re-write (4.8) as

$$z(k+1) = \underbrace{\begin{bmatrix} C & -\beta I \\ I & 0 \end{bmatrix}}_{\Gamma} z(k) + o(z(k)^2), \tag{4.37}$$

where $C \triangleq (1+\beta)I - \alpha WH$. Now, for non-zero vectors v_1 and v_2, consider the eigenvalue equation

$$\begin{bmatrix} C & -\beta I \\ I & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \phi(\Gamma) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

Since $v_1 = \phi(\Gamma)v_2$, the first row can be re-written as

$$\left(-\phi^2(\Gamma)I + \phi(\Gamma)C - \beta I\right) v_2 = 0. \tag{4.38}$$

Note that (4.38) is a polynomial in C and C is in turn a polynomial in WH. Hence, if λ denotes an eigenvalue of WH, we have

$$\phi^2(\Gamma) - (1 + \beta - \alpha\lambda)\,\phi(\Gamma) + \beta = 0. \tag{4.39}$$

The roots of (4.39) have the form

$$\phi(\Gamma) = \frac{1 + \beta - \alpha\lambda \pm \sqrt{\Delta}}{2}, \quad \Delta = (1 + \beta - \alpha\lambda)^2 - 4\beta. \tag{4.40}$$

If $\Delta \geq 0$, then $|\phi(\Gamma)| < 1$ is equivalent to

$$\begin{aligned} &(1 + \beta - \alpha\lambda)^2 - 4\beta \geq 0 \\ &-2 < 1 + \beta - \alpha\lambda \pm \sqrt{(1 + \beta - \alpha\lambda)^2 - 4\beta} < 2, \end{aligned}$$

which, after simplifications, yields $0 < \alpha < 2(1+\beta)/\lambda$.

Furthermore, if $\Delta < 0$, then $|\phi(\Gamma)| < 1$ is equivalent to

$$0 \leq \frac{(1 + \beta - \alpha\lambda)^2 - \Delta}{4} < 1,$$

which, after similar simplifications, implies $0 \leq \beta < 1$.

Note that the upper bound for α gives a necessary condition for λ. Here we find an upper bound for this eigenvalue. Since H is a positive diagonal matrix, under similarity equivalence we have $WH \sim H^{1/2}WHH^{-1/2} = H^{1/2}WH^{1/2}$. Without loss of generality assume $x \in \mathbb{R}^n$ and $x^\top x = 1$, then $x^\top WHx = x^\top H^{1/2}WH^{1/2}x = y^\top Wy$, where $y = H^{1/2}x$. Clearly, for $y^\top Wy$ it holds that $\lambda_1(W)y^\top y \leq y^\top Wy \leq \lambda_n(W)y^\top y$. Now, $\mu \leq y^\top y = x^\top Hx \leq L$, implies $\mu\lambda_1(W) \leq x^\top WHx \leq L\lambda_n(W)$. Hence, a sufficient condition on α reads

$$0 < \alpha < \frac{2(1+\beta)}{L\lambda_n(W)}. \tag{4.41}$$

Having proven the sufficient conditions for convergence stated in the theorem, we now proceed to estimate the convergence factor. To this end, we need the following lemmas describing the eigenvalue characteristics of WH and Γ.

Lemma 4.5

If W has $m < n$ zero eigenvalues, then WH has exactly $n - m$ nonzero eigenvalues, i.e. $\lambda_1(WH) = \cdots = \lambda_m(WH) = 0$, and $\lambda_i(WH) \neq 0$, for $i = m+1, \cdots, n$.

Proof. From [99] we know that if and only if all the principal sub-matrices of a matrix have nonnegative determinants then that matrix is positive semi-definite. Note that the i-th principal sub-matrix of WH, WH_i, is obtained by multiplication of the corresponding principal sub-matrix of W, W_i by the same principal sub-matrix of H, H_i from the right, and we have $\det(WH_i) = \det(W_i)\det(H_i)$. We know $\det(H_i) > 0$ and $\det(W_i) \geq 0$ because $W \geq 0$, thus $\det(WH_i) \geq 0$ and WH is positive semi-definite. Furthermore, $\text{rank}(WH) = \text{rank}(W)$. So $\text{rank}(WH) = n - m$ which means that WH has exactly m zero eigenvalues. □

Lemma 4.6

For any WH such that $\lambda_i(WH) = 0$ for $i = 1, \cdots, m$, and $\lambda_i(WH) \neq 0$, for $i = m+1, ..., n$., the matrix Γ has m eigenvalues equal to 1 and the absolute values of the rest of the $2n - m$ eigenvalues are strictly less than 1.

Proof. For complex $\phi_i(\Gamma)$ we have $|\phi_i(\Gamma)| = \beta < 1$. For real-valued $\phi_i(\Gamma)$, on the other hand, the bound on α implies that $\alpha(\lambda(WH))$ is a decreasing function of λ. In this case, $0 < \alpha < \frac{2(1+\beta)}{\overline{\lambda}}$ guarantees that $0 < \alpha < \frac{2(1+\beta)}{\lambda_i(WH)}$ for any $0 < \lambda_i(WH) \leq \overline{\lambda}$. Note that if we set a tighter bound on α, then it does not change satisfactory condition for having $|\phi(\Gamma)| < 1$. Only when $\lambda_i(WH) = 0$, we have $\lim_{x\to 0}\alpha = \infty$. For this case, if we substitute $\lambda_i(WH) = 0$ in (4.39) we obtain $\phi_{2i-1}(\Gamma) = 1$ and $\phi_{2i}(\Gamma) = \beta < 1$. □

We are now ready to prove the remaining parts of Theorem 4.1. By the lemmas above, Γ has $m < n$ eigenvalues equal to 1, which correspond to the m zero eigenvalues of W implied by the optimality condition (4.7). Hence, minimizing $m + 1$-th largest eigenvalue of (4.37) leads to the optimal convergence factor of the multi-step weighted gradient iterations (4.8). Calculating $\overline{\phi}_\Gamma \triangleq \min_{\alpha,\beta} \max_{1\leq j\leq 2n-m} |\phi_j(\Gamma)|$ yields the optima $\alpha^\star$ and $\beta^\star$. Noting that eigenvalues of Γ are given by (4.40), we have

$$\overline{\phi}_\Gamma = \frac{1}{2} \max\left\{|1+\beta-\alpha\lambda_i| + \sqrt{(1+\beta-\alpha\lambda_i)^2 - 4\beta}\right\},$$

where $\lambda_i \triangleq \lambda_i(WH)$, $\forall i = m+1, .., n$. There are two cases:

Case 1: $(1+\beta-\alpha\lambda_i)^2 - 4\beta \geq 0$. Then, a and b are non-negative and real with $a \geq b$. Hence, $a^2 - b^2 \geq (a-b)^2$ and consequently $a + \sqrt{a^2-b^2} \geq 2a - b \geq b$.

Case 2: $(1+\beta-\alpha\lambda_i)^2 - 4\beta < 0$. In this case, $\phi_i(\Gamma)$ is complex-valued. Consider $c, d \in \mathbb{R}^+$ with $c < d$. Then, $|c + \sqrt{c^2-d}| = \sqrt{c^2 - c^2 + d} = \sqrt{d} \geq 2c - \sqrt{d}$.

Substitute these results into $\overline{\phi}_\Gamma$ with $a = 1+\beta-\alpha\lambda_i$, $b = 2\sqrt{\beta}$, $c = |1+\beta-\alpha\lambda_i|$, and $d = 4\beta$, we get

$$\overline{\phi}_\Gamma \geq \max\left\{\sqrt{\beta}, \max\left\{|1+\beta-\alpha\lambda_i| - \sqrt{\beta}\right\}\right\},$$

which can be expressed in terms of $\underline{\lambda}$ and $\overline{\lambda}$:

$$\overline{\phi}_\Gamma \geq \max\left\{\sqrt{\beta}, |1+\beta-\alpha\underline{\lambda}| - \sqrt{\beta}, |1+\beta-\alpha\overline{\lambda}| - \sqrt{\beta}\right\}. \tag{4.42}$$

It can be verified that

$$\begin{aligned} \max \quad & \{|1+\beta-\alpha\underline{\lambda}| - \sqrt{\beta}, |1+\beta-\alpha\overline{\lambda}| - \sqrt{\beta}\} \\ & \geq |1+\beta-\alpha'\underline{\lambda}| - \sqrt{\beta}, \end{aligned} \tag{4.43}$$

where α' is such that $|1+\beta-\alpha'\underline{\lambda}| = \left|1+\beta-\alpha'\overline{\lambda}\right|$, i.e.,

$$\alpha' = \frac{2(1+\beta)}{\underline{\lambda}+\overline{\lambda}}. \tag{4.44}$$

Combining (4.42), (4.43), and (4.44), we thus obtain

$$\overline{\phi}_\Gamma \geq \max\left\{\sqrt{\beta}, (1+\beta)\frac{\overline{\lambda}-\underline{\lambda}}{\overline{\lambda}+\underline{\lambda}} - \sqrt{\beta}\right\}. \tag{4.45}$$

Again, the max-operator can be bounded from below by its value at the point where the arguments are equal. To this end, consider β' which satisfies $\sqrt{\beta'} = (1+\beta')\frac{\overline{\lambda}-\underline{\lambda}}{\overline{\lambda}+\underline{\lambda}} - \sqrt{\beta'}$, that is,

$$\beta' = \left(\frac{\sqrt{\overline{\lambda}} - \sqrt{\underline{\lambda}}}{\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}}\right)^2. \tag{4.46}$$

Since $\max\left\{\sqrt{\beta}, (1+\beta)\frac{\overline{\lambda}-\underline{\lambda}}{\overline{\lambda}+\underline{\lambda}} - \sqrt{\beta}\right\} \geq \sqrt{\beta'}$, we can combine this with (4.45) to conclude that

$$\overline{\phi}_\Gamma \geq \sqrt{\beta'} = \frac{\sqrt{\overline{\lambda}} - \sqrt{\underline{\lambda}}}{\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}}. \tag{4.47}$$

Our proof is concluded by noting that equality in (4.47) is attained for the smallest non-zero eigenvalue of Γ and the optimal step-sizes $\beta^\star$ and $\alpha^\star$ stated in the body of the theorem.

4.C.2 Proof of Proposition 4.1

As shown in the proof of Theorem 4.1, the eigenvalues of WH are equal to those of $H^{1/2}WH^{1/2}$. According to [99, p.225] for matrices W and $H^{1/2}WH^{1/2}$, there exists a nonnegative real number θ_k such that $\lambda_1(H) \leq \theta_k \leq \lambda_n(H)$ and $\lambda_k(H^{1/2}WH^{1/2}) = \theta_k \lambda_k(W)$. Letting $k = m+1$ and $k = n$, yields $\underline{\lambda} \geq \mu \underline{\lambda}_W$ and $\overline{\lambda} \leq L\overline{\lambda}_W$. The rest of the proof is similar to that of Theorem 4.1 and is omitted for brevity.

4.C.3 Proof of Proposition 4.2

Direct calculations yield $q^\star = (\sqrt{\overline{\lambda}} - \sqrt{\underline{\lambda}})/(\sqrt{\overline{\lambda}} + \sqrt{\underline{\lambda}}) = 1 - 2/((\overline{\lambda}/\underline{\lambda})^{1/2} + 1)$. Similarly, $\widetilde{q} = 1 - 2/((\overline{\lambda}_W/\underline{\lambda}_W)^{1/2} + 1)$. Hence, minimizing $q^\star$ and $\widetilde{q}$ are equivalent to minimizing the condition number of WH and W, respectively.

4.C.4 Proof of Proposition 4.3

As shown in the proof of Theorem 1, the eigenvalues of WH are equal to those of $\Omega \triangleq H^{1/2}WH^{1/2}$. Thus, combined with the constraint that $W \succeq 0$ the problem of minimizing $\overline{\lambda}/\underline{\lambda}$ is equivalent to minimizing L/μ, where L and μ are the largest and the smallest non-zero eigenvalues of Ω. Next we will construct the constraint set of this optimization problem. First, recall that W should provide the sparsity pattern induced by $\mathcal{G}$, i.e., $W \in \mathcal{A}$. Second, W fulfills (4.7). For the case that W is symmetric, this constraint can be rewritten in terms of Ω in the form $\Omega V = H^{1/2}WH^{1/2}V = 0$ where $V = H^{-1/2}A^\top$. Third constraint is to bound the remaining $n - m$ eigenvalues of Ω away from zero. Let $v \in \mathbb{R}^n$ be a column of i, and let $v^\perp \in \mathbb{R}^n$ be orthogonal to i. Since $\Omega \geq 0$ and $\Omega v = 0$ then we have $\quad x^\top \Omega x > 0 \quad \forall x \in v^\perp$. This condition is equivalent to $P^\top \Omega P > 0$, where $P = [p_1, p_2, ..., p_{n-m}] \in \mathbb{R}^{n \times n-m}$ is a matrix of orthonormal vectors spanning the null space of $V^\top$. More explicitly, one can define this subspace by unit vectors satisfying $p_i^\top v_j = 0$, $\forall i = 1, \ldots, n-m$, $j = 1, \ldots, n$, $p_i^\top p_k = 0$, $\forall i \neq k$. The optimization problem becomes

$$\begin{array}{ll} \text{minimize} & L/\mu \\ \text{subject to} & \mu I \preceq P^\top \Omega P \preceq LI, \\ & W \in \mathcal{A},\ W \succeq 0,\ \Omega V = 0. \end{array}$$

Denoting $t = L/\mu$ and $\gamma = 1/\mu$, this problem can be recast as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & I \preceq \gamma P^\top \Omega P \preceq tI,\ W \in \mathcal{A}, \\ & W \succeq 0,\ \Omega V = 0,\ \gamma > 0. \end{array} \tag{4.48}$$

Finally, the constraint on γ in (4.48) can be omitted. Specifically, consider $(W^\star, \alpha^\star, \beta^\star)$ to be the joint-optimal weight and stepsizes given by (4.48) and (4.9), respectively. One can replace any positive scale $\gamma W^\star$ in (4.9) and derive the step-sizes $\alpha = \alpha^\star/\gamma$ and $\beta = \beta^\star$. It is easy to check that the triple $(\gamma W^\star, \alpha, \beta)$ leads to an identical multi-step iterations (4.8) as the optimal ones.

4.C.5 Proof of Lemma 4.2

Since f is twice differentiable on $[x^\star, x]$, we have

$$\begin{aligned}\nabla f(x) &= \nabla f(x^\star) + \int_0^1 \nabla^2 f(x^\star + \tau(x - x^\star))(x - x^\star)d\tau \\ &= A^\top y^\star + H(x)(x - x^\star),\end{aligned}$$

where we used the fact that $\nabla f(x^\star) = A^\top y^\star$ and we introduced

$$H(x) = \int_0^1 \nabla^2 f(x^\star + \tau(x - x^\star))d\tau.$$

By virtue of Assumption 4.1, $H(x)$ is symmetric and nonnegative definite and satisfies $\mu I \preceq H(x) \preceq LI$ [52]. Hence from (4.6) and (4.7)

$$\begin{aligned}\|x^{(k+1)} - x^\star\| &= \|x^{(k)} - x^\star - \alpha W \nabla f\big(x^{(k)}\big)\| \\ &= \|x^{(k)} - x^\star - \alpha W(A^\top y^\star + H(x^{(k)})(x^{(k)} - x^\star))\| \\ &= \|(I - \alpha W H(x^{(k)}))(x^{(k)} - x^\star)\| \\ &\le \|I - \alpha W H(x^{(k)})\| \|x^{(k)} - x^\star\|.\end{aligned}$$

The rest of the proof follows the same steps as [52, Theorem 3]. Essentially for fixed step-size $0 < \alpha < 2/\overline{\lambda}$, (4.6) converges linearly with factor $q_2 = \max\{|1 - \alpha\underline{\lambda}|, |1 - \alpha\overline{\lambda}|\}$. The minimum convergence factor $q_G^\star = \frac{\overline{\lambda}-\underline{\lambda}}{\underline{\lambda}+\overline{\lambda}}$ is obtained by minimizing q_G over α, which yields the optimal step-size $\alpha^\star = \frac{2}{\underline{\lambda}+\overline{\lambda}}$.

4.C.6 Proof of Proposition 4.4

According to Lemma 4.2, the weighted gradient iterations (4.6) with estimated step-size $\widetilde{\alpha} = 2/(\underset{\sim}{\lambda} + \widetilde{\lambda})$ will converge provided that $0 < \widetilde{\alpha} < 2/\overline{\lambda}$, i.e., when $\overline{\lambda} < \underset{\sim}{\lambda} + \widetilde{\lambda}$. For the multi-step algorithm (4.8), Theorem 4.1 guarantees convergence if $0 \le \widetilde{\beta} < 1$, $0 < \widetilde{\alpha} < 2(1 + \widetilde{\beta})/\overline{\lambda}$. The assumption $0 < \underset{\sim}{\lambda} \le \widetilde{\lambda}$ implies that the condition on $\widetilde{\beta}$ is always satisfied. Regarding $\widetilde{\alpha}$, inserting the expression for $\widetilde{\beta}$ in the upper bound for $\widetilde{\alpha}$ and simplifying yield

$$\frac{4}{\left(\sqrt{\underset{\sim}{\lambda}} + \sqrt{\widetilde{\lambda}}\right)^2} < 2\frac{2(\widetilde{\lambda} + \underset{\sim}{\lambda})}{\left(\sqrt{\widetilde{\lambda}} + \sqrt{\underset{\sim}{\lambda}}\right)^2}\frac{1}{\overline{\lambda}},$$

which is satisfied if $0 < \overline{\lambda} < \widetilde{\lambda} + \underset{\sim}{\lambda}$. The statement is proven.

4.C.7 Proof of Lemma 4.3

We consider two cases. First, when $\underset{\sim}{\lambda} + \widetilde{\lambda} < \underline{\lambda} + \overline{\lambda}$ combined with the assumption that $0 < \overline{\lambda} < \underset{\sim}{\lambda} + \widetilde{\lambda}$ yields $\widetilde{\alpha}\overline{\lambda} > 1$, which means that $|1 - \widetilde{\alpha}\overline{\lambda}| = \widetilde{\alpha}\overline{\lambda} - 1$. Moreover, $\widetilde{\alpha}\overline{\lambda} - 1 \ge$

$1 - \tilde{\alpha}\underline{\lambda}$, so by Lemma 4.2,

$$\tilde{q}_G = \max\{\tilde{\alpha}\bar{\lambda} - 1, \max\{1 - \tilde{\alpha}\underline{\lambda}, \tilde{\alpha}\underline{\lambda} - 1\}\} = \tilde{\alpha}\bar{\lambda} - 1 = 2\bar{\lambda}/(\utilde{\lambda} + \tilde{\lambda}) - 1.$$

The second case is when $\utilde{\lambda} + \tilde{\lambda} > \underline{\lambda} + \bar{\lambda}$. Then, $\tilde{\alpha}\underline{\lambda} < 1$ and hence $|1 - \tilde{\alpha}\underline{\lambda}| = 1 - \tilde{\alpha}\underline{\lambda}$. Moreover, $1 - \tilde{\alpha}\underline{\lambda} \geq \tilde{\alpha}\bar{\lambda} - 1$, so

$$\tilde{q}_G = \max\{1 - \tilde{\alpha}\underline{\lambda}, \max\{\tilde{\alpha}\bar{\lambda} - 1, 1 - \tilde{\alpha}\bar{\lambda}\}\} = 1 - \tilde{\alpha}\underline{\lambda}.$$

The convergence factor of the multi-step iterations with inaccurate step-sizes (4.15) follows directly from Theorem 4.1.

4.C.8 Proof of Proposition 4.5

We analyze the four quadrants $\mathcal{Q}_1$ through $\mathcal{Q}_4$ in order.

$\mathcal{Q}_1$: when $(\utilde{\varepsilon}, \tilde{\varepsilon}) \in \mathcal{Q}_1$ we have $\utilde{\lambda} > \underline{\lambda}$ and $\tilde{\lambda} > \bar{\lambda} > \bar{\lambda}$. From the convergence factor of multi-step gradient method (4.17) it then follows that $\tilde{q} = 1 + \tilde{\beta} - \tilde{\alpha}\underline{\lambda} - \tilde{\beta}^{1/2}$. Moreover, since in this quadrant $\tilde{\lambda} + \utilde{\lambda} \geq \bar{\lambda} + \underline{\lambda}$, from (4.16) we have $\tilde{q}_G = 1 - 2\underline{\lambda}/(\utilde{\lambda} + \tilde{\lambda})$. A direct comparison between the two expressions yields that $\tilde{q} \leq \tilde{q}_G$.

$\mathcal{Q}_2$: when $(\utilde{\varepsilon}, \tilde{\varepsilon}) \in \mathcal{Q}_2$ we have $\underline{\lambda} < \utilde{\lambda}$ and $\tilde{\lambda} < \bar{\lambda}$. Combined with the stability assumption $\utilde{\lambda} + \tilde{\lambda} > \bar{\lambda}$, straightforward calculations show that the convergence factor of the multi-step iterations with inaccurate step-sizes (4.15) is

$$\tilde{q} = \begin{cases} \tilde{\alpha}\bar{\lambda} - \tilde{\beta} - 1 - \sqrt{\tilde{\beta}} & \tilde{\lambda} + \tilde{\lambda} \leq \underline{\lambda} + \bar{\lambda}, \\ 1 + \tilde{\beta} - \tilde{\alpha}\underline{\lambda} - \sqrt{\tilde{\beta}} & \text{otherwise.} \end{cases}$$

Moreover, for this quadrant the convergence factor of weighted gradient method is given by (4.16). To verify that $\tilde{q} < \tilde{q}_G$, we perform the following comparisons:

(a) If $\utilde{\lambda} + \tilde{\lambda} < \underline{\lambda} + \bar{\lambda}$ then we have $\tilde{q} = \tilde{\alpha}\bar{\lambda} - \tilde{\beta} - 1 - \tilde{\beta}^{1/2}$ and $\tilde{q}_G = (2\bar{\lambda})/(\utilde{\lambda} + \tilde{\lambda}) - 1$. To show that $\tilde{q} < \tilde{q}_G$, we rearrange it to obtain the following inequality

$$\Delta \triangleq (\bar{\lambda} - \tilde{\lambda} + \tilde{\lambda}^{1/2}\utilde{\lambda}^{1/2})(\tilde{\lambda} + \utilde{\lambda}) - 2\bar{\lambda}\tilde{\lambda}^{1/2}\utilde{\lambda}^{1/2} < 0.$$

After simplifications

$$\Delta = (\tilde{\lambda}^{1/2} - \utilde{\lambda}^{1/2})\Big(-\tilde{\lambda}^{1/2}(\tilde{\lambda} + \utilde{\lambda} - \bar{\lambda}) - \utilde{\lambda}^{1/2}\bar{\lambda}\Big) < 0.$$

Note that the negativity of above quantity comes from the stability condition, $\tilde{\lambda} + \utilde{\lambda} > \bar{\lambda}$.

(b) If $\utilde{\lambda} + \tilde{\lambda} > \underline{\lambda} + \bar{\lambda}$ then we have $\tilde{q} = 1 + \tilde{\beta} - \tilde{\alpha}\underline{\lambda} - (\tilde{\beta})^{1/2}$ and $\tilde{q}_G = 1 - (2\underline{\lambda})/(\utilde{\lambda} + \tilde{\lambda})$. After some simplifications, we see that $\tilde{q} < \tilde{q}_G$ boils down to the inequality

$$-(\utilde{\lambda} + \tilde{\lambda})\utilde{\lambda}^{1/2}\tilde{\lambda}^{1/2} + 2\underline{\lambda}\utilde{\lambda}^{1/2}\tilde{\lambda}^{1/2} - \underline{\lambda}(\utilde{\lambda} + \tilde{\lambda}) < 0,$$

or equivalently $-(\underset{\sim}{\lambda}+\widetilde{\lambda}-2\underline{\lambda})\underset{\sim}{\lambda}^{1/2}\widetilde{\lambda}^{1/2}-\underline{\lambda}(\underset{\sim}{\lambda}+\widetilde{\lambda})<0$, which holds by noting that $\underset{\sim}{\lambda}+\widetilde{\lambda}>\underline{\lambda}+\overline{\lambda}>2\underline{\lambda}$.

(c) for the case $\underset{\sim}{\lambda}+\widetilde{\lambda}=\underset{\sim}{\lambda}+\overline{\lambda}$, we have $\widetilde{q}=1+\widetilde{\beta}-\widetilde{\alpha}\underline{\lambda}-(\widetilde{\beta})^{1/2}$ and $\widetilde{q}_G=(\overline{\lambda}-\underline{\lambda})/(\underline{\lambda}+\overline{\lambda})$, which coincides with the optimal convergence factor of unperturbed gradient method. After some rearrangements we notice that $\widetilde{q}<\widetilde{q}_G$ reduces to checking that $(\widetilde{\lambda}^{1/2}-\underset{\sim}{\lambda}^{1/2})(\overline{\lambda}-\underline{\lambda})<(\underset{\sim}{\lambda}^{1/2}+\widetilde{\lambda}^{1/2})(\underset{\sim}{\lambda}+\widetilde{\lambda})$, which holds since $\widetilde{\lambda}^{1/2}-\underset{\sim}{\lambda}^{1/2}<\underset{\sim}{\lambda}^{1/2}+\widetilde{\lambda}^{1/2}$ and $\overline{\lambda}-\underline{\lambda}<\overline{\lambda}+\underline{\lambda}=\underset{\sim}{\lambda}+\widetilde{\lambda}$.

$\mathcal{Q}_3$: if $(\underset{\sim}{\varepsilon},\widetilde{\varepsilon})\in\mathcal{Q}_3$ we have $0<\underset{\sim}{\lambda}<\underline{\lambda}$ and $\widetilde{\lambda}<\overline{\lambda}$. Combined with the stability assumption $\widetilde{\lambda}+\underset{\sim}{\lambda}>\overline{\lambda}$, one can verify that the convergence factors of the two perturbed iterations are $\widetilde{q}_G=(2\overline{\lambda})/(\underset{\sim}{\lambda}+\widetilde{\lambda})-1$ and $\widetilde{q}=\widetilde{\alpha}\overline{\lambda}-\widetilde{\beta}-1-(\widetilde{\beta})^{1/2}$, respectively. The fact that $\widetilde{q}<\widetilde{q}_G$ was proven in step (a) of the analysis of $\mathcal{Q}_2$.

$\mathcal{Q}_4$: if $(\underset{\sim}{\varepsilon},\widetilde{\varepsilon})\in\mathcal{Q}_4$ then, (4.17) implies that $\widetilde{q}=\widetilde{\beta}^{1/2}$. On the other hand, for this region, (4.16) yields $\widetilde{q}_G=(\overline{\lambda}-\underline{\lambda})/(\underline{\lambda}+\overline{\lambda})$. To conclude, we need to verify that there exists $\widetilde{\lambda}$ and $\underset{\sim}{\lambda}$ such that $\widetilde{q}>\widetilde{q}_G$, i.e., $(\widetilde{\lambda}^{1/2}-\underset{\sim}{\lambda}^{1/2})/(\widetilde{\lambda}^{1/2}+\underset{\sim}{\lambda}^{1/2})>(\overline{\lambda}-\underline{\lambda})/(\underline{\lambda}+\overline{\lambda})$. We do so by multiplying both sides with $(\underline{\lambda}+\overline{\lambda})(\widetilde{\lambda}^{1/2}+\underset{\sim}{\lambda}^{1/2})$ and simplifying the result to find that the inequality holds if $\underline{\lambda}\widetilde{\lambda}^{1/2}>\overline{\lambda}\underset{\sim}{\lambda}^{1/2}$, or equivalently $\widetilde{\lambda}/\underset{\sim}{\lambda}>\overline{\lambda}^2/\underline{\lambda}^2$. The statement is proven.

4.C.9 Proof of Lemma 4.1

To prove (i), we exploit the equivalence of μ-strong convexity of $f(\cdot)$ and $1/\mu$-Lipschitz continuity of $\nabla f_\star$. Specially according to [105, Theorem 4.2.1], for nonzero $z_1,z_2\in\mathbb{R}^n$, Lipschitz continuity of $\nabla f_\star$ implies that

$$\langle\nabla f_\star(z_1)-\nabla f_\star(z_2),z_1-z_2\rangle\leq\frac{1}{\mu}\|z_1-z_2\|^2.$$

Now, for $-\nabla d(z)=-A\nabla f_\star(-A^\top z)+b$, change the right hand side of the former inequality to obtain

$$\langle-\nabla d(z_1)+\nabla d(z_2),z_1-z_2\rangle=\langle\nabla f_\star(-A^\top z_1)-\nabla f_\star(-A^\top z_2),-A^\top(z_1-z_2)\rangle.$$

In light of $1/\mu$-Lipschitzness of $\nabla f^\star$, we get

$$\begin{aligned}\langle\nabla f_\star(-A^\top z_1)-\nabla f_\star(-A^\top z_2),-A^\top(z_1-z_2)\rangle&\leq\frac{1}{\mu}\|A^\top(z_1-z_2)\|^2\\&\leq\frac{\lambda_n(AA^\top)}{\mu}\|z_1-z_2\|^2.\end{aligned}$$

(ii) According to [105, Theorem 4.2.2], If $\nabla f(\cdot)$ is L-Lipschitz continuous then $f_\star$ is $1/L$-strongly convex, i.e., for non-identical $z_1,z_2\in\mathbb{R}^n$ we have

$$\langle\nabla f_\star(z_1)-\nabla f_\star(z_2),z_1-z_2\rangle\geq\frac{1}{L}\|z_1-z_2\|^2.$$

One can manipulate above inequality as

$$\begin{aligned}
&\langle -\nabla d(z_1) + \nabla d(z_2), z_1 - z_2 \rangle \\
&= \langle \nabla f_\star(-A^\top z_1) - \nabla f_\star(-A^\top z_2), -A^\top (z_1 - z_2) \rangle \\
&\geq \frac{1}{L} \| - A^\top (z_1 - z_2)\|^2 \geq \frac{\lambda_1(AA^\top)}{L} \|z_1 - z_2\|^2,
\end{aligned}$$

where the last inequality holds since A is of full row rank.

4.C.10 Proof of Theorem 4.2

The result follows from Lemma 4.1 and Theorem 4.1 with $W = I$ and noting that

$$\frac{\lambda_1(AA^\top)}{L} I \preceq H \preceq \frac{\lambda_n(AA^\top)}{\mu} I.$$

4.C.11 Proof of Proposition 4.6

The iterations (4.25) and (4.27) are equivalent when

$$1 - \eta = -\beta, \qquad (1+\beta)I - \alpha W = \eta Q.$$

The first condition implies that $\eta^\star = 1 + \beta^\star$. Combining this expression with the second condition, we find

$$Q^\star = I - \frac{\alpha^\star}{1+\beta^\star} W^\star = I - \frac{2}{\underline{\lambda} + \overline{\lambda}} W^\star.$$

The proof is completed by noting that for the consensus case, $\underline{\lambda} = \lambda_2(W^\star)$ and $\overline{\lambda} = \lambda_n(W^\star)$.

4.C.12 Proof of Lemma 4.4

For the upper bound on $\lambda_n(RR^\top)$, we use a similar approach as [15, Lemma 3]. Specially, from [99, p.313], it yields

$$\lambda_n^2(RR^\top) = \|RR^\top\|^2 \leq \|RR^\top\|_\infty \|RR^\top\|_1 = \|RR^\top\|_\infty^2.$$

Hence,

$$\begin{aligned}
\lambda_n(RR^\top) &= \max_\ell \sum_{\ell'} [RR^\top]_{\ell\ell'} = \max_\ell \sum_{\ell'} \sum_s R_{\ell s} R_{\ell' s} \\
&\leq \max_\ell \sum_s R_{\ell s} \ell_{\max} \leq s_{\max} \ell_{\max}.
\end{aligned}$$

To find a lower bound on $\lambda_1(RR^\top)$, we consider the definition $\lambda_1(RR^\top) = \min_{\|x\|=1} \|R^\top x\|^2$. We have

$$[R^\top x]_s = \sum_{\ell=1}^{L} R^\top_{s\ell} x_\ell = \sum_{\ell=1}^{L} R_{\ell s} x_\ell.$$

According to Assumption 4.2, $R^\top$ has L independent rows that have only one non-zero (equal to 1) component. Hence,

$$\begin{aligned}\|R^\top x\|^2 &= \sum_{s=1}^{L} x_s^2 + \sum_{s=S-L+1}^{S} \left(\sum_{\ell=1}^{L} R_{\ell s} x_\ell\right)^2 \\ &= 1 + \sum_{s=S-L+1}^{n} \left(\sum_{\ell=1}^{L} R_{\ell s} x_\ell\right)^2 \geq 1,\end{aligned}$$

where the last equality is due to $\|x\| = 1$.

Chapter 5
Accelerating the ADMM algorithm: quadratic problems

IN the previous chapters, we mainly focused on gradient-like optimization algorithms and studied different techniques to speed up the algorithms. The aim of the next two chapters is to contribute to the understanding of the convergence properties of the ADMM method. We have chosen to focus on quadratic problems, since they allow for analytical tractability, yet have vast applications in estimation [106], multi-agent systems [107] and control [108]. Furthermore, many complex problems can be reformulated as or approximated by QPs [4], and optimal ADMM parameters for QPs can be used as a benchmark for more complex ADMM sub-problems e.g., ℓ_1-regularized problems [45].

In this chapter, we derive the algorithm parameters that minimize the convergence factor of the ADMM iterations for two classes of quadratic optimization problems: ℓ_2-regularized quadratic minimization and quadratic programming with linear inequality constraints. In both cases, we establish linear convergence rates and develop techniques to minimize the convergence factors of the ADMM iterates. These techniques allow us to give explicit expressions for the optimal algorithm parameters and the associated convergence factors. We also study over-relaxed ADMM iterations and demonstrate how to jointly choose the ADMM parameter and the over-relaxation parameter to improve the convergence times even further.

The chapter is organized as follows. Section 5.1 reviews the related literature. Section 5.2 studies ℓ_2-regularized quadratic programming and gives explicit expressions for the jointly optimal step-size and acceleration parameter that minimize the convergence factor. We then shift our focus to the quadratic programming with linear inequality constraints and derive the optimal step-sizes for such problems in Section 5.3. We also consider two acceleration techniques and discuss inexpensive ways to improve the speed of convergence. Our results are illustrated through numerical examples in Section 5.4. In Section 5.4 we perform an extensive Model Predictive Control (MPC) case study and evaluate the performance of ADMM with the proposed parameter selection rules. A comparison with an accelerated ADMM method from the literature is also performed. Finally, Section 5.5

summarizes the results of the chapter.

5.1 Related work

ADMM is a powerful algorithm for solving structured convex optimization problems. While the ADMM method was introduced for optimization in the 1970's, its origins can be traced back to techniques for solving elliptic and parabolic partial difference equations developed in the 1950's (see [45] and references therein). ADMM enjoys the strong convergence properties of the method of multipliers and the decomposability property of dual ascent, and is particularly useful for solving optimization problems that are too large to be handled by generic optimization solvers. The method has found a large number of applications in diverse areas such as compressed sensing [109], regularized estimation [110], image processing [111], machine learning [112], and resource allocation in wireless networks [113]. This broad range of applications has triggered a strong recent interest in developing a better understanding of the theoretical properties of ADMM [57, 114, 115].

Mathematical decomposition is a classical approach for parallelizing numerical optimization algorithms. If the decision problem has a favorable structure, decomposition techniques such as primal and dual decomposition allow to distribute the computations on multiple processors [116, 95]. The processors are coordinated towards optimality by solving a suitable master problem, typically using gradient or sub-gradient techniques. If problem parameters such as Lipschitz constants and convexity parameters of the cost function are available, the optimal step-size parameters and associated convergence rates are well-known [50]. A drawback of the gradient method is that it is sensitive to the choice of the step-size, even to the point where poor parameter selection can lead to algorithm divergence. In contrast, the ADMM technique is surprisingly robust to poorly selected algorithm parameters: under mild conditions, the method is guaranteed to converge for all positive values of its single parameter. Recently, an intense research effort has been devoted to establishing the rate of convergence of the ADMM method. It is now known that if the objective functions are strongly convex and have Lipschitz-continuous gradients, then the iterates produced by the ADMM algorithm converge linearly to the optimum in a certain distance metric e.g., [57]. The application of ADMM to quadratic problems was considered in [115] and it was conjectured that the iterates converge linearly in the neighborhood of the optimal solution. It is important to stress that even when the ADMM method has linear convergence rate, the number of iterations ensuring a desired accuracy, i.e., the convergence time, is heavily affected by the choice of the algorithm parameter. We will show that a poor parameter selection can result in arbitrarily large convergence times for the ADMM algorithm.

To the best of our knowledge, this is one of the first works that addresses the problem of optimal parameter selection for ADMM. A few recent papers have focused on the optimal parameter selection of ADMM algorithm for some variations of distributed convex programming subject to linear equality constraints e.g., [117, 118]. Moreover, [57] has recommended certain choices of the step-size parameter for the case when objective functions are strongly convex, have Lipschitz-continuous gradients, and the constraint matrices satisfy some full-rank assumptions. Compared to their results, we offer tighter bounds to compute the step-size parameter for quadratic programing problems.

In the rest of this chapter, we will consider the traditional ADMM iterations (2.15) and the relaxed version (2.19) for different classes of quadratic problems, and derive explicit expressions for the step-size ρ and the relaxation parameter α that minimize the convergence factors.

5.2 Optimal convergence factor for ℓ_2-regularized quadratic minimization

Regularized estimation problems

$$\text{minimize} \quad f(x) + \frac{\delta}{2}\|x\|_p^q,$$

where $\delta \in \mathbb{R}_{++}$ are abound in statistics, machine learning, and control. In particular, ℓ_1-regularized estimation where $f(x)$ is quadratic and $p = q = 1$, and *sum of norms* regularization where $f(x)$ is quadratic, $p = 2$, and $q = 1$, have recently received significant attention [119]. In this section we will focus on ℓ_2-regularized estimation, where $f(x)$ is quadratic and $p = q = 2$, i.e.,

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & \frac{1}{2}x^\top Qx + q^\top x + \frac{\delta}{2}\|z\|^2 \\ \text{subject to} & x - z = 0, \end{array} \tag{5.1}$$

for $Q \in \mathcal{S}_{++}^n$, $x, q, z \in \mathbb{R}^n$ and constant regularization parameter $\delta \in \mathbb{R}_{++}$. While these problems can be solved explicitly and do not motivate the ADMM machinery per se, they provide insight into the step-size selection for ADMM and allow us to compare the performance of an optimally tuned ADMM to direct alternatives (see Section 5.4).

5.2.1 Standard ADMM iterations

The standard ADMM iterations are given by

$$\begin{aligned} x^{(k+1)} &= (Q + \rho I)^{-1}(\rho z^{(k)} - y^{(k)} - q), \\ z^{(k+1)} &= \frac{y^{(k)} + \rho x^{(k+1)}}{\delta + \rho}, \\ y^{(k+1)} &= y^{(k)} + \rho(x^{(k+1)} - z^{(k+1)}). \end{aligned} \tag{5.2}$$

The z-update implies that $y^{(k)} = (\delta+\rho)z^{(k+1)} - \rho x^{(k+1)}$, so the y-update can be re-written as

$$y^{(k+1)} = (\delta + \rho)z^{(k+1)} - \rho x^{(k+1)} + \rho(x^{(k+1)} - z^{(k+1)}) = \delta z^{(k+1)}.$$

Hence, to study the convergence of (5.2) one can investigate how the errors associated with $x^{(k)}$ or $z^{(k)}$ vanish. Inserting the x-update into the z-update and using the fact that $y^{(k)} =$

$\delta z^{(k)}$, we find

$$\begin{aligned} z^{(k+1)} &= \underbrace{\frac{1}{\delta+\rho}\left(\delta I + \rho(\rho-\delta)\left(Q+\rho I\right)^{-1}\right)}_{E} z^{(k)} \\ &\quad - \frac{\rho}{\delta+\rho}(Q+\rho I)^{-1}q. \end{aligned} \tag{5.3}$$

Let $z^\star$ be a fixed-point of (5.3), i.e. $z^\star = Ez^\star - \dfrac{\rho(Q+\rho I)^{-1}}{\delta+\rho}q$. The dual error $e^{(k+1)} \triangleq z^{(k+1)} - z^\star$ then evolves as

$$e^{(k+1)} = Ee^{(k)}. \tag{5.4}$$

A direct analysis of the error dynamics (5.4) allows us to characterize the convergence of (5.2):

Theorem 5.1

For all values of the step-size $\rho > 0$ and regularization parameter $\delta > 0$, both $x^{(k)}$ and $z^{(k)}$ in the ADMM iterations (5.2) converge to $x^\star = z^\star$, the solution of optimization problem (5.1). Moreover, $z^{(k+1)} - z^\star$ converges at linear rate $\zeta \in (0,1)$ for all $k \geq 0$ where

$$\zeta \triangleq \limsup_{k\to\infty} \frac{\|z^{(k+1)} - z^\star\|}{\|z^{(k)} - z^\star\|}.$$

The pair of the optimal constant step-size $\rho^\star$ and convergence factor $\zeta^\star$ are given as

$$\rho^\star = \begin{cases} \sqrt{\delta\lambda_1(Q)} & \text{if } \delta < \lambda_1(Q), \\ \sqrt{\delta\lambda_n(Q)} & \text{if } \delta > \lambda_n(Q), \\ \delta & \text{otherwise.} \end{cases} \quad \zeta^\star = \begin{cases} \left(1 + \dfrac{\delta+\lambda_1(Q)}{2\sqrt{\delta\lambda_1(Q)}}\right)^{-1} & \text{if } \delta < \lambda_1(Q), \\ \left(1 + \dfrac{\delta+\lambda_n(Q)}{2\sqrt{\delta\lambda_n(Q)}}\right)^{-1} & \text{if } \delta > \lambda_n(Q), \\ \dfrac{1}{2} & \text{otherwise.} \end{cases} \tag{5.5}$$

Proof. See Appendix 5.A for all proofs of this chapter. □

Corollary 5.1

Consider the error dynamics described by (5.4) and E in (5.3). For $\rho = \delta$,

$$\lambda_i(E) = 1/2, \qquad i = 1, \dots, n,$$

and the convergence factor of the error dynamics (5.4) is independent of Q.

Remark 5.1 Note that the convergence factors in Theorem 5.1 and Corollary 5.1 are guaranteed for all initial values, and that iterates generated from specific initial values might converge even faster. Furthermore, the results focus on the dual error. For example, in Algorithm (5.2) with $\rho = \delta$ and initial condition $z^{(0)} = 0$, $y^{(0)} = 0$, the x-iterates converge in one iteration since $x^1 = -(Q + \delta I)^{-1}q = x^\star$. However, the constraint in (5.1) is not satisfied and a straightforward calculation shows that $e^{(k+1)} = 1/2e^{(k)}$. Thus, although $x^{(k)} = x^\star$ for $k \geq 1$, the dual residual $\|e^{(k)}\| = \|z^{(k)} - z^\star\|$ decays linearly with a factor of $1/2$.

Remark 5.2 The analysis above also applies to the more general case with cost function $\frac{1}{2}\bar{x}^\top \bar{Q}\bar{x} + \bar{q}^\top \bar{x} + \frac{\delta}{2}\bar{z}^\top \bar{P}\bar{z}$ where $\bar{P} \in \mathcal{S}^n_{++}$. A change of variables $x = \bar{P}^{1/2}\bar{x}$, $z = \bar{P}^{1/2}\bar{z}$, $q = \bar{P}^{-1/2}\bar{q}$, and $Q = \bar{P}^{-1/2}\bar{Q}\bar{P}^{-1/2}$ is then applied to transform the problem into the form (5.1).

5.2.2 Over-relaxed ADMM iterations

Recall from Section 2.2.3 that the over-relaxed ADMM iterations for (5.1) can be found by replacing $x^{(k+1)}$ by $\alpha x^{(k+1)} + (1-\alpha)z^{(k)}$ in the z- and y- updates of (5.2). The resulting iterations take the form

$$\begin{aligned}
x^{(k+1)} &= (Q + \rho I)^{-1}(\rho z^{(k)} - y^{(k)} - q),\\
z^{(k+1)} &= \frac{y^{(k)} + \rho(\alpha x^{(k+1)} + (1-\alpha)z^{(k)})}{\delta + \rho},\\
y^{(k+1)} &= y^{(k)} + \rho\left(\alpha(x^{(k+1)} - z^{(k+1)}) + (1-\alpha)\left(z^{(k)} - z^{(k+1)}\right)\right).
\end{aligned} \tag{5.6}$$

The next result demonstrates that in a certain range of α it is possible to obtain a guaranteed improvement of the convergence factor compared to the classical iterations (5.2).

Theorem 5.2
Consider the ℓ_2-regularized quadratic minimization problem (5.1) and its associated over-relaxed ADMM iterations (5.6). For all positive step-sizes $\rho > 0$ and all relaxation parameters

$$\alpha \in \left(0, 2\min_i\left\{\frac{(\lambda_i(Q) + \rho)(\rho + \delta)}{\rho\delta + \rho\lambda_i(Q)}\right\}\right),$$

the iterates $x^{(k)}$ and $z^{(k)}$ converge to the solution of (5.1). Moreover, $z^{(k+1)} - z^\star$ converges at linear rate $\zeta_R \in [0, 1)$ for all $k \geq 0$. For a given step-size ρ, the convergence factor $\zeta_R < 1$ is strictly smaller than that of the classical ADMM algorithm (5.2) if

$$\alpha \in \left(1, 2\min_i\left\{\frac{(\lambda_i(Q) + \rho)(\rho + \delta)}{\rho\delta + \rho\lambda_i(Q)}\right\}\right).$$

The jointly optimal step-size, relaxation parameter, and the convergence factor $(\rho^\star, \alpha^\star, \zeta_R^\star)$ are given by

$$\rho^\star = \delta, \quad \alpha^\star = 2, \quad \zeta_R^\star = 0. \tag{5.7}$$

With these parameters, the ADMM iterations converge in one iteration.

Remark 5.3 The upper bound on α which ensures faster convergence of the over-relaxed ADMM iterations (5.6) compared to (5.2) depends on the eigenvalues of Q, $\lambda_i(Q)$, which might be unknown. However, since $(\rho+\delta)(\rho+\lambda_i(Q)) > \rho(\lambda_i(Q)+\delta)$ the over-relaxed iterations are guaranteed to converge faster for all $\alpha \in (1,2]$, independently of Q.

5.3 Optimal convergence factor for quadratic programming

In this section, we consider a QP problem of the form

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}x^\top Qx + q^\top x \\ \text{subject to} & Ax \le c, \end{array} \tag{5.8}$$

where $Q \in \mathcal{S}^n_{++}$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ is full rank and $c \in \mathbb{R}^m$.

5.3.1 Standard ADMM iterations

The QP-problem (5.8) can be put on ADMM standard form (2.14) by introducing a slack vector z and putting an infinite penalty on negative components of z, *i.e.*

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top Qx + q^\top x + \mathcal{I}_+(z) \\ \text{subject to} & Ax - c + z = 0, \end{array} \tag{5.9}$$

where $\mathcal{I}_+(\cdot)$ is the indicator function of the positive orthant defined as the following

$$\mathcal{I}_+(z) = \begin{cases} 0 & \text{for } z \ge 0, \\ +\infty & \text{otherwise.} \end{cases}$$

The associated *augmented Lagrangian* is

$$L_\rho(x,z,u) = \frac{1}{2}x^\top Qx + q^\top x + \mathcal{I}_+(z) + \frac{\rho}{2}\|Ax - c + z + u\|^2,$$

where $u = y/\rho$, which leads to the scaled ADMM iterations

$$\begin{array}{ll} x^{(k+1)} & = -(Q+\rho A^\top A)^{-1}[q + \rho A^\top(z^{(k)} + u^{(k)} - c)], \\ z^{(k+1)} & = \max\{0, -Ax^{(k+1)} - u^{(k)} + c\}, \\ u^{(k+1)} & = u^{(k)} + Ax^{(k+1)} - c + z^{(k+1)}. \end{array} \tag{5.10}$$

To study the convergence of (5.10) we rewrite it in an equivalent form with linear time-varying matrix operators. To this end, we introduce a vector of indicator variables $d^{(k)} \in \{0,1\}^n$ such that $d_i^{(k)} = 0$ if $u_i^{(k)} = 0$ and $d_i^{(k)} = 1$ if $u_i^{(k)} \neq 0$. From the z- and u- updates in (5.10), one observes that $z_i^{(k)} \neq 0 \to u_i^{(k)} = 0$, i.e., $u_i^{(k)} \neq 0 \to z_i^{(k)} = 0$. Hence, $d_i^{(k)} = 1$ means that at the current iterate, the slack variable z_i in (5.9) equals zero; i.e., the i-th inequality constraint in (5.8) is active. We also introduce the variable vector $v^{(k)} \triangleq$

$z^{(k)} + u^{(k)}$ and let $D^{(k)} = \operatorname{diag}(d^{(k)})$ so that $D^{(k)}v^{(k)} = u^{(k)}$ and $(I - D^{(k)})v^{(k)} = z^{(k)}$. Now, the second and third steps of (5.10) imply that

$$v^{(k+1)} = \left|Ax^{(k+1)} + u^{(k)} - c\right| = F^{(k+1)}(Ax^{(k+1)} + D^{(k)}v^{(k)} - c),$$

where $F^{(k+1)} \triangleq \operatorname{diag}\big(\operatorname{sign}(Ax^{(k+1)} + D^{(k)}v^{(k)} - c)\big)$ and $\operatorname{sign}(\cdot)$ returns the signs of the elements of its vector argument. Hence, (5.10) becomes

$$\begin{aligned} x^{(k+1)} &= -(Q + \rho A^\top A)^{-1}[q + \rho A^\top(v^{(k)} - c)], \\ v^{(k+1)} &= \left|Ax^{(k+1)} + D^{(k)}v^{(k)} - c\right| = F^{(k+1)}(Ax^{(k+1)} + D^{(k)}v^{(k)} - c), \\ D^{(k+1)} &= \frac{1}{2}(I + F^{(k+1)}), \end{aligned} \tag{5.11}$$

where the $D^{(k+1)}$-update follows from the observation that

$$(D_{ii}^{(k+1)}, F_{ii}^{(k+1)}) = \begin{cases} (0, -1) & \text{if } v_i^{(k+1)} = -(Ax_i^{(k+1)} + u_i^{(k)} - c) \\ (1, 1) & \text{if } v_i^{(k+1)} = Ax_i^{(k+1)} + u_i^{(k)} - c \end{cases}$$

Since the $v^{(k)}$-iterations will be central in our analysis, we will develop them further. Inserting the expression for $x^{(k+1)}$ from the first equation of (5.11) into the second, we find

$$\begin{aligned} v^{(k+1)} = {} & F^{(k+1)}\Big(\Big(d^{(k)} - A(Q/\rho + A^\top A)^{-1}A^\top\Big)\, v^{(k)}\Big) \\ & - F^{(k+1)}\Big(A(Q + \rho A^\top A)^{-1}(q - \rho A^\top c) + c\Big). \end{aligned} \tag{5.12}$$

Noting that $D^{(k)} = \frac{1}{2}(I + F^{(k)})$ and introducing

$$M \triangleq A(Q/\rho + A^\top A)^{-1}A^\top, \tag{5.13}$$

we obtain

$$F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)} = \left(\frac{I}{2} - M\right)(v^{(k)} - v^{(k-1)}) + \frac{1}{2}\left(F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\right). \tag{5.14}$$

We now relate $v^{(k)}$ and $F^{(k)}v^{(k)}$ to the primal and dual residuals, $r^{(k)}$ and $s^{(k)}$, defined in (2.17) and (2.18):

Proposition 5.1

Consider $r^{(k)}$ and $s^{(k)}$ the primal and dual residuals of the QP-ADMM algorithm (5.10) and auxiliary variables $v^{(k)}$ and $F^{(k)}$. The following relations hold

$$F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)} = r^{(k+1)} - \frac{1}{\rho}A^\dagger s^{(k+1)} - \Pi_{\mathcal{N}(A^\top)}(z^{(k+1)} - z^{(k)}), \tag{5.15}$$

$$v^{(k+1)} - v^{(k)} = r^{(k+1)} + \frac{1}{\rho}A^\dagger s^{(k+1)} + \Pi_{\mathcal{N}(A^\top)}(z^{(k+1)} - z^{(k)}), \tag{5.16}$$

$$\|r^{(k+1)}\| \le \|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\|, \tag{5.17}$$

$$\|s^{(k+1)}\| \le \rho\|A\|\|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\|, \tag{5.18}$$

where

(i) $A^\dagger = A(A^\top A)^{-1}$ and $\Pi_{\mathcal{N}(A^\top)} = I - A(A^\top A)^{-1}A^\top$, if A has full column-rank;

(ii) $A^\dagger = (AA^\top)^{-1}A$ and $\Pi_{\mathcal{N}(A^\top)} = 0$, if A has full row-rank;

(iii) $A^\dagger = A^{-1}$ and $\Pi_{\mathcal{N}(A^\top)} = 0$, if A is invertible.

The next theorem guarantees that (5.14) converges linearly to zero in the auxiliary residuals (5.15) which implies R-linear convergence of the ADMM algorithm (5.10) in terms of the primal and dual residuals. The optimal step-size $\rho^\star$ and the smallest achievable convergence factor are characterized immediately afterwards.

Theorem 5.3

Consider the QP (5.8) and the corresponding ADMM iterations (5.10). For all values of the step-size $\rho \in \mathbb{R}_{++}$ the residual $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ converges to zero at linear rate. Furthermore, $r^{(k)}$ and $s^{(k)}$, the primal and dual residuals of (5.10), converge R-linearly to zero.

Theorem 5.4

Consider the QP problem (5.8) and the corresponding ADMM iterations (5.10). If the constraint matrix A is either full row-rank or invertible then the optimal step-size and convergence factor for the $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ residuals are

$$\begin{aligned} \rho^\star &= \left(\sqrt{\lambda_1(AQ^{-1}A^\top)\lambda_n(AQ^{-1}A^\top)}\right)^{-1}, \\ \zeta^\star &= \frac{\lambda_n(AQ^{-1}A^\top)}{\lambda_n(AQ^{-1}A^\top) + \sqrt{\lambda_1(AQ^{-1}A^\top)\lambda_n(AQ^{-1}A^\top)}}. \end{aligned} \tag{5.19}$$

Although the convergence result of Theorem 5.3 holds for all QPs of the form (5.8), optimality of the step-size choice proposed in Theorem 5.4 is only established for problems where the constraint matrix A has full row-rank or it is invertible. However, as shown next, the convergence factor can be arbitrarily close to 1 when rows of A are linearly dependent.

Theorem 5.5

Define variables

$$\epsilon_k \triangleq \frac{\|M(v^{(k)} - v^{(k-1)})\|}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|}, \qquad \delta_k \triangleq \frac{\|D^{(k)}v^{(k)} - D^{(k-1)}v^{(k-1)}\|}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|},$$

$$\tilde{\zeta}(\rho) \triangleq \max_{i:\ \lambda_i(AQ^{-1}A^\top)>0} \left\{ \left| \frac{\rho\lambda_i(AQ^{-1}A^\top)}{1+\rho\lambda_i(AQ^{-1}A^\top)} - \frac{1}{2} \right| + \frac{1}{2} \right\},$$

and $\underline{\zeta}^{(k)} \triangleq |\delta_k - \epsilon_k|$.

The convergence factor ζ of the residual $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ is lower bounded by

$$\underline{\zeta} \triangleq \max_{k} \ \underline{\zeta}^{(k)} < 1. \tag{5.20}$$

Furthermore, given an arbitrarily small $\xi \in (0, \ \frac{1}{2})$ and $\rho > 0$, we have the following results:

(i) the inequality $\underline{\zeta} < \tilde{\zeta}(\rho) < 1$ holds for all $\delta_k \in [0, \ 1]$ if and only if the nullity of A is zero;

(ii) when the nullity of A is nonzero and $\epsilon_k \geq 1 - \xi$, it holds that $\underline{\zeta} \leq \tilde{\zeta}(\rho) + \sqrt{\dfrac{\xi}{2}}$;

(iii) when the nullity of A is nonzero, $\delta_k \geq 1 - \xi$, and

$$\frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|}{\|v^{(k)} - v^{(k-1)}\|} \geq \sqrt{1 - \xi^2/\|M\|^2},$$

it follows that $\underline{\zeta} \geq 1 - 2\xi$.

The previous result establishes that slow convergence can occur locally for any value of ρ when the nullity of A is nonzero and ξ is small. However, as section (ii) of Theorem 5.5 suggests, in these cases, (5.19) can still work as a heuristic to reduce the convergence time if $\lambda_1(AQ^{-1}A^\top)$ is taken as the smallest nonzero eigenvalue of $AQ^{-1}A^\top$. In Section 5.4, we show numerically that this heuristic performs well with different problem setups.

5.3.2 Over-relaxed ADMM iterations

Consider the relaxation of (5.10) obtained by replacing $Ax^{(k+1)}$ in the z- and u-updates with $\alpha Ax^{(k+1)} - (1-\alpha)(z^{(k)} - c)$. The corresponding relaxed iterations read

$$\begin{aligned}
x^{(k+1)} &= -(Q + \rho A^\top A)^{-1}\Big[q + \rho A^\top (z^{(k)} + u^{(k)} - c)\Big], \\
z^{(k+1)} &= \max\Big\{\ 0, -\alpha(Ax^{(k+1)} - c) + (1-\alpha)z^{(k)} - u^{(k)}\Big\}, \\
u^{(k+1)} &= u^{(k)} + \alpha(Ax^{(k+1)} + z^{(k+1)} - c) + (1-\alpha)(z^{(k+1)} - z^{(k)}).
\end{aligned} \tag{5.21}$$

In next, we study convergence and optimality properties of these iterations. We observe:

Lemma 5.1

Any fixed-point of (5.21) corresponds to a global optimum of (5.9).

Like the analysis of (5.10), introduce $v^{(k)} = z^{(k)} + u^{(k)}$ and $d^{(k)} \in \mathbb{R}^n$ with $d_i^{(k)} = 0$ if $u_i^{(k)} = 0$ and $d_i^{(k)} = 1$ otherwise. Adding the second and the third step of (5.21) yields

$v^{(k+1)} = \big|\alpha(Ax^{(k+1)} - c) - (1-\alpha)z^{(k)} + u^{(k)}\big|$. Moreover, $D^{(k)} = \operatorname{diag}(d^{(k)})$ satisfies $D^{(k)}v^{(k)} = u^{(k)}$ and $(I - D^{(k)})v^{(k)} = z^{(k)}$, so (5.21) can be rewritten as

$$\begin{aligned}
x^{(k+1)} &= -(Q+\rho A^\top A)^{-1}\left[q + \rho A^\top (v^{(k)} - c)\right],\\
v^{(k+1)} &= F^{(k+1)}\Big(\alpha\left(Ax^{(k+1)} + D^{(k)}v^{(k)} - c\right)\Big) - F^{(k+1)}\Big((1-\alpha)(I - 2D^{(k)})v^{(k)}\Big),\\
D^{(k+1)} &= \frac{1}{2}(I + F^{(k+1)}),
\end{aligned} \tag{5.22}$$

where

$$F^{(k+1)} \triangleq \operatorname{diag}\Big(\operatorname{sign}\Big(\alpha(Ax^{(k+1)} + D^{(k)}v^{(k)} - c) - (1-\alpha)(I - 2D^{(k)})v^{(k)}\Big)\Big).$$

Defining $M \triangleq A(Q/\rho + A^\top A)^{-1}A^\top$ and substituting the expression for $x^{(k+1)}$ in (5.22) into the expression for $v^{(k+1)}$ yields

$$\begin{aligned}
v^{(k+1)} = F^{(k+1)}&\Big(\Big(-\alpha M + (2-\alpha)D^{(k)} - (1-\alpha)I\Big)\,v^{(k)}\Big)\\
&- F^{(k+1)}\Big(\alpha A(Q+\rho A^\top A)^{-1}(q - \rho A^\top c) + \alpha c\Big).
\end{aligned} \tag{5.23}$$

As in the previous section, we replace $D^{(k)}$ by $\frac{1}{2}(I + F^{(k)})$ in (5.23) and form $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$:

$$\begin{aligned}
F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)} = &\frac{\alpha}{2}\,(I - 2M)\,\Big(v^{(k)} - v^{(k-1)}\Big)\\
&+ (1 - \frac{\alpha}{2})\,\Big(F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\Big).
\end{aligned} \tag{5.24}$$

The next theorem characterizes the convergence rate of the relaxed ADMM iterations.

Theorem 5.6

Consider the QP (5.8) and the corresponding relaxed ADMM iterations (5.21). If

$$\rho \in \mathbb{R}_{++}, \quad \alpha \in (0, 2], \tag{5.25}$$

then the equivalent fixed point iteration (5.24) converges linearly in terms of $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ residual. Moreover, $r^{(k)}$ and $s^{(k)}$, the primal and dual residuals of (5.21), converge R-linearly to zero.

Next, we restrict our attention to the case where A is either invertible or full row-rank to be able to derive the jointly optimal step-size and over-relaxation parameter, as well as an explicit expression for the associated convergence factor. The result shows that the over-relaxed ADMM iterates can yield a significant speedup compared to the standard ADMM iterations.

Theorem 5.7

Consider the QP (5.8) and the corresponding relaxed ADMM iterations (5.21). If the constraint matrix A is of full row-rank or invertible then the joint optimal step-size, relaxation parameter and the convergence factor with respect to the $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ residual are

$$\begin{aligned}\rho^\star &= \left(\sqrt{\lambda_1(AQ^{-1}A^\top)\,\lambda_n(AQ^{-1}A^\top)}\right)^{-1}, \quad \alpha^\star = 2,\\ \zeta_R^\star &= \frac{\lambda_n(AQ^{-1}A^\top) - \sqrt{\lambda_1(AQ^{-1}A^\top)\,\lambda_n(AQ^{-1}A^\top)}}{\lambda_n(AQ^{-1}A^\top) + \sqrt{\lambda_1(AQ^{-1}A^\top)\,\lambda_n(AQ^{-1}A^\top)}}.\end{aligned} \tag{5.26}$$

Moreover, for a given step-size ρ, when the iterations (5.24) are over-relaxed; i.e., $\alpha \in (1, 2]$ their iterates have a smaller convergence factor than that of (5.14).

5.3.3 Optimal constraint preconditioning

In this section, we consider another technique to improve the convergence of the ADMM method. The approach is based on the observation that the optimal convergence factors $\zeta^\star$ and $\zeta_R^\star$ from Theorem 5.4 and Theorem 5.7 are monotone increasing in the ratio $\lambda_n(AQ^{-1}A^\top)/\lambda_1(AQ^{-1}A^\top)$. This ratio can be decreased –without changing the complexity of the ADMM algorithm (5.10)– by scaling the equality constraint in (5.9) by a diagonal matrix $L \in \mathcal{S}^m_{++}$, i.e., replacing $Ax - c + z = 0$ by $L\,(Ax - c + z) = 0$. Let $\bar{A} \triangleq LA$, $\bar{z} \triangleq Lz$, and $\bar{c} \triangleq Lc$. The resulting scaled ADMM iterations are derived by replacing A, z, and c in (5.10) and (5.21) by the new variables $\bar{A}$, $\bar{z}$, and $\bar{c}$, respectively. Furthermore, the results of Theorem 5.4 and Theorem 5.7 can be applied to the scaled ADMM iterations in terms of new variables. Although these theorems only provide the optimal step-size parameters for the QP when the constraint matrices are invertible or have full row-rank, we use the expressions as heuristics when the constraint matrix has full column-rank. Hence, in the following we consider $\lambda_n(\bar{A}Q^{-1}\bar{A}^\top)$ and $\lambda_1(\bar{A}Q^{-1}\bar{A}^\top)$ to be the largest and smallest nonzero eigenvalues of $\bar{A}Q^{-1}\bar{A}^\top = LAQ^{-1}A^\top L$, respectively and minimize the ratio λ_n/λ_1 in order to minimize the convergence factors $\zeta^\star$ and $\zeta_R^\star$.

Theorem 5.8

Let $R_q R_q^\top = Q^{-1}$ be the Choleski factorization of Q^{-1} and $P \in \mathbb{R}^{n\times n-s}$ be a matrix whose columns are orthonormal vectors spanning $\mathrm{Im}(R_q^\top A^\top)$ with s being the dimension of $\mathcal{N}(A)$. Moreover, let $\lambda_n(LAQ^{-1}A^\top L)$ and $\lambda_1(LAQ^{-1}A^\top L)$ be the largest and smallest nonzero eigenvalues of $LAQ^{-1}A^\top L$. The diagonal scaling matrix $L^\star \in \mathcal{S}^m_{++}$ that minimizes the eigenvalue ratio $\lambda_n(LAQ^{-1}A^\top L)/\lambda_1(LAQ^{-1}A^\top L)$ can be obtained by solving the convex problem

$$\begin{aligned}\underset{t\in\mathbb{R},\, w\in\mathbb{R}^m}{\text{minimize}} \quad & t\\ \text{subject to} \quad & W = \mathrm{diag}(w),\ w > 0,\\ & tI - R_q^\top A^\top W A R_q \in \mathcal{S}^n_+,\\ & P^\top(R_q^\top A^\top W A R_q - I)P \in \mathcal{S}^{n-s}_+,\end{aligned} \tag{5.27}$$

and setting $L^\star = W^{\star^{1/2}}$.

So far, we characterized the convergence factor of the ADMM algorithm based on general properties of the sequence $\{F^{(k)}v^{(k)}\}$. However, if we a priori know which constraints will be active during the ADMM iterations, our parameter selection rules (5.19) and (5.26) may not be optimal. To illustrate this fact, we will now analyze the two extreme situations where no and all constraints are active in each iteration and derive the associated optimal ADMM parameters.

5.3.4 Special cases of quadratic programming

The first result deals with the case where the constraints of (5.8) are never active. This could happen, for example, if we use the constraints to impose upper and lower bounds on the decision variables, and use very loose bounds.

Proposition 5.2
Assume that $F^{(k+1)} = F^{(k)} = -I$ for all epochs $k \in \mathbb{R}_+$ in (5.11) and (5.22). Then the modified ADMM algorithm (5.24) attains its minimal convergence factor for the parameters

$$\alpha = 1, \quad \rho \to 0. \tag{5.28}$$

In this case (5.24) coincide with (5.14) and their convergence factor is minimized: $\zeta = \zeta_R \to 0$.

The next proposition addresses another extreme scenario when the ADMM iterates are operating on the active set of the quadratic program (5.8). This could happen, for example, if the constraints are defined in a way that, at optimality, all the inequality constraints are satisfied with equality.

Proposition 5.3
Suppose that $F^{(k+1)} = F^{(k)} = I$ for all $k \in \mathbb{R}_+$ in (5.11) and (5.22). Then the relaxed ADMM algorithm (5.24) attains its minimal convergence factor for the parameters

$$\alpha = 1, \quad \rho \to \infty. \tag{5.29}$$

In this case (5.24) coincides with (5.14) and their convergence factors are minimized: $\zeta = \zeta_R \to 0$.

It is worthwhile to mention that when (5.8) is defined so that its constraints are all active (inactive) then the $s^{(k)}$ ($r^{(k)}$) residuals of the ADMM algorithm remain zero for all $k \geq 2$ updates.

5.4 Numerical examples

In this section, we evaluate our parameter selection rules on numerical examples. First, we illustrate the convergence factor of ADMM and gradient algorithms for a family of ℓ_2-regularized quadratic problems. These examples demonstrate that the ADMM method converges faster than the gradient method for certain ranges of the regularization parameter δ, and slower for other values. Then, we consider QP problems and compare the performance of the over-relaxed ADMM algorithm with an alternative accelerated ADMM method presented in [120]. The two algorithms are also applied to a MPC benchmark where QP problems are solved repeatedly over time for fixed matrices Q and A but varying vectors q and b.

5.4.1 ℓ_2-regularized quadratic minimization via ADMM

We consider ℓ_2-regularized quadratic minimization problem (5.1) for a $Q \in \mathcal{S}^{100}_{++}$ with condition number 1.2×10^3 and for a range of regularization parameters δ. Figure 5.1 shows how the optimal convergence factor of ADMM depends on δ. The results are shown for two step-size rules: $\rho = \delta$ and $\rho = \rho^\star$ given in (5.5). For comparison, the gray and dashed-gray curves show the optimal convergence factor of the gradient method

$$x^{(k+1)} = x^{(k)} - \gamma(Qx^{(k)} + q + \delta x^{(k)}),$$

with step-size $\gamma < 2/(\lambda_n(Q) + \delta)$ and the multi-step gradient iterations on the form

$$x^{(k+1)} = x^{(k)} - a(Qx^{(k)} + q + \delta x^{(k)}) + b(x^{(k)} - x^{(k-1)}).$$

Recall that the latter algorithm is known as the heavy-ball method and significantly outperforms the standard gradient method on ill-conditioned problems [52]. The algorithm has two parameters: $a < 2(1 + b)/(\lambda_n(Q) + \delta)$, and $b \in [0, 1]$. For our problem, since the cost function is quadratic and its Hessian $\nabla^2 f(x) = Q + \delta I$ is bounded between $l = \lambda_1(Q) + \delta$ and $u = \lambda_n(Q) + \delta$, the optimal step-size for the gradient method is $\gamma^\star = 2/(l + u)$ and the optimal parameters for the heavy-ball method are $a^\star = 4/(\sqrt{l} + \sqrt{u})^2$, and $b^\star = (\sqrt{u} - \sqrt{l})^2/(\sqrt{l} + \sqrt{u})^2$ (see Chapter 2).

Figure 5.1 illustrates the convergence properties of the ADMM method under both step-size rules. The optimal step-size rule gives significant speedups of the ADMM for small or large values of the regularization parameter δ. This phenomena can be intuitively explained based on the interplay of the two parts of the objective function in (5.1). For extremely small values of δ, one sees that the x-th part of the objective is becoming dominant compared to z-th part. Consequently, using the optimal step-size in (5.5), z- is dictated to quickly follow the value of x-update. A similar reasoning holds when δ is large, in which the x- has to obey the z-update.

It is interesting to observe that ADMM outperforms the gradient and heavy-ball methods for small δ (an ill-conditioned problem), but actually performs worse as δ grows large (i.e., when the regularization makes the overall problem well-conditioned). It is noteworthy that the relaxed ADMM method solves the same problem in one step (convergence factor $\zeta_R^\star = 0$).

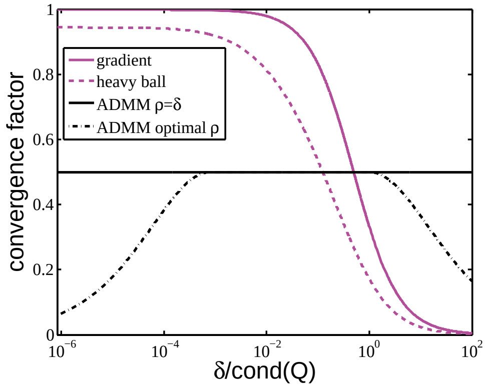


Figure 5.1: Convergence factor of the ADMM, gradient, and heavy-ball methods for ℓ_2 regularized minimization with fixed Q-matrix and different values of the regularization parameter δ.

5.4.2 Quadratic programming via ADMM

Next, we evaluate our step-size rules for ADMM-based quadratic programming and compare their performance with that of other accelerated ADMM variants from the literature.

Accelerated ADMM

One recent proposal for accelerating the ADMM-iterations is called *fast-ADMM* [120] and consists of the following iterations

$$\begin{aligned} x^{(k+1)} &= \underset{x}{\operatorname{argmin}}\, L_\rho(x, \hat{z}^{(k)}, \hat{u}^{(k)}), \\ z^{(k+1)} &= \underset{z}{\operatorname{argmin}}\, L_\rho(x^{(k+1)}, z, \hat{u}^{(k)}), \\ u^{(k+1)} &= \hat{u}^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c, \\ \hat{z}^{(k+1)} &= \alpha^{(k)} z^{(k+1)} + (1-\alpha^{(k)}) z^{(k)}, \\ \hat{u}^{(k+1)} &= \alpha^{(k)} u^{(k+1)} + (1-\alpha^{(k)}) u^{(k)}. \end{aligned} \tag{5.30}$$

The relaxation parameter $\alpha^{(k)}$ in the fast-ADMM method is defined based on the Nesterov's order-optimal method [50] combined with an innovative restart rule where $\alpha^{(k)}$ is given by

$$\alpha^{(k)} = \begin{cases} 1 + \dfrac{\beta^{(k)} - 1}{\beta^{(k+1)}} & \text{if } \dfrac{\max(\|r^{(k)}\|, \|s^{(k)}\|)}{\max(\|r^{(k-1)}\|, \|s^{(k-1)}\|)} < 1, \\ 1 & \text{otherwise,} \end{cases} \tag{5.31}$$

where

$$\beta^{(1)} = 1, \quad \beta^{(k+1)} = \frac{1 + \sqrt{1 + 4{\beta^{(k)}}^2}}{2}, \text{ for } k \in \mathbb{N}.$$

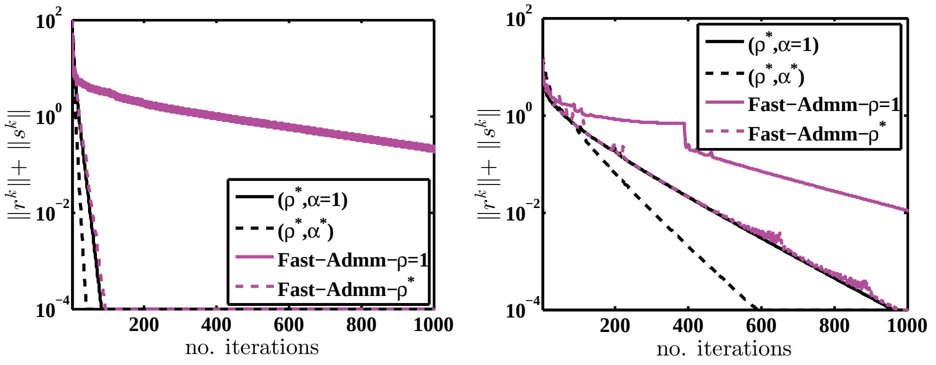


Figure 5.2: Convergence of primal plus dual residuals of four ADMM algorithms. The figure at left has $n = 100$ decision variables and $m = 50$ inequality constraints. The figure at right has $n = 100$ and $m = 200$ decision variables and inequality constraints, respectively.

The restart rule assures that (5.30) is updated in the descent direction with respect to the primal-dual residuals.

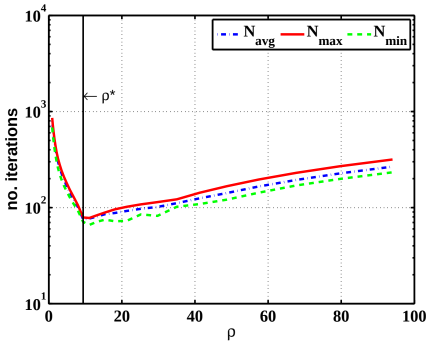
To compare the performance of the over-relaxed ADMM iterations with our proposed parameters to that of fast-ADMM, we conducted several numerical examples. For the first numerical comparison, we generated several instances of (5.8); Figure 5.2 shows the results for the two representative examples. In the first case, $A \in \mathbb{R}^{50\times100}$ and $Q \in \mathcal{S}^{100}_{++}$ with condition number 1.95×10^3; 32 constraints are active at the optimal solution. In the second case, $A \in \mathbb{R}^{200\times100}$ and $Q \in \mathcal{S}^{100}_{++}$, where the condition number of Q is 7.1×10^3. The polyhedral constraints correspond to random box-constraints, of which 66 are active at optimality. We evaluate for four algorithms: the ADMM iterates in (5.21) with and without over-relaxation and the corresponding tuning rules developed in this chapter, and the fast-ADMM iterates (5.30) with $\rho = 1$ as proposed by [120] and $\rho = \rho^\star$ of our result. The convergence of corresponding algorithms in terms of the summation of primal and dual residuals $\|r^{(k)}\| + \|s^{(k)}\|$ are depicted in Figure 5.2. The plots exhibit a significant improvement of our tuning rules compared to the fast-ADMM algorithm.

To the best of our knowledge, there are currently no results about optimal step-size parameters for the fast-ADMM method. However, based on our numerical investigations, we observed that the performance of fast-ADMM algorithm significantly improved by employing our optimal step-size $\rho^\star$ (as illustrated in 5.2). In the next section we perform another comparison between three algorithms, using the optimal ρ-value for fast-ADMM obtained by an extensive search.

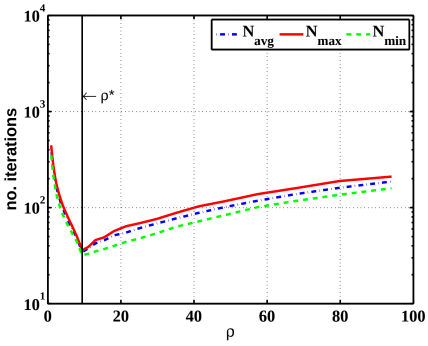
Model Predictive Control

Consider the discrete-time linear system

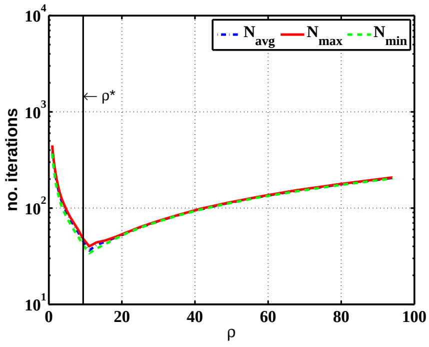
$$x_{t+1} = Hx_t + Ju_t + J_r r, \tag{5.32}$$



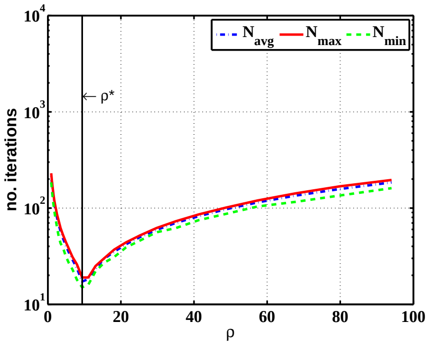
(a) $\alpha = 1$, $L = I$.



(b) $\alpha = 2$, $L = I$.



(c) $\alpha = 1$, $L = L^\star$.



(d) $\alpha = 2$, $L = L^\star$.

Figure 5.3: Number of iterations $k : \max\{\|r^{(k)}\|, \|s^{(k)}\|\} \leq 10^{-5}$ for ADMM applied to the MPC problem for different initial states x_0. The dashed green line denotes the minimum number of iterations taken over all the initial states, the dot-dashed blue line corresponds to the average, while the red solid line represents the maximum number of iterations.

where $t \geq 0$ is the time index, $x_t \in \mathbb{R}^{n_x}$ is the state, $u_t \in \mathbb{R}^{n_u}$ is the control input, $r \in \mathbb{R}^{n_r}$ is a constant reference signal, and $H \in \mathbb{R}^{n_x \times n_x}$, $J \in \mathbb{R}^{n_x \times n_u}$, and $J_r \in \mathbb{R}^{n_x \times n_r}$ are fixed matrices. Model predictive control aims at solving the following optimization problem

$$\begin{array}{ll}\underset{\{u_i\}_0^{N_p-1}}{\text{minimize}} & \frac{1}{2}\sum_{i=0}^{N_p-1}(x_i - x_r)^\top Q_x (x_i - x_r) \\ & +(u_i - u_r)^\top R(u_i - u_r) + (x_{N_p} - x_r)^\top Q_N (x_{N_p} - x_r) \\ \text{subject to} & x_{t+1} = Hx_t + Ju_t + J_r r \quad \forall t, \\ & x_t \in \mathcal{C}_x \quad \forall t, \\ & u_t \in \mathcal{C}_u \quad \forall t, \end{array} \tag{5.33}$$

where x_0, x_r, and u_r are given, $Q_x \in \mathcal{S}^{n_x}_{++}$, $R \in \mathcal{S}^{n_u}_{++}$, and $Q_N \in \mathcal{S}^{n_x}_{++}$ are the state, input, and terminal costs, and the sets $\mathcal{C}_x$ and $\mathcal{C}_u$ are convex. Suppose that the sets $\mathcal{C}_x$ and $\mathcal{C}_u$ correspond to component-wise lower and upper bounds, i.e., $\mathcal{C}_x = \{x \in \mathbb{R}^{n_x} | \mathbf{1}_{n_x}\bar{x}_{min} \le x \le \mathbf{1}_{n_x}\bar{x}_{max}\}$ and $\mathcal{C}_u = \{u \in \mathbb{R}^{n_u} | \mathbf{1}_{n_u}\bar{u}_{min} \le u \le \mathbf{1}_{n_u}\bar{u}_{max}\}$. Defining $\chi = [x_1^\top \dots x_{N_p}^\top]^\top$, $\upsilon = [u_0^\top \dots u_{N_p-1}^\top]^\top$, $\upsilon_r = [r^\top \dots r^\top]^\top$, (5.32) can be rewritten as $\chi = \Theta x_0 + \Phi\upsilon + \Phi_r\upsilon_r$. The latter relationship can be used to replace x_t for $t = 1, \dots, N_p$ in the optimization problem, yielding the following QP:

$$\begin{array}{ll} \underset{\upsilon}{\text{minimize}} & \frac{1}{2}\upsilon^\top Q\upsilon + q^\top\upsilon \\ \text{subject to} & A\upsilon \le b, \end{array} \tag{5.34}$$

where

$$\bar{Q} = \begin{bmatrix} I_{N_p-1} \otimes Q_x & 0 \\ 0 & Q_N \end{bmatrix}, \quad \bar{R} = I_{N_p} \otimes R,$$

$$A = \begin{bmatrix} \Phi \\ -\Phi \\ I \\ -I \end{bmatrix}, \quad b = \begin{bmatrix} \mathbf{1}_{n_xN_p}\bar{x}_{max} - \Theta x_0 - \Phi_r\upsilon_r \\ \mathbf{1}_{n_xN_p}\bar{x}_{min} + \Theta x_0 + \Phi_r\upsilon_r \\ \mathbf{1}_{n_uN_p}\bar{u}_{max} \\ \mathbf{1}_{n_uN_p}\bar{u}_{min} \end{bmatrix}, \tag{5.35}$$

and $Q = \bar{R} + \Phi^\top\bar{Q}\Phi$ and

$$q^\top = x_0^\top\Theta^\top\bar{Q}\Phi + \upsilon_r^\top\Phi_r^\top\bar{Q}\Phi - x_r^\top\left(\mathbf{1}_{N_p}^\top \otimes I_{n_x}\right)\bar{Q}\Phi - u_r^\top\left(\mathbf{1}_{N_p}^\top \otimes I_{n_u}\right)\bar{R}.$$

Below we illustrate the MPC problem for the quadruple-tank process [121]. The state of the process $x \in \mathbb{R}^4$ corresponds to the water levels of all tanks, measured in centimeters. The plant model was linearized at a given operating point and discretized with a sampling period of $2\,s$. The MPC prediction horizon was chosen as $N_p = 5$. A constant reference signal was used, while the initial condition x_0 was varied to obtain a set of MPC problems with different non-empty feasible sets and linear cost terms. In particular, we considered initial states of the form $x_0 = [x_1\, x_2\, x_3\, x_4]^\top$ where $x_i \in \{10,\ 11.25,\ 12.5,\ 13.75,\ 15\}$ for $i = 1, \dots, 4$. Out of the possible 625 initial values, 170 yields feasible QPs (each with $n = 10$ decision variables and $m = 40$ inequality constraints). We have made these QPs publicly available as a MATLAB formatted binary file [122]. To prevent possible ill-conditioned QP-problems, the constraint matrix A and vector b were scaled so that each row of A has unit-norm.

Figure 5.3 illustrates the convergence of the ADMM iterations for the 170 QPs as a function of the step-size ρ, scaling matrix L, and over-relaxation factor α. Since $A^\top$ has a non-empty null-space, the step-size $\rho^\star$ was chosen heuristically based on Theorem 5.4 as $\rho^\star = 1/\sqrt{\lambda_1(AQ^{-1}A^\top)\lambda_n(AQ^{-1}A^\top)}$, where $\lambda_1(AQ^{-1}A^\top)$ is the smallest nonzero eigenvalue of $AQ^{-1}A^\top$. As shown in Figure 5.3, the heuristic $\rho^\star$ results in a number of iterations close to the empirical minimum. Moreover, performance is improved by setting $L = L^\star$ and $\alpha = 2$.

The performance of the Fast-ADMM and ADMM algorithms is compared in Figure 5.4 for $L = I$ and $\alpha = 2$. The ADMM algorithm with the optimal over-relaxation factor $\alpha = 2$ uniformly outperforms the Fast-ADMM algorithm, even with suboptimal scaling matrix L.

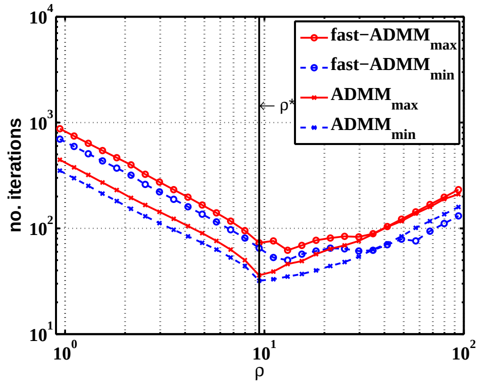
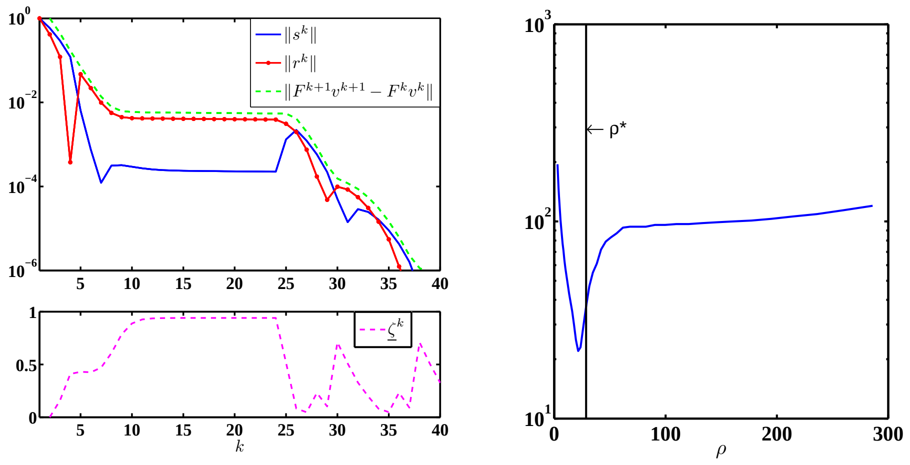


Figure 5.4: Number of iterations $k : \max\{\|r^{(k)}\|, \|s^{(k)}\|\} \leq 10^{-5}$ for ADMM with $L = I$ and $\alpha = 2$ and fast-ADMM algorithms applied to the MPC problem for different initial states x_0. The line in blue denotes the minimum number of iterations taken over all the initial states, while the red line represents the maximum number of iterations.



(a) Residuals and convergence factor lower bound for $\rho^\star = 28.6$.

(b) No. of iterations for $\rho \in [0.1\rho^\star,\ 10\rho^\star]$.

Figure 5.5: Slow convergence of ADMM algorithm for the example in (5.36) with $\alpha = 1$ and $L = I$. The residuals $r^{(k)}$, $s^{(k)}$, and $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ and the lower bound on the convergence factor $\underline{\zeta}^{(k)}$ are shown in the left, while the number of iterations for $\rho \in [0.1\rho^\star\ 10\rho^\star]$ are shown in the right.

Local convergence factor

To illustrate our results on the slow local convergence of ADMM, we consider a QP problem of the form (5.34) with

$$\begin{aligned} Q &= \begin{bmatrix} 40.513 & 0.069 \\ 0.069 & 40.389 \end{bmatrix}, \quad q = 0, \\ A &= \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0.1151 & 0.9934 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 6 \\ -0.3422 \end{bmatrix}. \end{aligned} \tag{5.36}$$

The ADMM algorithm was applied to the former optimization problem with $\alpha = 1$ and $L = I$. Given that the nullity of A is not 0, the step-size was chosen heuristically based on Theorem 5.4 as $\rho^\star = 1/\sqrt{\lambda_1(AQ^{-1}A^\top)\lambda_n(AQ^{-1}A^\top)} = 28.6$ with $\lambda_1(AQ^{-1}A^\top)$ taken to be the smallest nonzero eigenvalue of $AQ^{-1}A^\top$. The resulting residuals are shown in Figure 5.5, together with the lower bound on the convergence factor $\underline{\zeta}$ evaluated at each time-step. As expected from the results in Theorem 5.3, the residual $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ is monotonically decreasing. However, as illustrated by $\underline{\zeta}^{(k)}$, the lower bound on the convergence factor from Theorem 5.5, the auxiliary residual $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ and the primal-dual residuals show a convergence factor close to 1 over several time-steps. The heuristic step-size rule performs reasonably well as illustrated in the right subplot of Figure 5.5.

5.5 Summary

In this chapter, we studied optimal parameter selection for the ADMM algorithm for two classes of quadratic problems: ℓ_2-regularized quadratic minimization and quadratic programming under linear inequality constraints. For both problem classes, we established global convergence of the algorithm at linear rate and provided explicit expressions for the parameters that ensure the smallest possible convergence factors. We also considered iterations accelerated by over-relaxation, characterized the values of the relaxation parameter for which the over-relaxed iterates are guaranteed to improve the convergence times compared to the non-relaxed iterations, and derived jointly optimal step-size and relaxation parameters. We validated the analytical results on numerical examples and demonstrated superior performance of the tuned ADMM algorithms compared to existing methods from the literature.

Appendix

5.A Proofs

5.A.1 Proof of Theorem 5.1

From Proposition 2.1, the variables $x^{(k)}$ and $z^{(k)}$ in iterations (5.2) converge to the optimal values $x^\star$ and $z^\star$ of (5.1) if and only if the spectral radius of the matrix E in (5.3) is less than one. To express the eigenvalues of E in terms of the eigenvalues of Q, let $\lambda_i(Q), i = 1, \dots, n$ be the eigenvalues of Q sorted in ascending order. Then, the eigenvalues $\zeta(\rho, \lambda_i(Q))$ of E satisfy

$$\zeta(\rho, \lambda_i(Q)) = \frac{\rho^2 + \lambda_i(Q)\delta}{\rho^2 + \lambda_i(Q)\delta + (\lambda_i(Q) + \delta)\rho}. \tag{5.37}$$

Since $\lambda_i(Q), \rho, \delta \in \mathbb{R}_{++}$, we have $0 \leq \zeta(\rho, \lambda_i(Q)) < 1$ for all i, which ensures convergence.

To find the optimal step-size parameter and the associated convergence factor $(\rho^\star, \zeta^\star)$, note that, for a fixed ρ, the convergence factor $\zeta(\rho) = \max_{e^{(k)}} \|e^{(k+1)}\|/\|e^{(k)}\|$ corresponds to the spectral radius of E, i.e., $\zeta(\rho) = \max_i \{\zeta(\rho, \lambda_i(Q))\}$. It follows that the optimal pair $(\rho^\star, \zeta^\star)$ is given by

$$\rho^\star = \underset{\rho}{\operatorname{argmin}} \max_i \{\zeta(\rho, \lambda_i(Q))\}, \quad \zeta^\star = \max_i \{\zeta(\rho^\star, \lambda_i(Q))\}. \tag{5.38}$$

From (5.37), we can see that $\zeta(\rho, \lambda_i(Q))$ is monotone decreasing in $\lambda_i(Q)$ when $\rho > \delta$ and monotone increasing when $\rho < \delta$. Hence, we consider these two cases separately.

When $\rho > \delta$, the largest eigenvalue of E is given by $\zeta(\rho, \lambda_1(Q))$ and $\rho^\star = \operatorname{argmin}_\rho \zeta(\rho, \lambda_1(Q))$. By the first-order optimality conditions and the explicit expressions in (5.37) we have

$$\rho^\star = \sqrt{\delta\lambda_1(Q)}, \quad \zeta^\star = \zeta(\rho^\star, \lambda_1(Q)) = (1 + \frac{\delta + \lambda_1(Q)}{2\sqrt{\delta\lambda_1(Q)}})^{-1}.$$

However, this value of ρ is larger than δ only if $\delta < \lambda_1(Q)$. When $\delta \geq \lambda_1(Q)$, the assumption that $\rho > \delta$ implies that $0 \leq (\rho - \delta)^2 \leq (\rho - \delta)(\rho - \lambda_1(Q))$, so

$$\zeta(\rho, \lambda_1(Q)) = \frac{\rho^2 + \lambda_i(Q)\delta}{\rho^2 + \lambda_i(Q)\delta + (\lambda_i(Q) + \delta)\rho} \geq$$

$$\frac{\rho^2 + \lambda_1(Q)\delta}{\rho^2 + \lambda_1(Q)\delta + (\lambda_1(Q) + \delta)\rho + (\rho - \delta)(\rho - \lambda_1(Q))} = \frac{1}{2}.$$

Since $\rho = \delta$ attains $\zeta(\delta, \lambda_1(Q)) = 1/2$ it is optimal.

A similar argument applies to $\rho < \delta$. In this case, $\max_i \zeta(\rho, \lambda_i(Q)) = \zeta(\rho, \lambda_n(Q))$ and when $\delta > \lambda_n(Q)$, $\rho^\star = \sqrt{\delta\lambda_n(Q)}$ is the optimal step-size and the associated convergence factor is

$$\zeta^\star = \left(1 + \frac{\delta + \lambda_n(Q)}{2\sqrt{\delta\lambda_n(Q)}}\right)^{-1}.$$

For $\delta \leq \lambda_n(Q)$, the requirement $\rho < \delta$ implies

$$0 \leq (\delta - \rho)^2 \leq (\lambda_n(Q) - \rho)(\delta - \rho), \quad \zeta(\rho, \lambda_n(Q)) \geq \frac{1}{2},$$

which leads to $\rho = \delta$ being optimal.

5.A.2 Proof of Corollary 5.1

The proof is a direct consequence of evaluating (5.37) at $\rho = \delta$ for $i = 1, \dots, n$.

5.A.3 Proof of Theorem 5.2

The z-update in (5.6) implies that $y^{(k)} = (\delta + \rho)z^{(k+1)} - \rho(\alpha x^{(k+1)} + (1 - \alpha)z^{(k)})$, and that the y-update in (5.6) can be written as $y^{(k+1)} = \delta z^{(k+1)}$. Similarly to the analysis of Section 5.2.1, inserting the x-update into the z-update, we find

$$\begin{aligned} z^{(k+1)} = & \underbrace{\frac{1}{\delta + \rho}\left(\delta I + \rho\left(\alpha(\rho - \delta)(Q + \rho I)^{-1} + (1 - \alpha)I\right)\right)}_{E_R} z^{(k)} \\ & - \frac{1}{\delta + \rho}\rho\alpha(Q + \rho I)^{-1}q. \end{aligned}$$

Consider the fixed-point candidate $z^\star$ satisfying $z^\star = E_R z^\star - \dfrac{1}{\delta + \rho}\rho\alpha(Q + \rho I)^{-1}q$ and $z^{(k+1)} - z^\star = E_R(z^{(k)} - z^\star)$. The $z^{(k)}$-update in (5.6) converges (and so does the ADMM algorithm) if and only if the spectral radius of the error matrix in the above linear iterations is less than one. The eigenvalues of E_R can be written as

$$\zeta_R(\alpha, \rho, \lambda_i(Q)) = 1 - \frac{\alpha\rho(\lambda_i(Q) + \delta)}{(\rho + \lambda_i(Q))(\rho + \delta)}. \tag{5.39}$$

Since ρ, δ, and $\lambda_i(Q) \in \mathbb{R}_{++}$, we see that

$$0 < \alpha < 2\min_i \frac{(\rho + \delta)(\rho + \lambda_i(Q))}{\rho(\lambda_i(Q) + \delta)},$$

implies that $|\zeta_R(\alpha, \rho, \lambda_i(Q))| < 1$ for all i, which completes the first part of the proof.

For a fixed ρ and δ, we now characterize the values of α that ensure that the over-relaxed iterations (5.6) have a smaller convergence factor and thus a smaller ε-solution time than the classical ADMM iterates (5.2), i.e., $\zeta_R - \zeta < 0$. From (5.37) and (5.39) we have $\operatorname{argmax}_i \zeta_R(\alpha, \rho, \lambda_i(Q)) = \operatorname{argmax}_i \zeta(\rho, \lambda_i(Q))$, since ζ_R and ζ are equivalent up to an affine transformation and they have the same sign of the derivative with respect to $\lambda_i(Q)$. For any given $\lambda_i(Q)$ we have

$$\zeta_R - \zeta = \frac{\rho(1-\alpha)(\lambda_i(Q)+\delta)}{\rho^2 + (\lambda_i(Q)+\delta)\rho + \lambda_i(Q)\delta}$$

and we conclude that $\zeta_R - \zeta < 0$ when

$$\alpha \in \left(1,\ \frac{2(\rho+\delta)(\rho+\lambda_i(Q))}{\rho(\lambda_i(Q)+\delta)}\right).$$

Recalling the first part of the proof we conclude that, for given $\rho, \delta \in \mathbb{R}_{++}$, the over-relaxed iterations converge with a smaller convergence factor than classical ADMM for

$$1 < \alpha < 2\min_i \frac{(\rho+\delta)(\rho+\lambda_i(Q))}{\rho(\lambda_i(Q)+\delta)}.$$

To find $(\rho^\star, \alpha^\star, \zeta_R^\star)$, we define

$$(\rho^\star, \alpha^\star) = \underset{\rho,\alpha}{\operatorname{argmin}}\ \max_i |\zeta_R(\rho, \alpha, \lambda_i(Q))|, \quad \zeta_R^\star = \max_i |\zeta_R(\rho^\star, \alpha^\star, \lambda_i(Q))|. \tag{5.40}$$

One readily verifies that $\zeta_R(\delta, 2, \lambda_i(Q)) = 0$ for $i = 1, \dots n$. Since zero is the global minimum of $|\zeta_R|$ we conclude that the pair $(\rho^\star, \alpha^\star) = (\delta, 2)$ is optimal. Moreover, for $(\rho^\star, \alpha^\star) = (\delta, 2)$ the matrix E_R is a matrix of zeros and thus the algorithm (5.6) converges in one iteration.

5.A.4 Proof of Proposition 5.1

For the sake of brevity we derive the expressions only for $w_-^{(k+1)} \triangleq F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$, as similar computations also apply to $w_+^{(k+1)} \triangleq v^{(k+1)} - v^{(k)}$. First, since $v^{(k)} = z^{(k)} + u^{(k)}$, it holds that $F^{(k)}v^{(k)} = (2D^{(k)} - I)v^{(k)} = 2D^{(k)}v^{(k)} - u^{(k)} - z^{(k)}$. From the equality $D^{(k)}v^{(k)} = u^{(k)}$ we then have $F^{(k)}v^{(k)} = u^{(k)} - z^{(k)}$. The residual $w_-^{(k+1)}$ can be rewritten as $w_-^{(k+1)} = u^{(k+1)} - u^{(k)} - z^{(k+1)} + z^{(k)}$. From (2.17) and (5.10) we observe that $u^{(k+1)} - u^{(k)} = r^{(k+1)}$, so $w_-^{(k+1)} = r^{(k+1)} - (z^{(k+1)} - z^{(k)})$. Decomposing $z^{(k+1)} - z^{(k)}$ as $\Pi_{\mathrm{Im}(A)}(z^{(k+1)} - z^{(k)}) + \Pi_{\mathcal{N}(A^\top)}(z^{(k+1)} - z^{(k)})$ we then conclude that

$$w_-^{(k+1)} = r^{(k+1)} - \Pi_{\mathrm{Im}(A)}(z^{(k+1)} - z^{(k)}) - \Pi_{\mathcal{N}(A^\top)}(z^{(k+1)} - z^{(k)}).$$

We now examine each case (i)-(iii) separately:

(i) When A has full column rank, $\Pi_{\text{Im}(A)} = A(A^\top A)^{-1}A^\top$ and $\Pi_{\mathcal{N}(A^\top)} = I - \Pi_{\text{Im}(A)}$. In the light of the dual residual (2.18) we obtain $\Pi_{\text{Im}(A)}(z^{(k+1)} - z^{(k)}) = 1/\rho A(A^\top A)^{-1}s^{(k+1)}$.

(ii) Note that the nullity of $A^\top$ is 0 if A is full row-rank. Thus, $\Pi_{\mathcal{N}(A^\top)} = 0$ and $\Pi_{\text{Im}(A)} = I$. Moreover, since $AA^\top$ is invertible, $z^{(k+1)} - z^{(k)} = (AA^\top)^{-1}AA^\top(z^{(k+1)} - z^{(k)}) = 1/\rho(AA^\top)^{-1}As^{(k+1)}$.

(iii) When A is invertible, the result easily follows.

We now relate the norm of $r^{(k+1)}$ and $s^{(k+1)}$ to the one of $w_-^{(k+1)}$. From (5.15) and (5.16), we have

$$\begin{aligned}\|r^{(k+1)}\| &= \frac{1}{2}\|w_-^{(k+1)} + w_+^{(k+1)}\| \\ &\leq \frac{1}{2}(\|w_-^{(k+1)}\| + \|w_+^{(k+1)}\|) \leq \|w_-^{(k+1)}\|,\end{aligned}$$

where the first inequality is the triangle inequality and the last inequality holds as $v^{(k)}$'s are positive vectors, $\|w_+^{(k+1)}\| = \|v^{(k+1)} - v^{(k)}\| \leq \|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\| = \|w_-^{(k+1)}\|$.

For the dual residual, it can be verified that in case (i) and (ii),

$$A^\top(w_+^{(k+1)} - w_-^{(k+1)}) = \frac{2}{\rho}s^{(k+1)},$$

therefore,

$$\begin{aligned}\|s^{(k+1)}\| &= \frac{\rho}{2}\|A^\top(w_-^{(k+1)} - w_+^{(k+1)})\| \\ &\leq \frac{\rho}{2}\|A\|\left(\|w_-^{(k+1)} - w_+^{(k+1)}\|\right) \\ &\leq \frac{\rho}{2}\|A\|\left(\|w_-^{(k+1)}\| + \|w_+^{(k+1)}\|\right) \leq \rho\|A\|\|w_-^{(k+1)}\|.\end{aligned}$$

In case (iii), one finds $A(w_+^{(k+1)} - w_-^{(k+1)}) = (2/\rho)s^{(k+1)}$ and again the same bound can be achieved (by replacing $A^\top$ with A in above equality), thus concluding the proof.

5.A.5 Proof of Theorem 5.3

Note that since $v^{(k)}$ is positive and $F^{(k)}$ is diagonal with elements in ±1, $F^{(k+1)}v^{(k+1)} = F^{(k)}v^{(k)}$ implies $v^{(k+1)} = v^{(k)}$. Hence, it suffices to establish the convergence of $F^{(k)}v^{(k)}$. From (5.14) we have

$$\begin{aligned}\left\|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\right\| &\leq \frac{1}{2}\|2M - I\|\left\|v^{(k)} - v^{(k-1)}\right\| \\ &\quad + \frac{1}{2}\left\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\right\|.\end{aligned}$$

Furthermore, as $v^{(k)}$s are positive vectors, $\left\|v^{(k)} - v^{(k-1)}\right\| \le \left\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\right\|$, which implies

$$\left\|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\right\| \le \underbrace{\left(\frac{1}{2}\left\|2M - I\right\| + \frac{1}{2}\right)}_{\zeta}\left\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\right\|. \tag{5.41}$$

We conclude that if $\|2M - I\| < 1$, then $\zeta < 1$ and the iterations (5.14) converge to zero at a linear rate.

To determine for what values of ρ the iterations (5.14) converge, we characterize the eigenvalues of M. By the matrix inversion lemma

$$M = \rho AQ^{-1}A^\top - \rho AQ^{-1}A^\top(I + \rho AQ^{-1}A^\top)^{-1}\rho AQ^{-1}A^\top.$$

From [99, Cor. 2.4.4], $(I + \rho AQ^{-1}A^\top)^{-1}$ is a polynomial function of $\rho AQ^{-1}A^\top$ which implies that $M = f(t) \triangleq t - t(1+t)^{-1}t$ is a polynomial function of $t = \rho AQ^{-1}A^\top$. Applying [99, Thm. 1.1.6], the eigenvalues of M are given by $f(\lambda_i(\rho AQ^{-1}A^\top))$ and thus

$$\lambda_i(M) = \frac{\lambda_i(\rho AQ^{-1}A^\top)}{1 + \lambda_i(\rho AQ^{-1}A^\top)}. \tag{5.42}$$

If $\rho \in \mathbb{R}_{++}$, then $\lambda_i(\rho AQ^{-1}A^\top) \in \mathbb{R}_+$ and $\lambda_i(M) \in [0, 1)$. Hence $\|2M - I\| \le 1$ is guaranteed for all $\rho \in \mathbb{R}_{++}$ and equality only occurs if M has eigenvalues at 0. If A is invertible or has full row-rank, then M is invertible and all its eigenvalues are strictly positive, so $\|2M - I\| < 1$ and (5.14) is guaranteed to converge linearly.

The case when A is tall, i.e., $A^\top$ is rank deficient, is more challenging since M has zero eigenvalues and $\|2M - I\| = 1$. To prove convergence in this case, we analyze the 0-eigenspace of M and show that it can be disregarded. From the x-iterates given in (5.11) we have $x^{(k+1)} - x^{(k)} = -(Q/\rho + A^\top A)^{-1}A^\top(v^{(k)} - v^{(k-1)})$. Multiplying the former equality by A from the left on both sides yields $A(x^{(k+1)} - x^{(k)}) = -M(v^{(k)} - v^{(k-1)})$. Consider a nonzero vector $v^{(k)} - v^{(k-1)}$ in $\mathcal{N}(M)$. Then we have either $x^{(k+1)} = x^{(k)}$ or $x^{(k+1)} - x^{(k)} \in \mathcal{N}(A)$. Having assumed that A is full column-rank denies the second hypothesis. In other words, the 0-eigenspace of M corresponds to the stationary points of the algorithm (5.11). We therefore disregard this eigenspace and the convergence result holds.

Finally, the R-linear convergence of the primal and dual residuals follows from the linear convergence rate of $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ and Proposition 5.1.

5.A.6 Proof of Theorem 5.4

From the proof of Theorem 5.3 recall that

$$\left\|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\right\| \le \frac{1}{2}\Big(\|2M - I\| + 1\Big)\left\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\right\|.$$

Define

$$\zeta \triangleq \frac{1}{2}\|2M - I\| + \frac{1}{2} = \max_i \frac{1}{2}|2\lambda_i(M) - 1| + \frac{1}{2}$$
$$= \max_i \left| \frac{\rho\lambda_i(AQ^{-1}A^\top)}{1 + \rho\lambda_i(AQ^{-1}A^\top)} - \frac{1}{2} \right| + \frac{1}{2},$$

where the last equality follows from the definition of $\lambda_i(M)$ in (5.42). Since $\rho \in \mathbb{R}_{++}$ and for the case where A is either invertible or has full row-rank, $\lambda_i(AQ^{-1}A^\top) \in \mathbb{R}_{++}$ for all i, we conclude that $\zeta < 1$.

It remains to find $\rho^\star$ that minimizes the convergence factor, *i.e.*

$$\rho^\star = \underset{\rho}{\operatorname{argmin}} \max_i \left\{ \left| \frac{\rho\lambda_i(AQ^{-1}A^\top)}{1 + \rho\lambda_i(AQ^{-1}A^\top)} - \frac{1}{2} \right| + \frac{1}{2} \right\}. \tag{5.43}$$

Since $\dfrac{\rho\lambda_i(AQ^{-1}A^\top)}{1 + \rho\lambda_i(AQ^{-1}A^\top)}$ is a monotonically increasing function in $\lambda_i(AQ^{-1}A^\top)$, the maximum values of ζ happen for the two extreme eigenvalues $\lambda_1(AQ^{-1}A^\top)$ and $\lambda_n(AQ^{-1}A^\top)$:

$$\max_i \left\{ \zeta(\lambda_i(AQ^{-1}A^\top), \rho) \right\} = \begin{cases} \dfrac{1}{1 + \rho\lambda_1(AQ^{-1}A^\top)} & \text{if} \quad \rho \le \rho^\star, \\ \dfrac{\rho\lambda_n(AQ^{-1}A^\top)}{1 + \rho\lambda_n(AQ^{-1}A^\top)} & \text{if} \quad \rho > \rho^\star. \end{cases} \tag{5.44}$$

Since the left brace of $\max_i \left\{ \zeta(\lambda_i(AQ^{-1}A^\top), \rho) \right\}$, i.e. $(1 + \rho\lambda_1(AQ^{-1}A^\top))^{-1}$ is monotone decreasing in ρ and the right brace is monotone increasing, the minimum with respect to ρ happens at the intersection point (5.19).

5.A.7 Proof of Theorem 5.5

First we derive the lower bound on the convergence factor and show it is strictly smaller than 1. From (5.14) we have

$$\left\| F^{(k+1)} v^{(k+1)} - F^{(k)} v^{(k)} \right\| = \left\| D^{(k)} v^{(k)} - D^{(k-1)} v^{(k-1)} - M(v^{(k)} - v^{(k-1)}) \right\|.$$

By applying the reverse triangle inequality and dividing by $\|F^{(k)} v^{(k)} - F^{(k-1)} v^{(k-1)}\|$, we find

$$\frac{\|F^{(k+1)} v^{(k+1)} - F^{(k)} v^{(k)}\|}{\|F^{(k)} v^{(k)} - F^{(k-1)} v^{(k-1)}\|} \ge |\delta_k - \epsilon_k|.$$

Recalling from (2.1) that the convergence factor ζ is the maximum over k of the left hand-side yields the lower bound (5.20). Moreover, the inequality $1 > \zeta \ge \underline{\zeta}$ follows directly from Theorem 5.3.

The second part of the proof addresses the cases (i)-(iii) for $\rho > 0$. Consider case (i) and let $\mathcal{N}(A^\top) = \{0\}$. It follows from Theorem 5.4 that the convergence factor is given by $\tilde{\zeta}(\rho)$, thus proving the sufficiency of $\mathcal{N}(A^\top) = \{0\}$ in (i). The necessity follows directly from statement (iii), which is proved later.

Now consider the statement (ii) and suppose $\mathcal{N}(A^\top)$ is not zero-dimensional. Recall that $\lambda_1(AQ^{-1}A^\top)$ is the smallest nonzero eigenvalue of $AQ^{-1}A^\top$ and suppose that $\epsilon_k \geq 1-\xi$. Next we show that $\epsilon_k \geq 1 - \xi$ implies

$$\frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|}{\|v^{(k)} - v^{(k-1)}\|} \leq \sqrt{2\xi}.$$

Since $M\Pi_{\mathcal{N}(A^\top)} = 0$, $\|M\| < 1$, and $\|v^{(k)} - v^{(k-1)}\| \leq \|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|$ we have

$$\begin{aligned}\epsilon_k^2 &= \frac{\|M(I - \Pi_{\mathcal{N}(A^\top)})(v^{(k)} - v^{(k-1)})\|^2}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|^2} \leq \frac{\|\Pi_{\mathrm{Im}(A)}(v^{(k)} - v^{(k-1)})\|^2}{\|v^{(k)} - v^{(k-1)}\|^2} \\ &= 1 - \frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|^2}{\|v^{(k)} - v^{(k-1)}\|^2}.\end{aligned}$$

Using the above inequality and $\epsilon_k^2 \geq (1-\xi)^2$ we obtain

$$\frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|}{\|v^{(k)} - v^{(k-1)}\|} \leq \sqrt{2\xi - \xi^2} \leq \sqrt{2\xi}.$$

The latter inequality allows us to derive an upper-bound on $\underline{\zeta}$ as follows. Recalling (5.14), we have

$$\begin{aligned}\underline{\zeta} \leq& \frac{\|F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}\|}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|} \leq \frac{1}{2} + \frac{1}{2}\frac{\|(I - 2M)(v^{(k)} - v^{(k-1)})\|}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|} \\ &= \frac{1}{2} + \frac{1}{2}\sqrt{\frac{\|(I-2M)\Pi_{\mathrm{Im}(A)}(v^{(k)} - v^{(k-1)})\|^2}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|^2} + \frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|^2}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|^2}}.\end{aligned} \tag{5.45}$$

Using the inequalities $\|v^{(k)} - v^{(k-1)}\| \leq \|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|$ and $\sqrt{a^2+b^2} \leq a+b$ for $a, b \in \mathbb{R}_+$, the inequality (5.45) becomes

$$\underline{\zeta} \leq \frac{1}{2} + \frac{1}{2}\|(I-2M)\Pi_{\mathrm{Im}(A)}\| + \sqrt{\frac{\xi}{2}} \leq \tilde{\zeta}(\rho) + \sqrt{\frac{\xi}{2}},$$

which concludes the proof of (ii).

As for the third case (iii), note that $\epsilon_k \leq \xi$ holds if

$$\frac{\|\Pi_{\mathcal{N}(A^\top)}(v^{(k)} - v^{(k-1)})\|}{\|v^{(k)} - v^{(k-1)}\|} \geq \sqrt{1 - \frac{\xi^2}{\|M\|^2}}, \tag{5.46}$$

as the latter inequality implies that

$$\epsilon_k = \frac{\|M\Pi_{\mathrm{Im}(A)}(v^{(k)} - v^{(k-1)})\|}{\|F^{(k)}v^{(k)} - F^{(k-1)}v^{(k-1)}\|} \leq \|M\|\frac{\|\Pi_{\mathrm{Im}(A)}(v^{(k)} - v^{(k-1)})\|}{\|v^{(k)} - v^{(k-1)}\|} \leq \xi.$$

Supposing that there exists a non-empty set $\mathcal{K}$ such that $\delta_k \geq 1 - \xi$ and (5.46) holds for all $k \in \mathcal{K}$, we have $\underline{\zeta} \geq \max_{k\in\mathcal{K}}\ \delta_k - \epsilon_k \geq 1 - 2\xi$ regardless the choice of ρ.

5.A.8 Proof of Lemma 5.1

Let $(x^\star,\, z^\star,\, u^\star)$ denote a fixed-point of (5.21) and let y be the Lagrange multiplier associated with the equality constraint in (5.9). For the optimization problem (5.9), the KKT optimality conditions [50] are

$$\begin{aligned} 0 &= Qx + q + A^\top y, \qquad & z &\geq 0, \\ 0 &= Ax + z - b, & 0 &= \operatorname{diag}(y) z. \end{aligned}$$

Next we show that the KKT conditions hold for the fixed-point $(x^\star,\, z^\star,\, u^\star)$ with $y^\star = 1/\rho u^\star$. From the $u-$iterations we have $0 = \alpha(Ax^\star - c) - (1-\alpha)z^\star + z^\star = \alpha(Ax^\star + z^\star - c)$. It follows that $z^\star$ is given by $z^\star = \max\{0, -\alpha(Ax^\star + z^\star - c) + z^\star - u^\star\} = \max\{0, z^\star - u^\star\} \geq 0$. The $x-$iteration then yields $0 = Qx^\star + q + \rho A^\top(Ax^\star + z^\star - c + u^\star) = Qx^\star + q + A^\top y^\star$. Finally, from $z^\star \geq 0$ and the $z-$update, we have that $z_i^\star > 0 \Rightarrow u_i^\star = 0$ and $z_i^\star = 0 \Rightarrow u_i^\star \geq 0$. Thus, $\rho \operatorname{diag}(y^\star) z^\star = 0$.

5.A.9 Proof of Theorem 5.6

Taking the Euclidean norm of (5.24) and applying the Cauchy-Schwarz inequality yields

$$\begin{aligned} \left\| F^{(k+1)} v^{(k+1)} - F^{(k)} v^{(k)} \right\| \leq \frac{|\alpha|}{2} \left\| 2M - I \right\| \left\| v^{(k)} - v^{(k-1)} \right\| \\ + |1 - \frac{\alpha}{2}| \left\| F^{(k)} v^{(k)} - F^{(k-1)} v^{(k-1)} \right\|. \end{aligned}$$

Note that since $v^{(k)}$s are positive vectors we have

$$\left\| v^{(k)} - v^{(k-1)} \right\| \leq \left\| F^{(k)} v^{(k)} - F^{(k-1)} v^{(k-1)} \right\|,$$

and thus

$$\frac{\left\| F^{(k+1)} v^{(k+1)} - F^{(k)} v^{(k)} \right\|}{\left\| F^{(k)} v^{(k)} - F^{(k-1)} v^{(k-1)} \right\|} \leq \underbrace{\left(\frac{|\alpha|}{2} \left\| 2M - I \right\| + \left| 1 - \frac{\alpha}{2} \right| \right)}_{\zeta_R}. \tag{5.47}$$

Note that $\rho \in \mathbb{R}_{++}$ and recall from the proof of Theorem 5.3 that the 0-eigenspace of M can be disregarded. Let $\tau \triangleq \max_{\lambda_i} \frac{1}{2} |2\lambda_i(M) - 1|$ with $i = 1, \ldots, m - \dim(\mathcal{N}(M))$, where the upper bound on i is to discard the 0-eigenspace of M. Note that $\tau < \frac{1}{2}$ and we have

$$\zeta_R = \alpha\tau + |1 - \frac{\alpha}{2}| < \frac{\alpha}{2} + |1 - \frac{\alpha}{2}|$$

Hence, we conclude that for $\rho \in \mathbb{R}_{++}$ and $\alpha \in (0, 2]$, it holds that $\zeta_R < 1$, which implies that (5.24) converges linearly to a fixed-point. By Lemma 5.1 this fixed-point is also a global optimum of (5.8). Now, denote $w_-^{(k+1)} \triangleq F^{(k+1)} v^{(k+1)} - F^{(k)} v^{(k)}$ and $w_+^{(k+1)} \triangleq$

$v^{(k+1)} - v^{(k)}$. Following the same steps as Proposition 5.1, it is easily verified that $w_-^{(k+1)} = u^{(k+1)} - u^{(k)} + z^{(k)} - z^{(k+1)}$ and $w_+^{(k+1)} = u^{(k+1)} - u^{(k)} + z^{(k+1)} - z^{(k)}$ from which combined with (5.21) one obtains

$$s^{(k+1)} = \rho \frac{A^\top}{2}(w_+^{(k+1)} - w_-^{(k+1)}), \quad r^{(k+1)} = \frac{1}{2} w_+^{(k+1)} + \frac{2-\alpha}{2\alpha} w_-^{(k+1)}.$$

We only upper-bound $\|r^{(k+1)}\|$, since an upper bound for $\|s^{(k+1)}\|$ was already established in (5.18). Taking the Euclidean norm of the second equality above and using the triangle inequality

$$\|r^{(k+1)}\| \le \frac{1}{2}\|w_+^{(k+1)}\| + \frac{2-\alpha}{2\alpha}\|w_-^{(k+1)}\| \le \frac{1}{\alpha}\|w_-^{(k+1)}\|. \tag{5.48}$$

The R-linear convergence of the primal and dual residuals now follows from the linear convergence rate of $F^{(k+1)}v^{(k+1)} - F^{(k)}v^{(k)}$ and the bounds in (5.18) and (5.48).

5.A.10 Proof of Theorem 5.7

Define

$$\begin{aligned} \zeta_R\Big(\rho, \alpha, \lambda_i(AQ^{-1}A^\top)\Big) &= \alpha \left| \frac{\rho\lambda_i(AQ^{-1}A^\top)}{1+\rho\lambda_i(AQ^{-1}A^\top)} - \frac{1}{2} \right| + 1 - \frac{\alpha}{2}, \\ \zeta_R^\star &= \max_i \min_{\rho,\alpha} \Big\{ \zeta_R(\rho, \alpha, \lambda_i(AQ^{-1}A^\top)) \Big\}. \end{aligned} \tag{5.49}$$

Since

$$\left| \frac{\rho\lambda_i(AQ^{-1}A^\top)}{1+\rho\lambda_i(AQ^{-1}A^\top)} - \frac{1}{2} \right| < \frac{1}{2},$$

it follows that $\zeta_R(\rho, \alpha, \lambda_i(AQ^{-1}A^\top))$ is monotone decreasing in α. Thus, $\zeta_R(\rho, \alpha, \lambda_i(AQ^{-1}A^\top))$ is minimized by $\alpha^\star = 2$. To determine

$$\rho^\star = \underset{\rho}{\operatorname{argmin}} \ \max_i \left\{ \zeta_R(\rho, 2, \lambda_i(AQ^{-1}A^\top)) \right\}, \tag{5.50}$$

we note that (5.43) and (5.50) are equivalent up to an affine transformation, hence we have the same minimizer $\rho^\star$. It follows from the proof of Theorem 5.4 that

$$\rho^\star = \frac{1}{\sqrt{\lambda_1(AQ^{-1}A^\top)\,\lambda_n(AQ^{-1}A^\top)}}.$$

Using $\rho^\star$ in (5.49) results in the convergence factor (5.26).

For given A, Q, and ρ, we can now find the range of values of α for which (5.21) have a smaller convergence factor than (5.10), i.e., for which $\zeta_R - \zeta < 0$. By (5.41) and (5.47) it holds that

$$\zeta_R - \zeta = \frac{\alpha}{2}\|2M - I\| + 1 - \frac{\alpha}{2} - \frac{1}{2}\|2M - I\| - \frac{1}{2}$$

$$= (1-\alpha)\left(\frac{1}{2} - \frac{1}{2}\|2M - I\|\right).$$

This means that $\zeta_R - \zeta < 0$ when $\alpha > 1$. Therefore, the iterates produced by the relaxed algorithm (5.21) have smaller convergence factor than the iterates produced by (5.10) for all values of the relaxation parameter $\alpha \in (1, 2]$. This concludes the proof.

5.A.11 Proof of Theorem 5.8

Note that the non-zero eigenvalues of $LAQ^{-1}A^\top L$ are the same as the ones of $R_q^\top A^\top WAR_q$ where $W = L^2$ and $R_q^\top R_q = Q^{-1}$ is its Choleski factorization [99]. Defining $\lambda_n(R_q^\top A^\top WAR_q)$ and $\lambda_1(R_q^\top A^\top WAR_q)$ as the largest and smallest nonzero eigenvalues of $LAQ^{-1}A^\top L$, the optimization problem we aim at solving can be formulated as

$$\begin{array}{ll}\underset{\bar{\lambda}\in\mathbb{R},\, \underline{\lambda}\in\mathbb{R},\, l\in\mathbb{R}^m}{\text{minimize}} & \bar{\lambda}/\underline{\lambda} \\ \text{subject to} & \bar{\lambda} > \lambda_n(R_q^\top A^\top WAR_q), \\ & \lambda_1(R_q^\top A^\top WAR_q) > \underline{\lambda}, \\ & W = \operatorname{diag}(w),\ w > 0. \end{array} \tag{5.51}$$

In the proof we show that the optimization problem (5.51) is equivalent to (5.27).

Define $T(\bar{\lambda}) \triangleq \bar{\lambda}I - R_q^\top A^\top WAR_q$. First observe that $\bar{\lambda} \geq \lambda_n(R_q^\top A^\top WAR_q)$ holds if and only if $T(\bar{\lambda}) \in \mathcal{S}_+^n$, which proves the first inequality in the constraint set (5.27).

To obtain a lower bound on $\lambda_1(R_q^\top A^\top WAR_q)$ one must disregard the zero eigenvalues of $R_q^\top A^\top WAR_q$ (if they exist). This can be performed by restricting ourselves to the subspace orthogonal to $\mathcal{N}(R_q^\top A^\top WAR_q) = \mathcal{N}(AR_q)$. In fact, letting s to be the dimension of the nullity of AR_q or simply A and denoting $P^{n\times n-s}$ as a basis of $\operatorname{Im}(R_q^\top A^\top)$, we have that $\underline{\lambda} \leq \lambda_1$ if and only if $x^\top P^\top T(\underline{\lambda})Px \leq 0$ for all $x \in \mathbb{R}^{n-s}$. Note that for the case when the nullity of A is 0 ($s = 0$), all the eigenvalues of $R_q^\top A^\top WAR_q$ are strictly positive and, hence, one can set $P = I$. We conclude that $\underline{\lambda} \leq \lambda_1(R_q^\top A^\top WAR_q)$ if and only if $P^\top \left(R_q^\top A^\top WAR_q - \underline{\lambda}I\right) P \in \mathcal{S}_+^{n-s}$.

Note that $\lambda_1(R_q^\top A^\top WAR_q) > 0$ can be chosen arbitrarily by scaling W, which does not affect the ratio $\lambda_n(R_q^\top A^\top WAR_q)/\lambda_1(R_q^\top A^\top WAR_q)$. Without loss of generality, one can suppose $\underline{\lambda}^\star = 1$ and thus the lower bound on $\lambda_1(R_q^\top A^\top WAR_q) \geq \underline{\lambda}^\star = 1$ corresponds to the last inequality in the constraint set of (5.27). Observe that the optimization problem now reduces to minimizing $\bar{\lambda}$. The proof concludes by rewriting (5.51) as (5.27), which is a convex problem.

5.A.12 Proof of Proposition 5.2

Assuming $F^{(k+1)} = F^{(k)} = -I$, (5.23) reduces to $v^{(k+1)} - v^{(k)} = ((1-\alpha)I + \alpha M)\,(v^{(k)} - v^{(k-1)})$. By taking the Euclidean norm of both sides and applying the Cauchy inequality, we find

$$\|v^{(k+1)} - v^{(k)}\| \leq \|(1-\alpha)I + \alpha M\|\|v^{(k)} - v^{(k-1)}\|.$$

Since the eigenvalues M are $\rho\lambda_i(AQ^{-1}A^\top)/\left(1+\rho\lambda_i(AQ^{-1}A^\top)\right)$, the convergence factor ζ_R is

$$\zeta_R(\rho,\alpha,\lambda_i(AQ^{-1}A^\top)) = 1-\alpha+\alpha\frac{\rho\lambda_i(AQ^{-1}A^\top)}{1+\rho\lambda_i(AQ^{-1}A^\top)}.$$

It is easy to check that the smallest value of $|\zeta_R|$ is obtained when $\alpha = 1$ and $\rho \to 0$. Since $\alpha = 1$ the relaxed ADMM iterations (5.21) coincide with (5.10) and consequently $\zeta = \zeta_R$.

5.A.13 Proof of Proposition 5.3

The proof follows similarly to the one of Proposition 5.2 but with $F^{(k+1)} = F^{(k)} = I$.

Chapter 6
Accelerating the ADMM algorithm: distributed quadratic problems

THIS chapter presents the parameter selection of the ADMM method for distributed quadratic programming. In this class of problems, a number of agents collaborate with neighbors in a graph to minimize a quadratic objective function over a combination of shared and private variables.

In particular, we consider equality-constrained quadratic problems, where the constraints enforce consistency among the local decision variables. By analyzing these equality-constrained quadratic programming problems, we are able to characterize the optimal step-size, over-relaxation and constraint scalings for the associated ADMM iterations.

The outline of this chapter is as follows. Section 6.1 discusses the related work. Section 6.2 illustrates how the ADMM method can be used to formulate distributed problems as equality-constrained optimization problems. The ADMM iterations for equality-constrained quadratic programming problems are formulated and analyzed in Section 6.3. Distributed quadratic programming and optimal networked-constrained scaling of the ADMM algorithm are addressed in Section 6.4. Numerical examples illustrating our results and comparing them to state-of-the-art techniques are presented in Section 6.5. Section 6.6 concludes the chapter.

6.1 Related work

Recently, a number of applications have triggered a strong interest in distributed algorithms for large-scale quadratic programming. Such applications are in areas as varied as multi-agent systems [107, 54], distributed model predictive control [25, 123], and state estimation in networks [106].

As these systems become larger and their complexity increases, the need for developing scalable and efficient algorithms become of central importance. As argued in the previous chapters, the ADMM method is a particularly powerful approach for structured problems. One attractive feature of ADMM is that it is guaranteed to converge for all (positive) values of its step-size parameter. This contrasts with many alternative techniques, such as dual decomposition, where mis-tuning of the step-size parameter for the gradient iterations can lead to divergence.

The ADMM method has been observed to converge fast in many distributed applications [45, 124, 125, 126, 118]. However, the solution times are sensitive to the choice of the step-size parameter, and as we observed in the previous chapter, when this parameter is not properly tuned, the ADMM iterations may converge (much) slower than the standard gradient algorithm. In practice, the ADMM algorithm parameters are tuned empirically for each specific application. For example, [124, 125, 126] propose different rules of thumb for picking the step-size for different distributed quadratic programming applications, and empirical results for choosing the best relaxation parameter can be found in [45]. However, a thorough analysis and design of the optimal step-size, relaxation parameter, and scaling rules for the distributed ADMM algorithm is still missing in the literature.

6.2 ADMM for distributed optimization

In this section, we describe an equality-constrained distributed optimization problem and discuss how it can be posed in ADMM standard format.

Consider a network of agents, each endowed with a local convex loss function $f_i(x)$, that collaborate to find the decision vector $x \in \mathbb{R}^{n_x}$ that results in a minimal total loss. In particular, recall problem (2.21) and it's equivalent locally separable form (2.22).

Moreover, recall from Chapter 2 that the interactions among agents are described by a connected undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. When the communication graph is connected, all equality constraints in (2.22) that do not correspond to neighboring nodes in $\mathcal{G}$ can be removed without altering the optimal solution. The remaining inequality constraints can be accounted for in different ways as described next (c.f. [54]).

6.2.1 Enforcing agreement with edge variables

One way to ensure agreement between all nodes is to enforce all pairs of nodes connected by an edge to have the same value, i.e., $x_i = x_j$ for all $\{i,j\} \in \mathcal{E}$. To include this constraint in the ADMM formulation, one can form the extended edge set $\bar{\mathcal{E}}$ (see Section 2.3) and introduce an auxiliary variable $z_{\{i,j\}}$ for the pair of edges (i,j), (j,i) in $\bar{\mathcal{E}}$. The local constraints $x_i = z_{\{i,j\}}$ and $x_j = z_{\{i,j\}}$ are then introduced for neighboring nodes i and j, and an equivalent form of (2.22) is formulated as

$$\begin{array}{ll} \underset{\{x_i\},\{z_{\{i,j\}}\}}{\text{minimize}} & \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & R_{(i,j)} x_i = R_{(i,j)} z_{\{i,j\}}, \quad \forall i \in \mathcal{V},\ \forall (i,j) \in \bar{\mathcal{E}}. \end{array} \tag{6.1}$$

Here, $R_{(i,j)} \in \mathbb{R}^{n_x \times n_x}$ acts as a scaling factor for the constraint defined along each edge $(i,j) \in \bar{\mathcal{E}}$ and $W_{(i,j)} \triangleq R_{(i,j)}^\top R_{(i,j)}$ is the weight of the edge $(i,j) \in \bar{\mathcal{E}}$. The edge weights $W_{(i,j)}$ are included to increase the degrees of freedom available for nodes to improve the performance of the algorithm. We will discuss optimal design of these constraint scalings in Section 6.3.2. Note that when the edge variables $z_{\{i,j\}}$ are fixed, (6.1) is separable and each agent i can find the optimal x_i without interacting with the other agents.

The optimization problem (6.1) can be written in the ADMM standard form as follows. Define $x = [x_1^\top \cdots x_{|\mathcal{V}|}^\top]^\top$, $z = [z_{e_1}^\top \cdots z_{e_{|\mathcal{E}|}}]^\top$, $f(x) = \sum_{i\in\mathcal{V}} f_i(x_i)$, and $\mathring{B} \in \mathbb{R}^{|\bar{\mathcal{E}}|\times|\mathcal{V}|}$ such that $\mathring{B}_{kj} = 1$ if j is the head of $\bar{e}_k \in \bar{\mathcal{E}}$ and $\mathring{B}_{kj} = 0$ otherwise. Problem (6.1) can then be rewritten as

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & f(x) \\ \text{subject to} & REx + RFz = 0, \end{array} \tag{6.2}$$

where

$$E = \mathring{B} \otimes I_{n_x}, \quad F = -\begin{bmatrix} I_{|\mathcal{E}|} \\ I_{|\mathcal{E}|} \end{bmatrix} \otimes I_{n_x}, \quad R = \operatorname{diag}(\{R_{\bar{e}_i}\}_{\bar{e}_i\in\bar{\mathcal{E}}}). \tag{6.3}$$

Note that the problems considered in this chapter, are in a particular form of the standard ADMM formulation (2.14) in which $g(z) = 0$.

6.2.2 Enforcing agreement with node variables

Another way of enforcing the agreement among the decision makers is via node variables. In this setup, each agent i has to agree with all the neighboring agents, including itself. In the ADMM formulation, this constraint is formulated as $x_i = z_j$ for all $j \in \mathcal{N}_i \cup \{i\}$, where $z_i \in \mathbb{R}^{n_x}$ is an auxiliary variable created per each node i. The optimization problem can be written as

$$\begin{array}{ll} \underset{\{x_i\},\{z_i\}}{\text{minimize}} & \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & R_{(i,j)}x_i = R_{(i,j)}z_j, \quad \forall i \in \mathcal{V},\ \forall j \in \{\mathcal{N}_i \cup \{i\}\}, \end{array} \tag{6.4}$$

where $R_{(i,j)} \in \mathbb{R}^{n_x\times n_x}$ and $W_{(i,j)} = R_{i,(i,j)}^\top R_{i,(i,j)}$ is the weight of the edge $(i,j) \in \bar{\mathcal{E}}$. Additionally, we also have $R_{(i,i)} \in \mathbb{R}^{n_x\times n_x}$ and define $W_{(i,i)} \triangleq R_{(i,i)}^\top R_{(i,i)}$ as the matrix-valued weight of the self-loop (i,i). Recall $\mathring{B}$ defined in the previous section, and define $\hat{B}$ such that $\hat{B}_{kj} = 1$ if j is the tail of $\bar{e}_k \in \bar{\mathcal{E}}$ and $\hat{B}_{kj} = 0$ otherwise. The distributed quadratic problem (6.4) can be rewritten as(6.2) with the constraints set

$$E = \begin{bmatrix} \mathring{B} \\ I_{|\mathcal{V}|} \end{bmatrix} \otimes I_{n_x}, \quad F = -\begin{bmatrix} \hat{B} \\ I_{|\mathcal{V}|} \end{bmatrix} \otimes I_{n_x}, \quad R = \operatorname{diag}(\{R_{\bar{e}_k}\}_{\bar{e}_k\in\bar{\mathcal{E}}},\ \{R_{(i,i)}\}_{i\in\mathcal{V}}). \tag{6.5}$$

6.3 ADMM for equality-constrained quadratic programming

In this section, we analyze and optimize scaled ADMM iterations for solving (6.2) with quadratic cost functions. Consider the following class of equality-constrained quadratic programming problems

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & \frac{1}{2}x^\top Qx + q^\top x + c^\top z \\ \text{subject to} & REx + RFz = 0. \end{array} \tag{6.6}$$

where $Q \in \mathcal{S}^n_{++}, q \in \mathbb{R}^n, c \in \mathbb{R}^m, E \in \mathbb{R}^{p\times n}$, and $F \in \mathbb{R}^{p\times m}$. We assume that E, and F have full-column rank. An important difference compared to the standard ADMM form is that the original constraints $Ex + Fz = 0$ have been scaled by a matrix $R \in \mathbb{R}^{r\times p}$.

Assumption 6.1 The scaling matrix R is chosen so that no non-zero vector v of the form $v = Ex + Fz$ belongs to the null-space of R.

In other words, after the scaling with R, the set of feasible solution to $Ex + Fz = 0$ is the same as the ones to $REx + RFz = 0$.

Our aim is to find the optimal scaling that minimizes the convergence factor of the corresponding ADMM iterations. In the next lemma we show that (6.6) can be converted to a more general form:

$$\begin{array}{ll} \underset{x,z}{\text{minimize}} & \frac{1}{2}x^\top Qx + d^\top x + c^\top z \\ \text{subject to} & REx + RFz = Rh, \end{array} \tag{6.7}$$

for some $h \in \mathbb{R}^{p\times 1}$.

Lemma 6.1
Let $(x^\star, z^\star)$ be the optimal solution to (6.6) and let $(\hat{x}, \hat{z})$ be any feasible solution to (6.7). Then the optimization problem (6.7) has the optimal solution $(x^\star + \hat{x}, z^\star + \hat{z})$ if the parameters q and d in (6.6) and (6.7) satisfy $q = Q\hat{x} + d$.

Proof. See Appendix 6.A for all the proofs of this chapter. □

Without loss of generality we thus assume $h = 0$ in the remainder of the chapter. Letting $\bar{E} = RE$ and $\bar{F} = RF$, the penalty term in the augmented Lagrangian becomes $\rho/2\|\bar{E}x + \bar{F}z - \bar{h}\|^2$. The scaled ADMM iterations for (6.6) with fixed relaxation parameter $\alpha^{(k)} = \alpha$ for all k then read

$$\begin{aligned} x^{(k+1)} =& (Q + \rho\bar{E}^\top\bar{E})^{-1}\left(-q - \rho\bar{E}^\top(\bar{F}z^{(k)} + u^{(k)})\right) \\ z^{(k+1)} =& -(\bar{F}^\top\bar{F})^{-1}\left(\bar{F}^\top\left(\alpha\bar{E}x^{(k+1)} - (1-\alpha)\bar{F}z^{(k)} + u^{(k)}\right) + c/\rho\right) \\ u^{(k+1)} =& u^{(k)} + \alpha\bar{E}x^{(k+1)} - (1-\alpha)\bar{F}z^{(k)} + \bar{F}z^{(k+1)}. \end{aligned} \tag{6.8}$$

Inserting the expression for $z^{(k+1)}$ in the u-update yields

$$u^{(k+1)} = \Pi_{\mathcal{N}(\bar{F}^\top)}\left(u^{(k)} + \alpha\bar{E}x^{(k+1)}\right) - \bar{F}(\bar{F}^\top\bar{F})^{-1}c/\rho.$$

Since $\mathcal{N}(\bar{F}^\top)$ and $\text{Im}(\bar{F})$ are orthogonal complements, this implies that $\Pi_{\text{Im}(\bar{F})}u^{(k)} = -\bar{F}(\bar{F}^\top\bar{F})^{-1}c/\rho$ for all k. Thus

$$\bar{F}z^{(k+1)} = (1-\alpha)\bar{F}z^{(k)} - \alpha\Pi_{\text{Im}(\bar{F})}\bar{E}x^{(k+1)}. \tag{6.9}$$

By inserting this expression in the u-update and applying the simplified iteration recursively, we find that

$$u^{(k+1)} = \Pi_{\mathcal{N}(\bar{F}^\top)} \left(u^{(0)} + \alpha \sum_{i=1}^{k+1} \bar{E}x^{(i)} \right) - \bar{F}(\bar{F}^\top \bar{F})^{-1} c/\rho. \tag{6.10}$$

We now apply (6.9) and (6.10) to eliminate u from the x-updates:

$$\begin{aligned} x^{(k+1)} &= \alpha\rho(Q+\rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top \left(\Pi_{\mathrm{Im}(\bar{F}^\top)} - \Pi_{\mathcal{N}(\bar{F}^\top)}\right)\bar{E}x^{(k)} \\ &\quad + x^{(k)} + \alpha\rho(Q+\rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\bar{F}z^{(k-1)}. \end{aligned} \tag{6.11}$$

Thus, using (6.11) and defining $\chi^{(k)} \triangleq \bar{E}^\top \bar{F} z^{(k)}$, the ADMM iterations can be rewritten in the following matrix form

$$\begin{bmatrix} x^{(k+1)} \\ \chi^{(k)} \end{bmatrix} = \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & (1-\alpha)I \end{bmatrix}}_{M} \begin{bmatrix} x^{(k)} \\ \chi^{(k-1)} \end{bmatrix}, \tag{6.12}$$

for $k \geq 1$ with $x^{(1)} = -(Q+\rho\bar{E}^\top\bar{E})^{-1}\left(q+\rho\bar{E}^\top(\bar{F}z^{(0)}+u^{(0)})\right)$, $\chi^0 = \bar{E}^\top\bar{F}z^{(0)}$, $z^{(0)} = -(\bar{F}^\top\bar{F})^{-1}c/\rho$, $u^{(0)} = \bar{F}z^{(0)}$, and

$$\begin{aligned} M_{11} &= \alpha\rho(Q+\rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\left(\Pi_{\mathrm{Im}(\bar{F})} - \Pi_{\mathcal{N}(\bar{F}^\top)}\right)\bar{E} + I, \\ M_{12} &= \alpha\rho(Q+\rho\bar{E}^\top\bar{E})^{-1}, \quad M_{21} = -\alpha\bar{E}^\top\Pi_{\mathrm{Im}(\bar{F})}\bar{E}. \end{aligned} \tag{6.13}$$

The next theorem shows how the convergence properties of the ADMM iterations are characterized by the spectral properties of the matrix M.

Theorem 6.1

Define $\sigma^{(k+1)} \triangleq [x^{(k+1)^\top} \; \chi^{(k)^\top}]^\top$, $s \triangleq \dim(\mathcal{X})$, where $\mathcal{X}$ is the feasibility subspace

$$\mathcal{X} \triangleq \left\{x \in \mathbb{R}^n,\, z \in \mathbb{R}^m \,|\, \bar{E}x + \bar{F}z = 0\right\},$$

and let $\{\varphi_i\}$ be the eigenvalues of M ordered so that $|\varphi_1| \leq \cdots \leq \cdots \leq |\varphi_{2n}|$. Then $s = \dim\left(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\right)$ and the ADMM iterations (6.8) converge linearly to the optimal solution of (6.6) if and only if $s \geq 1$ and $|\varphi_{2n-s}| < \varphi_{2n-s+1} = \cdots = \varphi_{2n} = 1$. Moreover, the convergence factor of the ADMM iterates in terms of the sequence $\{\sigma^{(k)}\}$ equals $|\varphi_{2n-s}|$.

Let us discuss the results of Theorem 6.1 by an example. Consider a problem of the form (2.21), in which a network of $|\mathcal{V}|$ agents, each endowed with a local strongly convex quadratic loss function, collaborate to find a n_x-dimensional decision vector that results in a minimal total loss. For this example, one can use either edge-variable or node-variable formulation to obtain the standard ADMM problem (6.2) with $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$ where

$n = |\mathcal{V}| \cdot n_x$. It can be easily verified that the overall problem is feasible[1], and that the dimension of the feasibility subspace equals to $s = n_x$. Then Theorem 6.1 specifies that the corresponding ADMM algorithm (6.8) converges linearly toward the optimum point. Moreover, M takes n_x unit eigenvalues and the convergence factor of the ADMM algorithm is given by $|\varphi_{2n-n_x}| < 1$.

Below we state the main problem to be addressed in the remainder of this chapter.

Problem 6.1 What values of ρ, α, and R minimize $|\varphi_{2n-s}|$, the convergence factor of the ADMM iterates?

As the initial step to tackle Problem 6.1, we characterize, φ_i, the eigenvalues of M. Our analysis will be simplified by choosing an R that satisfies the following assumption.

Assumption 6.2 The scaling matrix R is such that $E^\top R^\top R E = \bar{E}^\top \bar{E} = \kappa Q$ for some $\kappa > 0$ and $\bar{E}^\top \bar{E} \succ 0$.

Remark 6.1 The eigenvalues of M are given by a Quadratic Eigenvalue Problem (QEP) [127]. Characterizing the spectral properties of a QEP is typically a daunting task and requires numerical methods. With Assumption 6.2, However, we are able to exploit some structural properties of matrix M that help us to analytically characterize its eigenvalues. Moreover, Assumption 6.2 may appear restrictive at first sight, but we will later describe several techniques for finding such an R, even for the distributed setting outlined in Section 6.2. Finally, we note that our scaling parameter R has two purposes. First, it is employed to satisfy Assumption 6.2 and, the second, it is used to speedup the ADMM algorithm.

Replacing $\bar{E}^\top \bar{E} = \kappa Q$ in (6.13) and using the identity

$$\Pi_{\text{Im}(\bar{F})} - \Pi_{\mathcal{N}(\bar{F}^\top)} = 2\Pi_{\text{Im}(\bar{F})} - I,$$

yields

$$\begin{aligned} M_{11} &= \alpha \frac{\rho\kappa}{1+\rho\kappa} (\bar{E}^\top \bar{E})^{-1} \bar{E}^\top \left(2\Pi_{\text{Im}(\bar{F})} - I\right) \bar{E} + I, \\ M_{12} &= \alpha \frac{\rho\kappa}{1+\rho\kappa} (\bar{E}^\top \bar{E})^{-1}, \quad M_{21} = -\alpha \bar{E}^\top \Pi_{\text{Im}(\bar{F})} \bar{E}. \end{aligned}$$

These expressions allow us to explicitly characterize the eigenvalues of M in (6.12).

Theorem 6.2
Consider the ADMM iterations (6.12) and suppose that $\bar{E}^\top \bar{E} = \kappa Q$. Let v_i be a generalized eigenvector of $\left(\bar{E}^\top \left(2\Pi_{\text{Im}(\bar{F})} - I\right) \bar{E},\ \bar{E}^\top \bar{E}\right)$ with associated generalized eigenvalue λ_i. Then, M has two right eigenvectors on the form $[v_i^\top\ w_{i1}^\top]^\top$ and $[v_i^\top\ w_{i2}^\top]^\top$ whose associated eigenvalues φ_{i1} and φ_{i2} are the solutions to the quadratic equation

[1]In fact, we know much more about the centralize problem in this example. One can analytically find the optimal primal-dual variables $(x^\star, z^\star, y^\star)$ by solving the KKT system obtained from the Lagrangian of the centralized problem of the form (6.2) with quadratic cost function.

$$\varphi_i^2 + a_1(\lambda_i)\varphi_i + a_0(\lambda_i) = 0, \tag{6.14}$$

where

$$a_1(\lambda_i) \triangleq \alpha - \alpha\beta\lambda_i - 2, \quad a_0(\lambda_i) \triangleq \alpha\beta(1 - \frac{\alpha}{2})\lambda_i + \frac{1}{2}\alpha^2\beta + 1 - \alpha, \tag{6.15}$$

and

$$\beta \triangleq \frac{\rho\kappa}{(1 + \rho\kappa)}.$$

From (6.14) and (6.15) one directly sees that α, ρ (or, equivalently, β), and R affect the eigenvalues of M. We will use $\varphi(\alpha, \beta, \lambda_i)$ to emphasize this dependence. In the next section we study the properties of (6.14) with respect to β, α, and λ_i.

6.3.1 Optimal parameter selection

To minimize the convergence factor of the iterates (6.8), we combine Theorem 6.1, which relates the convergence factor of the ADMM iterates to the spectral properties of the matrix M, with Theorem 6.2, which gives explicit expressions for the eigenvalues of M in terms of the ADMM parameters. The following result is useful for the development of our analysis.

Proposition 6.1 (Jury's stability test [128])
The quadratic polynomial $a_2\varphi_i^2 + a_1\varphi_i + a_0$ with real coefficients $a_2 > 0$, a_1, and a_0 has its roots inside the unit-circle, i.e., $|\varphi_i| < 1$, if and only if the following three conditions hold:

$$\begin{aligned} &\text{i)} && a_0 + a_1 + a_2 > 0; \\ &\text{ii)} && a_2 > a_0; \\ &\text{iii)} && a_0 - a_1 + a_2 > 0. \end{aligned}$$

The next sequence of lemmas derive some useful properties of λ_i and of the eigenvalues of M.

Lemma 6.2
The generalized eigenvalues of $(\bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E},\ \bar{E}^\top \bar{E})$ are real scalars in $[-1, 1]$.

Lemma 6.3
Let λ_i be the i-th generalized eigenvalue of $(\bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E},\ \bar{E}^\top \bar{E})$, ordered as $\lambda_n \geq \cdots \geq \lambda_i \geq \cdots \geq \lambda_1$ and let $\dim\left(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\right) = s$. If the optimization problem (6.6) is feasible, we have $s \geq 1$ and $\lambda_i = 1$, for all $i = n, \ldots, n - s + 1$.

Lemma 6.4
Consider the eigenvalues $\{\varphi_i\}$ of the matrix M in (6.12), ordered as $|\varphi_{2n}| \geq \cdots \geq |\varphi_i| \geq \cdots \geq |\varphi_1|$. Then $\varphi_{2n} = \cdots = \varphi_{2n-s+1} = 1$ where $s = \dim\left(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\right)$. Moreover, for $\beta \in (0,\ 1)$ and $\alpha \in (0,\ 2]$ we have $|\varphi_i| < 1$ for $i \leq 2n - s$.

Lemma 6.4 and Theorem 6.1 establish that the convergence factor of the ADMM iterates, $|\varphi_{2n-s}|$, is strictly less than 1 for $\beta \in (0,\ 1)$ and $\alpha \in (0,\ 2]$. Next, we characterize $|\varphi_{2n-s}|$ explicitly in terms of α, β and λ_i.

Theorem 6.3

Consider the eigenvalues $\{\varphi_i\}$ of M ordered as in Lemma 6.4. For fixed $\alpha \in (0,\ 2]$ and $\beta \in (0,\ 1)$, the magnitude of φ_{2n-s} is given by

$$|\varphi_{2n-s}| \triangleq \max\left\{g_r^+,\ g_r^-,\ g_c,\ g_1\right\} \tag{6.16}$$

where

$$\begin{aligned}
g_r^+ &\triangleq 1 + \frac{\alpha}{2}\beta\lambda_{n-s} - \frac{\alpha}{2} + \frac{\alpha}{2}\sqrt{\lambda_{n-s}^2\beta^2 - 2\beta + 1 + s_r^+},\\
g_r^- &\triangleq -1 - \frac{\alpha}{2}\beta\lambda_1 + \frac{\alpha}{2} + \frac{\alpha}{2}\sqrt{\lambda_1^2\beta^2 - 2\beta + 1 + s_r^-},\\
g_c &\triangleq \sqrt{\frac{1}{2}\alpha^2\beta(1-\lambda_{n-s}) + 1 - \alpha + \alpha\beta\lambda_{n-s} + s_c},\\
g_1 &\triangleq |1 - \alpha(1-\beta)|,\\
s_r^+ &\triangleq \max\{0,\ -(\beta^2\lambda_{n-s}^2 - 2\beta + 1)\},\\
s_r^- &\triangleq \max\{0,\ -(\beta^2\lambda_1^2 - 2\beta + 1)\},\\
s_c &\triangleq \max\{0,\ -a_0(\lambda_{n-s})\}.
\end{aligned} \tag{6.17}$$

Moreover, we have $|\varphi_{2n-s}| > g_r^+$, $|\varphi_{2n-s}| > g_r^-$, and $|\varphi_{2n-s}| > g_c$ if $s_r^+ > 0$, $s_r^- > 0$, and $s_c > 0$, respectively.

Given the latter result, the problem of minimizing $|\varphi_{2n-s}|$ with respect to α and β can be written as

$$\min_{\alpha\in(0,\,2],\ \beta\in(0,\,1)} \max\left\{g_r^+,\ g_r^-,\ g_c,\ g_1\right\}.$$

Numerical studies have suggested that under-relaxation, i.e., letting $\alpha < 1$, does not improve the convergence speed of ADMM, see, e.g., [45]. The next result establishes formally that this is indeed the case for our considered class of problems.

Proposition 6.2

Let $\beta \in (0,\ 1)$ be fixed and consider $\varphi_{2n-s}(\alpha, \beta)$. For $\alpha < 1$, it holds

$$|\varphi_{2n-s}(1, \beta)| < |\varphi_{2n-s}(\alpha, \beta)|.$$

The main result presented below provides explicit expressions for the optimal parameters α and β that minimize $|\varphi_{2n-s}|$ over given intervals.

Theorem 6.4

Consider the optimization problem (6.6) under Assumption 6.2 and its associated ADMM iterates (6.12). The parameters $\alpha^\star$ and $\beta^\star$ that minimize the convergence factor $|\varphi_{2n-s}|$ over $\alpha \in (0, \alpha^\star]$ and $\beta \in (0, 1)$ are:

Case I : if $\lambda_{n-s} > 0$ and $\lambda_{n-s} \geq |\lambda_1|$,

$$\beta^\star = \frac{1 - \sqrt{1-\lambda_{n-s}^2}}{\lambda_{n-s}^2}, \quad \alpha^\star = 2, \quad |\varphi_{2n-s}| = \frac{1 - \sqrt{1-\lambda_{n-s}^2}}{\lambda_{n-s}}; \tag{6.18}$$

Case II : if $|\lambda_1| \geq \lambda_{n-s} > 0$,

$$\beta^\star = \frac{1 - \sqrt{1-\lambda_{n-s}^2}}{\lambda_{n-s}^2}, \quad \alpha^\star = \frac{4}{2 - \left(\lambda_{n-s} + \lambda_1 - \sqrt{\lambda_1^2 - \lambda_{n-s}^2}\right)\beta^\star},$$
$$|\varphi_{2n-s}| = 1 + \frac{\alpha^\star}{2}\lambda_{n-s}\beta^\star - \frac{\alpha^\star}{2}; \tag{6.19}$$

Case III : if $0 \geq \lambda_{n-s} \geq \lambda_1$,

$$\beta^\star = \frac{1}{2}, \quad \alpha^\star = \frac{4}{2-\lambda_1}, \quad |\varphi_{2n-s}| = \frac{-\lambda_1}{2-\lambda_1}. \tag{6.20}$$

Considering the standard ADMM iterations with $\alpha = 1$, the next result immediately follows.

Corollary 6.1

For $\alpha = 1$, the optimal $\beta^\star$ that minimizes the convergence factor $|\varphi^\star_{2n-s}|$ is

$$\beta^\star = \begin{cases} \dfrac{1 - \sqrt{1-\lambda_{n-s}^2}}{\lambda_{n-s}^2} & \lambda_{n-s} > 0, \\ \dfrac{1}{2} & \lambda_{n-s} \leq 0. \end{cases} \tag{6.21}$$

Moreover, the corresponding convergence factor is

$$|\varphi^\star_{2n-s}| = \begin{cases} \dfrac{1}{2}\left(1 + \dfrac{\lambda_{n-s}}{1+\sqrt{1-\lambda_{n-s}^2}}\right) & \lambda_{n-s} > 0, \\ \dfrac{1}{2} & \lambda_{n-s} \leq 0. \end{cases} \tag{6.22}$$

6.3.2 Optimal constraint scaling

As seen in Theorem 6.4, the convergence factor of ADMM depends in a piecewise fashion on λ_{n-s} and λ_1. In Case I and Case II of Theorem 6.4, the convergence factor is monotonically increasing with respect to λ_{n-s}, and it makes sense to choose the constraint scaling matrix R to minimize λ_{n-s} while satisfying the structural constraint imposed by Assumption 6.2. To formulate the selection of R as a quasi-convex optimization problem, we first enforce the constraint $\kappa Q = E^\top W E$ by using the following result.

Lemma 6.5

Consider the optimization problem (6.6) with $Q \succeq 0$ and let $P \in \mathbb{R}^{n\times s}$ be an orthonormal basis for $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$. Let $W = R^\top R$ and assume that $E^\top W E \succ 0$. If $P^\top E^\top W E P = P^\top Q P \succ 0$, then the optimal solution to (6.6) remains unchanged when Q is replaced with $E^\top W E$.

The following result provides necessary and sufficient conditions for Assumption 6.1 to hold.

Lemma 6.6

Let P_1 be an orthonormal basis for the orthogonal complement to $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$ and define $W = R^\top R \succeq 0$. The following statements are true:

i) Assumption 6.1 holds if and only if $F^\top W F \succ 0$ and $P_1^\top \bar{E}^\top \Pi_{\mathcal{N}(\bar{F}^\top)} \bar{E} P_1 \succ 0$;

ii) If Assumption 6.1 holds, then $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E) = \mathcal{N}(\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E})$.

Next, we derive a tight upper bound on λ_{n-s}.

Lemma 6.7

Let P_1 be an orthonormal basis for the orthogonal complement to $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$. Defining $W = R^\top R \succeq 0$ and letting $\lambda \le 1$, we have $\lambda \ge \lambda_{n-s}$ if and only if

$$\begin{bmatrix} (\lambda+1)P_1^\top E^\top W E P_1 & P_1^\top E^\top W F \\ F^\top W E P_1 & \frac{1}{2}F^\top W F \end{bmatrix} \succ 0. \tag{6.23}$$

Moreover, Assumption 6.1 holds for a given W satisfying (6.23) with $\lambda \le 1$.

Using the previous results, the matrix W minimizing λ_{n-s} can be computed as follows.

Theorem 6.5

Let $P_1 \in \mathbb{R}^{n\times n-s}$ be an orthonormal basis for the orthogonal complement to $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$, define $P \in \mathbb{R}^{n\times s}$ as an orthonormal basis for $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$, and denote $\mathcal{A}$ as a given sparsity pattern. The matrix $W = R^\top R \in \mathcal{A}$ that minimizes λ_{n-s} while satisfying Assumptions 6.2 and 6.1 is the solution to the quasi-convex optimization problem

$$\begin{aligned} \underset{W,\lambda}{\text{minimize}} \quad & \lambda \\ \text{subject to} \quad & W \in \mathcal{A},\ W \succeq 0,\ (6.23), \\ & P^\top E^\top W E P = P^\top Q P. \end{aligned} \tag{6.24}$$

The results derived in the current section contribute to improve the convergence properties of the ADMM algorithm for equality-constrained quadratic programming problems. The procedure to determine suitable choices of ρ, α, and R is summarized in Algorithm 6.1.

Algorithm 6.1 Optimal Constraint Scaling and Parameter Selection

1. Compute $W^\star$ and the corresponding λ_{n-s} and λ_1 according to Theorem 6.5;
2. Use Lemma 6.5 to replace Q with $E^\top W E$ and let $\kappa = 1$;
3. Given λ_{n-s} and λ_1, use the ADMM parameters $\rho^\star = \frac{\beta^\star}{1-\beta^\star}$ and $\alpha^\star$ according to Theorem 6.4.

6.4 ADMM for distributed quadratic programming

We are now ready to develop optimal scalings for the ADMM iterations for distributed quadratic programming. Specifically, we consider (2.22) with $f_i(x_i) = (1/2)x_i^\top Q_i x_i + q_i^\top x_i$ and $Q_i \succ 0$ and use the results derived in the previous section to derive optimal algorithm parameters for the ADMM iterations in both edge- and node-variable formulations.

6.4.1 Enforcing agreement with edge variables

In the edge variable formulation, we introduce auxiliary variables $z_{\{i,j\}}$ for each edge $\{i,j\} \in \mathcal{E}$ and re-write the optimization problem in the form of (6.1). The resulting ADMM iterations for node i can be written as

$$\begin{aligned}
x_i^{(k+1)} &= \underset{x_i}{\operatorname{argmin}}\ \frac{1}{2}x_i^\top Q_i x_i + q_i^\top x_i + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i} \|R_{(i,j)}x_i - R_{(i,j)}z_{\{i,j\}}^{(k)} + R_{(i,j)}u_{(i,j)}^{(k)}\|^2,\\
\gamma_{(j,i)}^{(k+1)} &= \alpha x_j^{(k+1)} + (1-\alpha)z_{\{i,j\}}^{(k)}, \quad \forall j \in \mathcal{N}_i,\\
z_{\{i,j\}}^{(k+1)} &= \underset{z_{\{i,j\}}}{\operatorname{argmin}} \Big\{ \|R_{(i,j)}\gamma_{(i,j)}^{(k+1)} + R_{(i,j)}u_{(i,j)}^{(k)} - R_{(i,j)}z_{\{i,j\}}\|^2 \\
&\qquad + \|R_{(j,i)}\gamma_{(j,i)}^{(k+1)} + R_{(j,i)}u_{(j,i)}^{(k)} - R_{(j,i)}z_{\{i,j\}}\|^2 \Big\},\\
u_{(i,j)}^{(k+1)} &= u_{(i,j)}^{(k)} + \gamma_{(i,j)}^{(k+1)} - z_{\{i,j\}}^{(k+1)}.
\end{aligned} \tag{6.25}$$

Here, $u_{(i,j)}$ is the scaled Lagrange multiplier, private to node i, associated with the constraint $R_{(i,j)}x_i = R_{(i,j)}z_{\{i,j\}}$, and the variables $\gamma_{(i,j)}$ have been introduced to write the iterations in a more compact form. Note that the algorithm is indeed distributed, since each node i only needs the current iterates $x_j^{(k+1)}$ and $u_{(j,i)}^{(k)}$ from its neighboring nodes $j \in \mathcal{N}_i$.

We can also re-write the problem formulation as an equality constrained quadratic program on the form (6.2) with $f(x) = (1/2)x^\top Q x + q^\top x$, $Q = \operatorname{diag}\big(\{Q_i\}_{i\in\mathcal{V}}\big)$, and $q^\top = [q_1^\top \dots q_{|\mathcal{V}|}^\top]$. As shown in Section 6.3, the associated ADMM iterations can be written in vector form (6.8) and the step-size and the relaxation parameter that minimize the convergence factor of the iterates are given in Theorem 6.4.

Recall the assumptions that $W \succeq 0$ is chosen so that $E^\top W E = \kappa Q$ for $\kappa > 0$. The next result shows that such assumptions can be satisfied locally by each node.

Proposition 6.3

Consider the distributed optimization problem described by (6.2) and (6.3) and let $W = R^\top R$. The equation $E^\top W E = \kappa Q$ can be ensured for any $\kappa > 0$ by following a weight-assignment scheme satisfying the local constraints $\sum_{j \in \mathcal{N}_i} W_{(i,j)} = \kappa Q_i$ for all $i \in \mathcal{V}$.

Next, we analyze in more detail the scalar case with symmetric edge weights.

Scalar case

Consider the scalar case $n_x = 1$ with $n = |\mathcal{V}|$ and let the edge weights be symmetric with $W_{(i,j)} = W_{(j,i)} = w_{\{i,j\}} \geq 0$ for all $(i,j) \in \bar{\mathcal{E}}$. As derived in Section 6.3, the ADMM iterations can be written in matrix form as (6.12). Exploiting the structure of E and F, we derive

$$M_{11} = \alpha\rho(Q + \rho D)^{-1} A + I, \quad M_{12} = \alpha\rho(Q+\rho D)^{-1}, \quad M_{21} = -\frac{\alpha}{2}(D + A). \tag{6.26}$$

The optimal step-size $\rho^\star$ and relaxation parameter $\alpha^\star$ that minimizes the convergence factor $|\varphi_{2n-1}|$ are given in Theorem 6.4, where the eigenvalues $\{\lambda_i\}$ in the corresponding theorems are the set of ordered generalized eigenvalues of (A, D). Here, we briefly comment on the relationship between the generalized eigenvalues of (A, D) and the eigenvalues of the normalized Laplacian, $\mathcal{L}_{\text{norm}} = I - D^{-1/2} A D^{-1/2}$. In particular, we have $1 - \lambda = \psi$, where ψ is any eigenvalue of the normalized Laplacian and λ is a generalized eigenvalue of (A, D) corresponding to ψ. For certain well-known classes of graphs the value of the eigenvalues of the normalised Laplacian are known, e.g. see [60] and [129] for more information. As a result, one can identify which case of Theorem 6.4 is applied to each of these graphs. The following proposition establishes one such result.

Proposition 6.4

Adopt the hypothesis of Theorem 6.4. The following statements are true.

i) Case II of Theorem 6.4 holds for path graphs with $|\mathcal{V}| \geq 4$, cycle graphs with $|\mathcal{V}| \geq 5$, and wheel-graphs with $|\mathcal{V}| \geq 6$.

ii) Case III of Theorem 6.4 holds for complete graphs, bi-partite graphs, star graphs, path graphs with $|\mathcal{V}| = 3$, cycle graphs with $|\mathcal{V}| \in \{3, 4\}$ and wheel-graphs with $|\mathcal{V}| \in \{4, 5\}$.

Figure 6.1 illustrates some of the topologies that are mentioned in Proposition 6.4. For general topologies, without computing the generalized eigenvalues it is not easy to know which case of Theorem 6.4 applies. Moreover, when we use non-unity edge-weights, optimizing these for one case might alter the generalized eigenvalues so that another case applies. In extensive simulations, we have found that a good heuristic is to use scalings that attempt to reduce the magnitude of both the smallest and the second-largest generalized

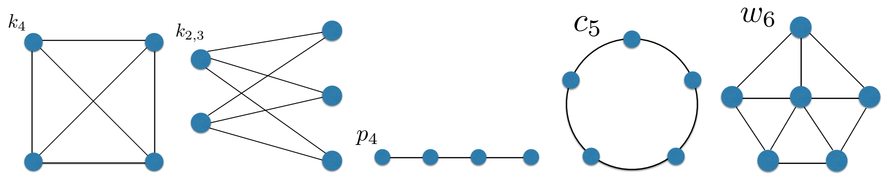


Figure 6.1: From left to right: a complete graph with $|\mathcal{V}| = 4$, a complete bi-partite graph with $|\mathcal{V}| = 5$, a path graph with $|\mathcal{V}| = 4$, a cycle graph with $|\mathcal{V}| = 4$, and a wheel graph with $|\mathcal{V}| = 6$ nodes.

eigenvalues. The next lemma shows how to compute such scalings for the edge-variable formulation.

Lemma 6.8

Consider the weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$. The non-negative edge-weights $\{w_{\{i,j\}}\}$ that jointly minimize and maximize the second largest and smallest generalized eigenvalue of (A, D), λ_{n-1} and λ_1, are obtained from the optimal solution to the quasi-convex problem

$$\begin{array}{lll}
\underset{\{w_{\{i,j\}}\},\lambda}{\text{minimize}} & \lambda & \\
\text{subject to} & w_{\{i,j\}} \geq 0, & \forall\, i,j \in \mathcal{V}, \\
& A_{ij} = w_{\{i,j\}}, & \forall\, \{i,j\} \in \mathcal{E}, \\
& A_{ij} = 0, & \forall\, \{i,j\} \notin \mathcal{E}, \\
& D = \operatorname{diag}(A\mathbf{1}_n), & \\
& D \succ \epsilon I, & \\
& P^\top (A - \lambda D)\, P \prec 0, & \\
& A + \lambda D \succ 0, &
\end{array} \tag{6.27}$$

where the columns of $P \in \mathbb{R}^{n \times n-1}$ form an orthonormal basis of $\mathcal{N}(\mathbf{1}_n^\top)$ and $\epsilon \in \mathbb{R}_{++}$.

6.4.2 Enforcing agreement with node variables

Recall the node variable formulation (6.4) using the auxiliary variables $z_i \in \mathbb{R}^{n_x}$ for each node $i \in \mathcal{V}$ described in Section 6.2.2. The ADMM iterations for node i can be rewritten as

$$\begin{aligned}
x_i^{(k+1)} &= \underset{x_i}{\operatorname{argmin}}\ \frac{1}{2} x_i^\top Q_i x_i + q_i^\top x_i + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i \cup \{i\}} \|R_{(i,j)} x_i - R_{(i,j)} z_j^{(k)} + R_{(i,j)} u_{(i,j)}^{(k)}\|^2, \\
\gamma_{(j,i)}^{(k+1)} &= \alpha x_j^{(k+1)} + (1-\alpha) z_i^{(k)}, \qquad \forall j \in \mathcal{N}_i \cup \{i\}, \\
z_i^{(k+1)} &= \underset{z_i}{\operatorname{argmin}} \sum_{j \in \mathcal{N}_i \cup \{i\}} \|R_{(j,i)} \gamma_{(j,i)}^{(k+1)} + R_{(j,i)} u_{(j,i)}^{(k)} - R_{(j,i)} z_i\|^2, \\
u_{(i,j)}^{(k+1)} &= u_{(i,j)}^{(k)} + \gamma_{(i,j)}^{(k+1)} - z_j^{(k+1)}, \qquad \forall j \in \mathcal{N}_i \cup \{i\},
\end{aligned} \tag{6.28}$$

where $u_{(i,j)}$ is the scaled Lagrange multiplier, private to node i, associated with the constraint $R_{(i,j)}x_i = R_{(i,j)}z_j$, and $\gamma_{(i,j)}^{(k)}$ is an auxiliary variable private to node i and associated with the edge (i,j). Note that the algorithm is distributed, since it only requires communication between neighbors. However, unlike the previous formulation with edge variables, here two communication rounds must take place: the first to exchange the private variables $x_j^{(k+1)}$ and $u_{(j,i)}^{(k)}$, required for the z_i-update; the second to exchange the private variables $z_j^{(k+1)}$, required for the $u_{(i,j)}$- and x_i-updates.

Let $x = [x_1^\top \cdots x_{|\mathcal{V}|}^\top]^\top$, $z = [z_1^\top \cdots z_{|\mathcal{V}|}^\top]^\top$, $Q = \operatorname{diag}\left(\{Q_i\}_{i\in\mathcal{V}}\right)$, and $q^\top = [q_1 \, \dots \, q_{|\mathcal{V}|}]$. The cost function in (6.4) takes the form $f(x) = (1/2)x^\top Qx + q^\top x$ while E and F are given in (6.5). Note that the matrix E is the same as for the edge-variable case, thus Proposition 6.3 may also be applied to the present formulation to ensure $E^\top WE = \kappa Q$.

Next we consider the scalar case with symmetric weights.

Scalar case

Consider the scalar case $n_x = 1$ with $n = |\mathcal{V}|$ and let the edges weights be symmetric with $W_{(i,j)} = W_{(j,i)} = w_{\{i,j\}} \geq 0$ for all $(i,j) \in \bar{\mathcal{E}}$. Using the structure of E and F, the fixed point equation (6.12) can be formulated by the following relations

$$\begin{aligned} M_{11} &= \alpha\rho(Q+\rho D)^{-1}(2AD^{-1}A - D) + I, \\ M_{12} &= \alpha\rho(Q+\rho D)^{-1}, \quad M_{21} = -\alpha AD^{-1}A. \end{aligned} \tag{6.29}$$

The optimal step-size $\rho^\star$ and relaxation parameter $\alpha^\star$ that minimizes the convergence factor $|\varphi_{2n-s}|$ are given in Theorem 6.4, where the eigenvalues $\{\lambda_i\}$ are the set of ordered generalized eigenvalues of $(2AD^{-1}A - D,\ D)$. For the node-variable formulation, we have not been able to formulate a weight optimization corresponding to Lemma 6.8. However, in the numerical evaluations, we will propose a modification that ensures that Case I of Theorem 6.4 applies, and then minimize the second-largest generalized eigenvalue.

6.5 Numerical examples

In this section, we illustrate our results via numerical examples. In particular, we consider a toy example with three nodes collaborating in a path topology trying to agree on a global variable. We demonstrate how our parameter selection rules can accelerate the decision making process. Our second application is devoted to distributed averaging where we compare ADMM with several state-of-the-art techniques.

6.5.1 Distributed quadratic programming

As a first example, we consider a distributed quadratic programming problem with 3 agents, a decision vector $x \in \mathbb{R}^4$, and an objective function on the form

$$f(x) = \sum_{i\in\mathcal{V}} 1/2x^\top Q_i x + q_i^\top x$$

with

$$Q_1 = \begin{bmatrix} 0.4236 & -0.0235 & -0.0411 & 0.0023 \\ -0.0235 & 0.0113 & 0.0023 & -0.0001 \\ -0.0411 & 0.0023 & 0.4713 & -0.0262 \\ 0.0023 & -0.0001 & -0.0262 & 0.0115 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 0.8417 & -0.1325 & -0.0827 & 0.0132 \\ -0.1325 & 0.0311 & 0.0132 & -0.0021 \\ -0.0827 & 0.0132 & 0.9376 & -0.1477 \\ 0.0132 & -0.0021 & -0.1477 & 0.0335 \end{bmatrix}$$

$$Q_3 = \begin{bmatrix} 0.0122 & 0.0308 & -0.0002 & -0.0031 \\ 0.0308 & 0.4343 & -0.0031 & -0.0422 \\ -0.0002 & -0.0031 & 0.0125 & 0.0344 \\ -0.0031 & -0.0422 & 0.0344 & 0.4833 \end{bmatrix}$$

$$q_1 = q_2 = 0,\ q_3^\top = \begin{bmatrix} -0.1258 & 0.0087 & 0.0092 & -0.1398 \end{bmatrix}.$$

The communication graph is a line graph where node 2 is connected to nodes 1 and 3. The distributed optimization problem is formulated using edge variables and solved by executing the resulting ADMM iterations. The convergence behavior of the iterates for different choices of scalings and algorithm parameters are presented in Figure 6.1.

The optimal constraint scaling matrix and ADMM parameters are computed using Algorithm 6.1, resulting in $\rho = 1$ and $\alpha = 1.33$. In the "local" algorithm, the nodes determined the constraint scalings in a distributed manner in accordance to Proposition 6.3, while the optimal parameters computed using Theorem 6.4 are $\rho = 1.44$ and $\alpha = 1.55$. The remaining iterations correspond to ADMM algorithms with unitary edge weights, fixed relaxation parameter α, and manually optimized step-size ρ. The parameter α is fixed at 1.0, 1.5, and 1.8, while the corresponding ρ is chosen as the empirical best.

Figure 6.1 shows that the manually tuned ADMM algorithm exhibits worse performance than the optimally and locally scaled algorithms. Here, the best parameters for the scaled versions are computed systematically using the results derived earlier, while the best parameters for the unscaled algorithms are computed through exhaustive search.

6.5.2 Distributed averaging

In this section we apply our methodology to derive optimally scaled ADMM iterations for a particular problem instance usually referred to as average consensus. The problem amounts to devising a distributed algorithm that ensures that all agents $i \in \mathcal{V}$ in a network reach agreement on the network-wide average of local values $x_i \in \mathbb{R}$ held by the individual agents. This problem can be formulated as a particular case of (2.22). That is the problem (4.19) which was previously studied in chapter 4.

We consider edge-variable and node-variable formulations and compare the performance of the corresponding ADMM iterates with the relevant state-of-the-art algorithms. As performance indicator, we use the convergence factors computed as the second largest eigenvalue of the linear fixed point iterations associated with each method. We generated communication graphs from the Erdős-Rényi and the Random Geometric Graph (RGG) families (see, e.g., [130]). Having generated $|\mathcal{V}|$ number of nodes, in Erdős-Rényi graphs we

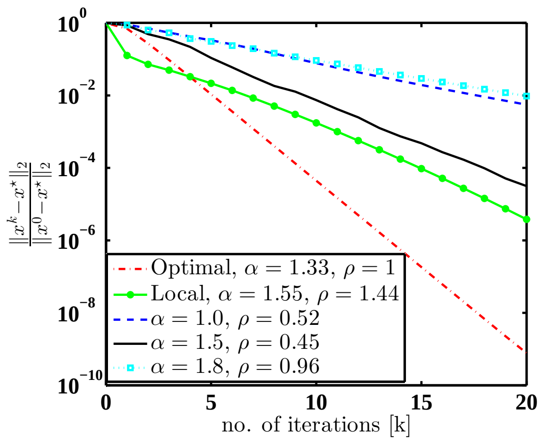


Figure 6.1: Normalized error for the scaled ADMM algorithm with $W^\star$ from Theorem 6.5, local scaling from Proposition 6.3, and unitary edge weights with fixed over-relaxation parameter α. The ADMM parameters for the scaled algorithms are computed from Theorem 6.4. The step-sizes for the unscaled algorithms are empirically chosen.

connected each pair of nodes with probability $p = (1+\epsilon)\log(|\mathcal{V}|)/|\mathcal{V}|$ where $\epsilon \in (0,1)$. In RGG, $|\mathcal{V}|$ nodes were randomly deployed in the unit square and an edge was introduced between each pair of nodes whose inter-distance is at most $2\log(|\mathcal{V}|)/|\mathcal{V}|$; this guarantees that the graph is connected with high probability [131].

Figure 6.2 presents Monte Carlo simulations of the convergence factors versus the number of nodes $|\mathcal{V}| \in [10, 50]$. Each data point is the average convergence factor in 60 instances of randomly generated graphs with the same number of nodes. In our simulations, we consider both edge-variable and node-variable formulations. For both formulations, we consider three versions of the ADMM algorithm with our parameter settings: the standard one (with step-size given in Corollary 6.1), an over-relaxed version with parameters in Theorem 6.4, and the scaled-relaxed-ADMM that uses weight optimization in addition to the optimal parameters in Theorem 6.4.

In the edge-variable scenario, we compare the ADMM iterates to three other algorithms: fast-consensus [54] from the ADMM literature and two state-of-the-art algorithms from the literature on the accelerated consensus: Oreshkin et al. [101] from the shift-register type of acceleration and the multi-step method developed in Section 4.6.2 (and termed here as Ghadimi et al.). In these algorithms, a two-tap memory mechanism is implemented so that the values of two last iterates are taken into account when computing the next. All the averaging algorithms in our experiment employ their best known weight scheme. For Ghadimi et al., the optimal weight is given by Proposition 4.3 while fast-consensus and Oreshkin et al. use the optimal weights in [47]. The scaled-relaxed-ADMM method employs the weight heuristic presented in Lemma 6.8. Figures 6.2a, 6.2c and 6.2e show a significant improvement of our design rules compared to the alternatives for both RGG and Erdős-Rényi graphs in sparse ($\epsilon = 0.2$) and dense ($\epsilon = 0.8$) topologies. We observe that in all cases, the convergence factor decreases with increasing network size on RGG, while it stays almost constant on Erdős-Rényi graphs.

For the node-variable formulation, we compare the three variants of our ADMM algorithm to the fast-consensus [54] algorithm. The reason why we exclude two other methods from the comparison is that they do not (yet) exist for the node-variable formulation. By comparing their explicit x-updates in (6.25) and (6.28), it is apparent that while each iterate of the consensus algorithms based on edge-variable formulation requires a single message exchange within the neighborhood of each node, the node-variable based algorithms require at least twice the number of message exchanges per round.

In the scaled-relaxed-ADMM method, we first minimize the second largest generalized eigenvalue of $(2AD^{-1}A - D, D)$ using the quasi-convex program (6.24). Let $A^\star$ and $D^\star$ be the adjacency and the degree matrices associated with the optimal solution of (6.24). After extensive simulations, we observed that formulating the fixed point equation (6.12) as

$$\begin{aligned} M_{11} &= \alpha\rho(Q + \rho D^\star)^{-1}(A^\star {D^\star}^{-1} A^\star) + I, \\ M_{12} &= \alpha\rho(Q + \rho D^\star)^{-1}, \quad M_{21} = -\frac{\alpha}{2}(A^\star {D^\star}^{-1} A^\star + D^\star), \end{aligned} \tag{6.30}$$

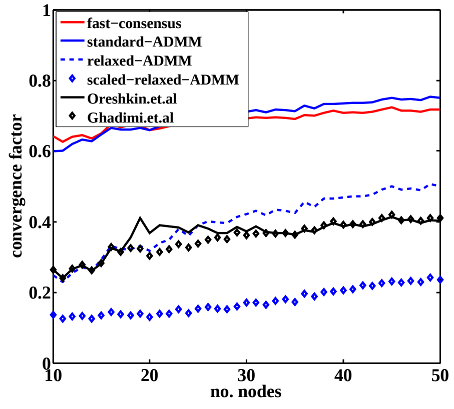
instead of using (6.29), often significantly improves the convergence factor of the ADMM algorithm for the node-variable formulation. Note that this reformulation leads to $\{\lambda_i\}$ (in Theorem 6.4) being the generalized eigenvalues of $(AD^{-1}A,\ D)$. These eigenvalues have several nice properties, e.g., they are positive and satisfy Case I, for which we presented the optimal ADMM parameters $(\alpha^\star, \rho^\star)$ in Theorem 6.4. The algorithm formulated by (6.30) corresponds to running the ADMM algorithm over a network with possible self loops with the adjacency matrix $\tilde{A} = \tilde{A}^\top$ such that

$$A^\star {D^\star}^{-1} A^\star = \frac{1}{2}(\tilde{A} {D^\star}^{-1} \tilde{A} + D^\star),\ \mathrm{diag}(\tilde{A}\mathbf{1}_n) = D^\star.$$

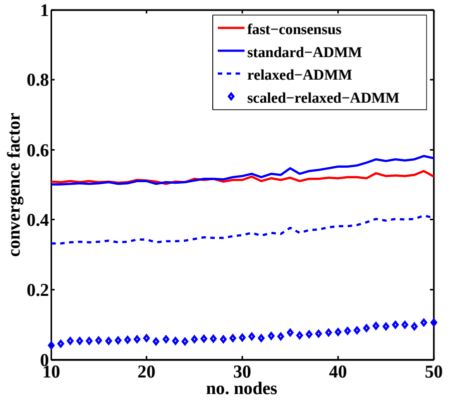
Figures 6.2b, 6.2d and 6.2f illustrate the performance benefits of employing optimal parameter settings developed in this chapter compared to the alternative fast-consensus for different random topologies.

Finally, we compare the performance of different averaging algorithms under fully distributed implementations, where all the algorithm parameters are either computed or estimated in a distributed fashion. For this aim, $|\mathcal{V}| = 200$ nodes were deployed in an RGG topology with initial values $x_i^{(0)} = i/n$. The ADMM (edge-variable) algorithm with local weights given in Proposition 6.3 is compared to the traditional linear averaging algorithm [47], fast-consensus and Oreshkin et al., all with Metropolis-Hastings (MH) weight matrices [47]. While our weight matrix is constructed locally with each node only knowing its own degree, in MH weights each node needs to know its neighbors' degrees as well. We restricted our method and Oreshkin et al. to Case I (Oreshkin et al. also required a similar condition). Following a similar mapping as the node-variable case presented in in (6.30), we replaced the adjacency matrix A in ADMM iterations (6.26) with $(A + D)/2$ that can be computed locally.

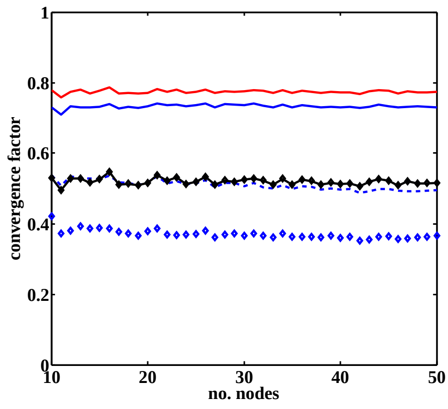
To compute the algorithm parameters for all methods, one has to compute the second largest eigenvalue of the corresponding weight matrices. The parameter can be obtained



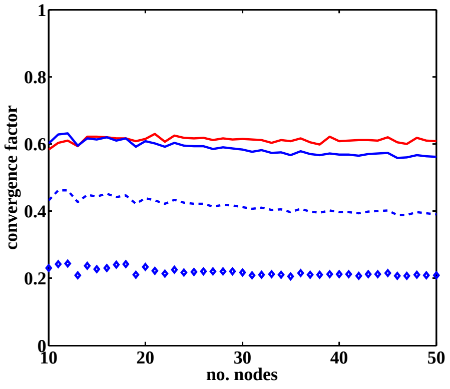
(a) RGG edge variable



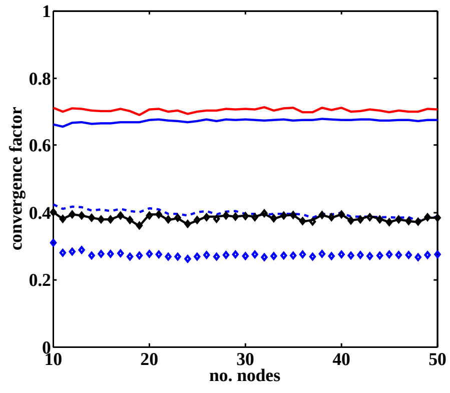
(b) RGG node variable



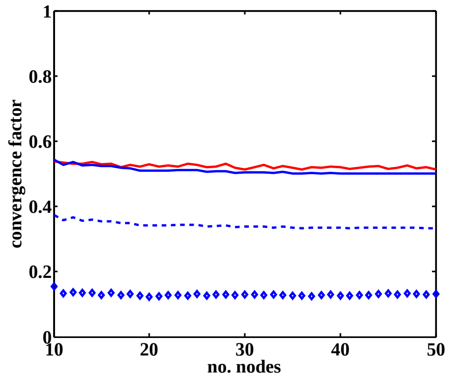
(c) $\epsilon = 0.2$ edge variable



(d) $\epsilon = 0.2$ node variable



(e) $\epsilon = 0.8$ edge variable



(f) $\epsilon = 0.8$ node variable

Figure 6.2: Performance comparison of the proposed optimal scaling for the ADMM algorithm with state-of-the-art algorithms fast-consensus [54], Oreshkin et.al [101] and the multi-step method (Ghadimi et.al). The network of size $n = [10, 50]$ is randomly generated by RGG (random geometric graphs) and Erdős-Rényi graphs with low and high densities $\epsilon = \{0.2, 0.8\}$.

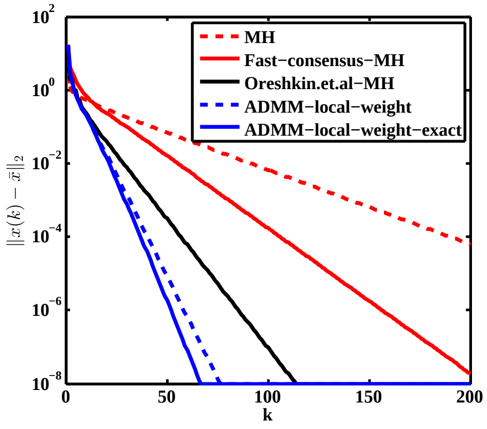


Figure 6.3: MSE versus iteration number for $n = 200$ nodes in RGG topology.

by a distributed power method scheme presented in [101]. In particular, nodes iterate 50 consensus rounds to compute the estimated parameter. We neglected this initialization cost to conduct the experiment.

Figure 6.3 compares the Mean Square Error (MSE) decay rate of different algorithms versus the number of iterations. The ADMM algorithm with the exact knowledge of the second largest eigenvalue is also included as a reference. The figure indicates that our design rules outperform the alternatives.

6.6 Summary

In this chapter, we addressed the optimal parameter selection of the ADMM method for distributed quadratic problems. Two formulations that yield ADMM iterations which can be executed in a distributed manner were considered. For each formulation, we derived the step-size and relaxation parameters that minimize the convergence factor of the iterates. Under mild assumptions on the communication graph, analytical expressions for the optimal parameters were derived and related to the spectral properties of the communication graph. Supposing the optimal constant parameters were chosen, the convergence factor was further minimized by optimizing the edge weights. Several numerical examples validated our theoretical results and confirmed significant performance improvements over the state-of-the-art techniques.

Appendix

6.A Proofs

6.A.1 Proof of Lemma 6.1

Rewrite (6.6) in terms of the variables $\tilde{x} = x + \hat{x}$ and $\tilde{z} = z + \hat{z}$:

$$\begin{array}{ll} \underset{\tilde{x},\tilde{z}}{\text{minimize}} & \frac{1}{2}(\tilde{x}-\hat{x})^\top Q(\tilde{x}-\hat{x}) + q^\top(\tilde{x}-\hat{x}) + c^\top(\tilde{z}-\hat{z}) \\ \text{subject to} & RE(\tilde{x}-\hat{x}) + RF(\tilde{z}-\hat{z}) = 0. \end{array}$$

Collecting the terms in the objective and noting that the feasible solution $(\hat{x}, \hat{z})$ satisfies $RE\hat{x} + RF\hat{z} = Rh$, one can rewrite this problem as

$$\begin{array}{ll} \underset{\tilde{x},\tilde{z}}{\text{minimize}} & \frac{1}{2}\tilde{x}^\top Q\tilde{x} + (q - Q\hat{x})^\top \tilde{x} + c^\top \tilde{z} + k \\ \text{subject to} & RE\tilde{x} + RF\tilde{z} = Rh, \end{array}$$

where k collects the constant terms. Since k does not affect the minimizer, it can be removed and the problem is equivalent to (6.7) when $d = q - Q\hat{x}$.

6.A.2 Proof of Theorem 6.1

First, we show that $s = \dim\big(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\big)$. Consider the quadratic programming problem (6.6) with $\bar{E} = RE$, $\bar{F} = RF$ and $h = 0$. The dimension of $\mathcal{X}$ is given by $\dim(\mathcal{X}) = \dim\big(\mathcal{N}([\bar{E}\,\bar{F}])\big)$. Since $\bar{E} \in \mathbb{R}^{p\times n}$ and $\bar{F} \in \mathbb{R}^{p\times m}$ are of full column rank, we observe that $\dim\big(\mathrm{Im}(\bar{E})\big) = n$ and $\dim\big(\mathrm{Im}(\bar{F})\big) = m$. Using the equalities $\dim\big(\mathrm{Im}([\bar{E}\,\bar{F}])\big) = \dim\big(\mathrm{Im}(\bar{E})\big) + \dim\big(\mathrm{Im}(\bar{F})\big) - \dim\big(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\big)$ and $\dim\big(\mathcal{N}([\bar{E}\,\bar{F}])\big) + \dim\big(\mathrm{Im}([\bar{E}\,\bar{F}])\big) = n + m$, we conclude

$$\dim(\mathcal{X}) = \dim\big(\mathrm{Im}(\bar{E}) \cap \mathrm{Im}(\bar{F})\big) = s.$$

Provided that (6.6) is feasible and under the assumption that there exists a (non-trivial) non-zero tuple $(x, z) \in \mathcal{X}$, we have $s \geq 1$. From Proposition 2.1 it follows that the ADMM iteration (6.6) with feasible solutions $(x, z) \in \mathcal{X}$ converges linearly to a fixed-point in the 1-eigenspace of M with multiplicity $\bar{s} \geq 1$ if and only if $|\varphi_{2n-\bar{s}}| < 1$.

Next, we show that $\bar{s} = s$. **i.** Suppose $\bar{s} > s$, then the 1-eigenspace of M (i.e., the subspace containing the fixed points of (6.6) with dimension $\bar{s}$) cannot be a subset of the feasibility subspace $\mathcal{X}$ with dimension s. Therefore, depending on the initial condition, a fixed-point of (6.6) might be infeasible. Thus, we conclude $\bar{s} \leq s$. **ii.** We show that each base of $\mathcal{X}$ relates to a 1-eigenvector of M. Let $V_{\mathcal{X}} \in \mathbb{R}^{(n+m)\times s}$ be a matrix whose columns are a basis for the feasibility subspace $\mathcal{X}$ and partition this matrix as $V_{\mathcal{X}} = [V_x^\top \, V_z^\top]^\top$. Given the partitioning of $V_{\mathcal{X}}$ we have that $\bar{E}V_x + \bar{F}V_z = 0$ and for all columns v of V_x it holds $\bar{E}v \in \operatorname{Im}(\bar{F})$. Let $u \triangleq [u_1^\top, u_2^\top]^\top$ be a 1-eigenvector of (6.6). From the identity $Mu = u$ it yields

$$\left(M_{11} + \frac{1}{\alpha}M_{12}M_{21} - I\right)u_1 = 0, \quad u_2 = \frac{1}{\alpha}M_{21}u_1.$$

By substituting M_{11}, M_{12}, and M_{21} from (6.6) in former identities we have

$$-\alpha\rho(Q + \rho\bar{E}^\top\bar{E})^{-1}\bar{E}^\top\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}u_1 = 0.$$

From $\bar{E}v \in \operatorname{Im}(\bar{F})$ it follows $\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}v = 0$ and, therefore,

$$u = \begin{bmatrix} I \\ \frac{1}{\alpha}M_{21} \end{bmatrix} v,$$

is a 1-eigenvector of M. This means that each column of V_x corresponds to a 1-eigenvector of M; i.e., $\operatorname{rank}(V_x) \leq \bar{s}$. Next, we derive the rank of V_x. Given that $\bar{F}$ has full column rank, using the equation $\bar{E}V_x + \bar{F}V_z = 0$ we have that $V_z = -\bar{F}^\dagger\bar{E}V_x$. Hence, we conclude that $\operatorname{rank}(V_x) = \operatorname{rank}(V_{\mathcal{X}}) = \dim(\mathcal{X}) = s$. Thus, $s \leq \bar{s}$. Combining **i.** and **ii.**, we have $\bar{s} = s$.

Finally, we show that fixed-points of the ADMM iterations satisfy the optimality conditions of (6.6) in terms of the augmented Lagrangian. The fixed-point of the ADMM iterations (6.8) satisfy the system of equations

$$\begin{bmatrix} Q + \rho\bar{E}^\top\bar{E} & \rho\bar{E}^\top\bar{F} & \rho\bar{E}^\top \\ \alpha\bar{F}^\top\bar{E} & \alpha\bar{F}^\top\bar{F} & \bar{F}^\top \\ \bar{E} & \bar{F} & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ z^\star \\ u^\star \end{bmatrix} = \begin{bmatrix} -q \\ -c/\rho \\ 0 \end{bmatrix}. \tag{6.31}$$

From Karush-Kuhn-Tucker optimality conditions of the augmented Lagrangian

$$L_\rho(x, z, u) = \frac{1}{2}x^\top Qx + q^\top x + c^\top z + \frac{\rho}{2}\|\bar{E}x + \bar{F}z\|^2 + \rho u^\top(\bar{E}x + \bar{F}z),$$

it yields

$$\begin{bmatrix} Q + \rho\bar{E}^\top\bar{E} & \rho\bar{E}^\top\bar{F} & \rho\bar{E}^\top \\ \rho\bar{F}^\top\bar{E} & \rho\bar{F}^\top\bar{F} & \rho\bar{F}^\top \\ \bar{E} & \bar{F} & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ z^\star \\ u^\star \end{bmatrix} = \begin{bmatrix} -q \\ -c \\ 0 \end{bmatrix},$$

which is equivalent to (6.31) by noting that $\bar{F}^\top\bar{E}x^\star + \bar{F}^\top\bar{F}z^\star = \bar{F}^\top(\bar{E}x^\star + \bar{F}z^\star) = 0$.

6.A.3 Proof of Theorem 6.2

To satisfy the eigenvalue equation $M[v_i^\top \; w_i^\top]^\top = \varphi_i [v_i^\top \; w_i^\top]^\top$, v_i and w_i should satisfy

$$\begin{aligned}
&\left(M_{11} + \frac{1}{\varphi_i - 1 + \alpha} M_{12} M_{21} - \varphi_i I\right) v_i = 0,\\
&w_i = \frac{1}{(\varphi_i - 1 + \alpha)} M_{21} v_i.
\end{aligned}$$

When $\bar{E}^\top \bar{E} = \kappa Q$, we have

$$\begin{aligned}
&\left(M_{11} + \frac{1}{\varphi_i - 1 + \alpha} M_{12} M_{21} - \varphi_i I\right) v_i\\
&= \alpha\beta (\bar{E}^\top \bar{E})^{-1} \bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E} v_i + v_i\\
&- \frac{\alpha^2 \beta}{\varphi_i - 1 + \alpha} (\bar{E}^\top \bar{E})^{-1} \bar{E}^\top \Pi_{\mathrm{Im}(\bar{F})} \bar{E} v_i - \varphi_i v_i\\
&= (\alpha\beta\lambda_i + 1) v_i - \frac{\alpha^2\beta}{2} \frac{\lambda_i + 1}{\varphi_i - 1 + \alpha} v_i - \varphi_i v_i\\
&= \left(\alpha\beta\lambda_i + 1 - \varphi_i - \frac{\alpha^2 \beta(\lambda_i + 1)}{2(\varphi_i - 1 + \alpha)}\right) v_i,
\end{aligned}$$

where the last steps follow from the generalized eigenvalue assumption. Thus, the eigenvalues of M are given as the solution of

$$\varphi_i^2 + (\alpha - \alpha\beta\lambda_i - 2)\varphi_i + \alpha\beta\lambda_i(1 - \frac{\alpha}{2}) + \frac{1}{2}\alpha^2\beta + 1 - \alpha = 0.$$

6.A.4 Proof of Lemma 6.2

Recall that a complex number λ_i is a generalized eigenvalue of $\left(\bar{E}^\top (2\Pi_{\mathrm{Im}(\bar{F})} - I)\bar{E},\ \bar{E}^\top \bar{E}\right)$ if there exists a non-zero vector $\nu_i \in \mathbb{C}^n$ such that $\left(\bar{E}^\top (2\Pi_{\mathrm{Im}(\bar{F})} - I)\bar{E} - \lambda_i \bar{E}^\top \bar{E}\right) \nu_i = 0$. Since $\bar{E}$ has full column rank, $\bar{E}^\top \bar{E}$ is invertible and we observe that λ_i is an eigenvalue of the symmetric matrix $(\bar{E}^\top \bar{E})^{-1/2} \bar{E}^\top (2\Pi_{\mathrm{Im}(\bar{F})} - I)\bar{E}(\bar{E}^\top \bar{E})^{-1/2}$. Since the latter is a real symmetric matrix, we conclude that the generalized eigenvalues and eigenvectors are real.

For the second part of the proof, note that the following bounds hold for a generalized eigenvalue λ_i

$$\min_{\nu \in \mathbb{R}^n} \frac{2\nu^\top \bar{E}^\top \Pi_{\mathrm{Im}(\bar{F})} \bar{E} \nu}{\nu^\top \bar{E}^\top \bar{E} \nu} - 1 \leq \lambda_i \leq \max_{\nu \in \mathbb{R}^n} \frac{2\nu^\top \bar{E}^\top \Pi_{\mathrm{Im}(\bar{F})} \bar{E} \nu}{\nu^\top \bar{E}^\top \bar{E} \nu} - 1.$$

Since the projection matrix $\Pi_{\mathrm{Im}(\bar{F})}$ only takes 0 and 1 eigenvalues we have $0 \leq 2\nu^\top \bar{E}^\top \Pi_{\mathrm{Im}(\bar{F})} \bar{E} \nu \leq 2\nu^\top \bar{E}^\top \bar{E} \nu$ which shows that $\lambda_i \in [-1, 1]$.

6.A.5 Proof of Lemma 6.3

Recall $V_{\mathcal{X}} = [V_x^\top \, V_z^\top]^\top \in \mathbb{R}^{(n+m)\times s}$, the basis for the feasibility subspace $\mathcal{X}$, from Section 6.A.2. We first show that the generalized eigenvectors associated with the unit generalized eigenvalues $\lambda_i = 1$ are in V_x.

Given the partitioning of $V_{\mathcal{X}}$ we have that $\bar{E}V_x + \bar{F}V_z = 0$ and $\bar{E}\nu \in \operatorname{Im}(\bar{F})$ for $\nu \in V_x$. Hence, we have $\Pi_{\operatorname{Im}(\bar{F})}\bar{E}\nu = \bar{E}\nu$, yielding $(\nu^\top(\bar{E}^\top(2\Pi_{\operatorname{Im}(\bar{F})} - I)\bar{E})\nu)/(\nu^\top(\bar{E}^\top\bar{E})\nu) = 1$. Moreover, as 1 is the upper bound for λ_i according to Lemma 6.2, we conclude that $\lambda_n = 1$ is a generalized eigenvalue associated with the eigenvector ν. Recalling from Section 6.A.2 that $\operatorname{rank}(V_x) = s$, we conclude that there exist s generalized eigenvalues equal to 1.

6.A.6 Proof of Lemma 6.4

Recall from Lemma 6.3 that for a feasible problem of the form (6.6) we have $\lambda_i = 1$ for $i \geq n - s + 1$. From (6.14) it follows that each $\lambda_i = 1$ results in two eigenvalues $\varphi = 1$ and $\varphi = 1 - \alpha(1-\beta)$. Thus we conclude that M has at least s eigenvalues equal to 1. Moreover, since $\beta \in (0,1)$ and $\alpha \in (0,2]$, we observe that $|1 - \alpha(1-\beta)| < 1$. Next we consider $i < n - s + 1$ and show that the resulting eigenvalues of M are inside the unit circle for all $\beta \in (0,1)$ and $\alpha \in (0,2]$ using the necessary and sufficient conditions from Proposition 6.1.

The first condition of Proposition 6.1 can be rewritten as $a_0 + a_1 + a_2 = 1/2\alpha^2\beta(1 - \lambda_i) > 0$, which holds for $\lambda_i \in [-1,1)$. Having $\alpha > 0$ and $\lambda_i < 1$, the condition $a_2 > a_0$ can be rewritten as $\alpha < (2(1-\beta\lambda_i))\,/\,(\beta(1-\lambda_i))$. For $\beta < 1$, that the right hand side term is greater than 2, from which we conclude that the second condition is satisfied. It remains to show $a_0 - a_1 + 1 > 0$. Replacing the terms on the left-hand-side, they form a convex quadratic polynomial on α, i.e.,

$$D(\alpha) = \frac{1}{2}\alpha^2\beta(1-\lambda_i) + 2\alpha(\beta\lambda_i - 1) + 4.$$

The minimizer of $D(\alpha)$ is at $\alpha = (2(1-\beta\lambda_i))\,/\,(\beta(1-\lambda_i))$, which was shown to be greater than 2 when addressing the second condition. Since $D(2) = 2\beta(1+\lambda) > 0$, we conclude $D(\alpha) > 0$ for all $\alpha \leq 2$ and the third condition holds.

6.A.7 Proof of Theorem 6.3

The magnitude of φ_{2n-s} can be characterized with Jury's stability test as follows. Consider the non-unit generalized eigenvalues $\{\lambda_i\}_{i\leq n-s}$ and let $\varphi_i = r\tilde{\varphi}_i$ for $r \geq 0$. Substituting φ_i in the eigenvalue polynomials (6.14) yields $r^2\tilde{\varphi}_i{}^2 + ra_1(\lambda_i)\tilde{\varphi}_i + a_0(\lambda_i) = 0$. Therefore, having the roots of these polynomials inside the unit circle is equivalent to having $|\varphi_i| < r$. From the stability of ADMM iterates (see Lemma 6.4) it follows that it is always possible to find $r < 1$. Using the necessary and sufficient conditions from Proposition 6.1, $|\varphi_{2n-s}|$ is obtained as

$$\begin{array}{lll} \underset{r\geq 0}{\text{minimize}} & r & \\ \text{subject to} & a_0(\lambda_i)+ra_1(\lambda_i)+r^2\geq 0 & \\ & r^2\geq a_0(\lambda_i) & \forall\, i\leq n-s \\ & a_0(\lambda_i)-ra_1(\lambda_i)+r^2\geq 0 & \\ & r\geq |1-\alpha(1-\beta)|. & \end{array} \tag{6.32}$$

Next, we remove redundant constraints from (6.32). Considering the first constraint, we aim at finding $\lambda\in\{\lambda_i\}_{i\leq n-s}$ such that $a_0(\lambda_i)+ra_1(\lambda_i)+r^2\geq a_0(\lambda)+ra_1(\lambda)+r^2$ for all $i\leq n-s$. Observing that the former inequality can be rewritten as $\alpha\beta(\lambda-\lambda_i)(1-\frac{\alpha}{2}-r)\leq 0$, we conclude that $\lambda=\lambda_{n-s}$ if $1-\frac{\alpha}{2}\leq r$ and $\lambda=\lambda_1$ otherwise. Hence the constraints $a_0(\lambda_i)+ra_1(\lambda_i)+r^2\geq 0$ for $1<i<n-s$ are redundant. As for the second condition, note that $a_0(\lambda_{n-s})-a_0(\lambda_i)=\alpha\beta(\lambda_{n-s}-\lambda_i)(1-\frac{\alpha}{2})\geq 0$ for all $i\leq n-s$, since $\alpha\in(0,\,2]$. Consequently, the constraints $r^2\geq a_0(\lambda_i)$ for $i<n-s$ can be removed. Regarding the third constraint, we aim at finding $\lambda\in\{\lambda_i\}_{i\leq n-s}$ such that $a_0(\lambda_i)-ra_1(\lambda_i)+r^2\geq a_0(\lambda)-ra_1(\lambda)+r^2$ for all $i\leq n-s$. Since the previous inequality can be rewritten as $\alpha\beta(\lambda-\lambda_i)(1-\frac{\alpha}{2}+r)\leq 0$, which holds for $\lambda=\lambda_1$, we conclude that the constraints for $1<i\leq n-s$ are redundant. Removing the redundant constraints, the optimization problem (6.32) can be rewritten as

$$\begin{array}{ll} \underset{r\geq 0,\{s_i\}}{\text{minimize}} & r \\ \text{subject to} & a_0(\lambda_{n-s})+ra_1(\lambda_{n-s})+r^2-s_1=0 \\ & a_0(\lambda_1)+ra_1(\lambda_1)+r^2-s_2=0 \\ & r^2-a_0(\lambda_{n-s})-s_3=0 \\ & a_0(\lambda_1)-ra_1(\lambda_1)+r^2-s_4=0 \\ & r-|1-\alpha(1-\beta)|-s_5=0 \\ & s_i\geq 0\quad \forall i\leq 5, \end{array} \tag{6.33}$$

where $\{s_i\}$ are slack variables. Subtracting the fourth equation from the second we obtain the following equivalent problem

$$\begin{array}{ll} \underset{\{s_i\}}{\text{minimize}} & \max_i\{r_i\} \\ \text{subject to} & s_i\geq 0,\quad \forall i\leq 5 \\ & r_i\geq 0,\quad \forall i\leq 7 \\ & a_0(\lambda_{n-s})+s_3\geq 0 \\ & \beta^2\lambda_{n-s}^2-2\beta+1+s_1\geq 0 \\ & \beta^2\lambda_1^2-2\beta+1+s_4\geq 0, \end{array} \tag{6.34}$$

where

$$\begin{aligned}
r_1 &= 1 - \frac{\alpha}{2} + \frac{\alpha}{2}\beta\lambda_{n-s} + \frac{\alpha}{2}\sqrt{\beta^2\lambda_{n-s}^2 - 2\beta + 1 + s_1} \\
r_2 &= \frac{s_2 - s_4}{2a_1(\lambda_1)} \\
r_3 &= \sqrt{a_0(\lambda_{n-s}) + s_3} \\
r_4 &= -1 + \frac{\alpha}{2} - \frac{\alpha}{2}\beta\lambda_1 + \frac{\alpha}{2}\sqrt{\beta^2\lambda_1^2 - 2\beta + 1 + s_4} \\
r_5 &= |1 - \alpha(1-\beta)| + s_5 \\
r_6 &= 1 - \frac{\alpha}{2} + \frac{\alpha}{2}\beta\lambda_{n-s} - \frac{\alpha}{2}\sqrt{\beta^2\lambda_{n-s}^2 - 2\beta + 1 + s_1} \\
r_7 &= -1 + \frac{\alpha}{2} - \frac{\alpha}{2}\beta\lambda_1 - \frac{\alpha}{2}\sqrt{\beta^2\lambda_1^2 - 2\beta + 1 + s_4}.
\end{aligned}$$

In the above equation, $\{r_1, r_6\}$, r_2, r_3, $\{r_4, r_7\}$, and r_5 are solutions to the first, second, third, forth and fifth equality constraints in (6.33), respectively. The last three inequalities impose that r_1, r_3, r_4, r_6, and r_7 are real values. Moreover, the last two constraints ensure that the inequalities $r_1 \geq r_6$ and $r_4 \geq r_7$ hold. Performing the minimization of each r_i with respect to the corresponding slack variable s_i we obtain $|\varphi_{2n-s}| = \max\{r_1^\star, r_3^\star, r_4^\star, r_5^\star\}$ where $r_i^\star$ are computed as in (6.34) with

$$\begin{aligned}
s_1^\star &= \max\{0, \ -(\beta^2\lambda_{n-s}^2 - 2\beta + 1)\}, \\
s_2^\star &= s_4^\star = \max\{0, \ -(\beta^2\lambda_1^2 - 2\beta + 1)\}, \\
s_3^\star &= \max\{0, \ -a_0(\lambda_{n-s})\}, \quad s_5^\star = 0.
\end{aligned}$$

The proof concludes by noting that the optimum solutions to the optimization problem (6.32) are attained at the boundary of its feasible set. Therefore, having a zero slack variable, i.e., $s_i^\star = 0$, is a necessary condition for $|\varphi_{2n-s}| = r_i^\star$.

6.A.8 Proof of Proposition 6.2

Recalling that $|\varphi_{2n-s}|$ is characterized by (6.16), the proof follows from showing that the inequalities

i) $g_r^-(1, \beta, \lambda) < g_r^+(1, \beta, \lambda)$

ii) $g_r^+(1, \beta, \lambda) < \max\{g_r^+(\alpha, \beta, \lambda), \ g_r^-(\alpha, \beta, \lambda)\}$

iii) $g_1(1, \beta) < g_1(\alpha, \beta)$

iv) $g_c(1, \beta, \lambda) < g_c(\alpha, \beta, \lambda)$

hold for $\alpha < 1$, $\beta \in (0, \ 1)$, and $\lambda \in \{\lambda_i\}_{i \leq n-s}$.

The first inequality i) can be rewritten as $-\beta\lambda < 1$, which holds since $\lambda \geq -1$. As for the second inequality ii), it suffices to show $\Delta g^+ \triangleq g_r^+(1, \beta, \lambda) - g_r^+(\alpha, \beta, \lambda) < 0$. Forming

Δg^+, after simplifications, it yields

$$\Delta g^+ = \frac{1-\alpha}{2}\left(\sqrt{\lambda^2\beta^2 - 2\beta + 1 + s_r^+} + \lambda\beta - 1\right)$$

and observe that $\Delta g^+ < 0$ holds if $\sqrt{\lambda^2\beta^2 - 2\beta + 1 + s_r^+} + \lambda\beta - 1 < 0$. The latter inequality holds for $\lambda \in [-1, 1)$, hence we conclude that $g_r^+(1,\beta,\lambda) < g_r^+(\alpha,\beta,\lambda) \leq \max\{g_r^+(\alpha,\beta,\lambda),\ g_r^-(\alpha,\beta,\lambda)\}$.

Next we consider the third inequality iii). For $\alpha \leq 1$ we have $g_1(\alpha,\beta) = 1 - \alpha(1-\beta)$. It directly follows that $g_1(1,\beta) < g_1(\alpha,\beta)$ for $\alpha < 1$, since $g_1(1,\beta) - g_1(\alpha,\beta) = \alpha - 1$.

In the last step of the proof we address iv). In particular, since $g_c(\alpha,\beta,\lambda)$ is positive, having $\Delta g_c \triangleq g_c(1,\beta,\lambda)^2 - g_c(\alpha,\beta,\lambda)^2 < 0$ is equivalent to iv). Thus we study the sign of $\Delta g_c = (1-\alpha)\left(\frac{1}{2}\beta\alpha(1-\lambda) + \frac{1}{2}\lambda\beta + \frac{1}{2}\beta - 1\right) + s_c(1) - s_c(\alpha)$. Using the equality $\frac{1}{2}\beta(1-\lambda) + \frac{1}{2}\lambda\beta + \frac{1}{2}\beta - 1 = \beta - 1$ and $1 - \lambda > 0$, for $\alpha < 1$ we have

$$\Delta g_c < (1-\alpha)\,(\beta - 1) + s_c(1) - s_c(\alpha).$$

Recall from Theorem 6.3 that we can only have $|\varphi_{2n-s}(\alpha,\beta,\lambda)| = g_c(\alpha,\beta,\lambda)$ when $s_c(\alpha) = 0$. Note that the case when $s_c(\alpha) > 0$ corresponds to

$$|\varphi_{2n-s}(\alpha,\beta,\lambda)| = \max\{g_r^+(\alpha,\beta,\lambda),\ g_r^-(\alpha,\beta,\lambda),\ g_1(\alpha,\beta)\},$$

which is covered in the previous part of the proof. In the following we let $s_c(\alpha) = 0$ and derive the upper bound $s_c(1) < -(1-\alpha)(\beta-1)$. Given the definition of $s_c(1) = \max\{0, -(1/2\beta(1-\lambda) + \beta\lambda)\}$ in Theorem 6.3, the latter upper bound holds if the following inequalities are satisfied: $(1-\alpha)(\beta-1) < 0$ and $\Delta s_c \triangleq (1-\alpha)(\beta-1) - (\beta(1-\lambda)/2 + \beta\lambda) < 0$. The proof concludes by observing that, for $\alpha < 1$, $\beta \in (0,1)$, and $\lambda \in [-1,\ 1)$, the former inequality holds, which in turn satisfies the latter inequality, since $\Delta s_c < -(\beta(1-\lambda)/2 + \beta\lambda) = -1/2\beta(1+\lambda) < 0$.

6.A.9 Proof of Theorem 6.4

Some preliminary results are derived before proving the theorem.

Lemma 6.9

For a fixed $\alpha \in [1,\ 2]$, $\lambda \in \{\lambda_i\}_{i \leq n-s}$, and $s_c = 0$, the function $g_c(\alpha,\beta,\lambda)$, defined in (6.17), is monotonically increasing with $\beta \in (0,\ 1)$.

Proof. The derivative of $g_c(\alpha,\beta,\lambda)$ with respect to β is

$$\nabla_\beta g_c = \frac{1}{2}(\frac{1}{2}\alpha^2\beta(1-\lambda) + 1 - \alpha + \alpha\beta\lambda)^{-1/2}(\frac{1}{2}\alpha(1-\lambda) + \lambda),$$

which is nonnegative if and only if $\frac{1}{2}\alpha(1-\lambda) + \lambda \geq 0$. The inequality can be rewritten as $\alpha \geq \dfrac{-2\lambda}{1-\lambda}$, which holds for all $\alpha \in [1,\ 2]$ and $\lambda \geq -1$. $\square$

Lemma 6.10

For a fixed $\alpha \in (0, 2]$, $\lambda \in \{\lambda_i\}_{i \leq n-s}$, and $s_r^+ = s_r^- = 0$, the functions $g_r^+(\alpha, \beta, \lambda)$ and $g_r^-(\alpha, \beta, \lambda)$ are monotonically decreasing with respect to β.

Proof. Considering first $g_r^+(\alpha, \beta, \lambda)$, its derivative with respect to β is

$$\nabla_\beta g_r^+(\alpha, \beta, \lambda) = \frac{\alpha}{2}\left((\lambda^2\beta - 1)(\lambda^2\beta^2 - 2\beta + 1)^{-1/2} + \lambda\right).$$

Since $\beta \in (0,\ 1)$ and recalling from Lemma 6.2 that $|\lambda| \leq 1$, we have $\lambda^2\beta - 1 < 0$ and thus

$$\begin{aligned}\nabla_\beta g_r^+(\alpha, \beta, \lambda) &\leq \frac{\alpha}{2}\left((\lambda^2\beta - 1)(\beta^2 - 2\beta + 1)^{-1/2} + \lambda\right)\\ &= \frac{\alpha(\lambda - 1)(1 + \lambda\beta)}{2(1 - \beta)} < 0.\end{aligned}$$

Considering $g_r^-(\alpha, \beta, \lambda)$, we have $\nabla_\beta g_r^-(\alpha, \beta, \lambda) = \frac{\alpha}{2}\left((\lambda^2\beta - 1)(\lambda^2\beta^2 - 2\beta + 1)^{-1/2} - \lambda\right)$. Similarly as before, $\nabla_\beta g_r^-(\alpha, \beta, \lambda)$ can be upper-bounded by

$$\begin{aligned}\nabla_\beta g_r^-(\alpha, \beta, \lambda) &\leq \frac{\alpha}{2}\left((\lambda^2\beta - 1)(\beta^2 - 2\beta + 1)^{-1/2} - \lambda\right)\\ &= \frac{\alpha(\lambda + 1)(\lambda\beta - 1)}{2(1 - \beta)} \leq 0.\end{aligned}$$

□

[Proof of Theorem 6.4] Recall from Proposition 6.2 that the minimizing relaxation parameter $\alpha^\star$ lies in the interval $[1,\ 2]$.

First, suppose that $s_r^+ = s_c = 0$ and observe that $g_r^+ \geq g_r^-$ is equivalent to

$$\alpha \leq \frac{4}{\eta} \tag{6.35}$$

with

$$\eta = 2 - (\lambda_{n-s} + \lambda_1)\beta + \sqrt{\lambda_1^2\beta^2 - 2\beta + 1} - \sqrt{\lambda_{n-s}^2\beta^2 - 2\beta + 1}.$$

Recall that $g_1 = \max\{1 - \alpha(1 - \beta),\ -1 + \alpha(1 - \beta)\}$. For $\beta \leq 1/2$, $g_r^- \geq -1 + \alpha(1 - \beta)$. Therefore, given (6.35):

$$|\varphi_{2n-s}| = \max\{g_r^+, g_c, 1 - \alpha(1 - \beta)\} \tag{6.36}$$

Note that g_r^+ is decreasing with respect to β while g_c and $1 - \alpha(1 - \beta)$ are increasing with respect to β. Hence, $\beta^\star$ satisfies $g_r^+ = \max\{g_c, 1 - \alpha(1 - \beta)\}$. Next, we consider the following cases.

Case I and Case II: Suppose that $\lambda_{n-s} \neq 0$ and $\beta^\star$ is the solution to $g_r^+ = g_c$, yielding $\beta^\star = (1 - \sqrt{1 - \lambda_{n-s}^2})/\lambda_{n-s}^2$. We show $\beta^\star$ is the minimizer by deriving $g_r^+(\alpha, \beta^\star, \lambda_{n-s}) \geq 1 - \alpha(1 - \beta^\star)$. Since $\beta^{\star 2}\lambda_{n-s}^2 - 2\beta^\star + 1 = 0$, the latter inequality can be rewritten as $\beta^\star \leq 1/(2 - \lambda_{n-s})$, which is equivalent to $1 - \lambda_{n-s}^2 \geq \left(1 - \frac{\lambda_{n-s}^2}{2-\lambda_{n-s}}\right)^2$. After manipulations, the former condition reduces to $\lambda_{n-s}(1 - \lambda_{n-s}) \geq 0$, which holds for $1 > \lambda_{n-s} \geq 0$. Hence the parameter $\beta^\star$ occurs for $g_r^+ = g_c$. Moreover, note that $\beta^{\star 2}\lambda_{n-s}^2 - 2\beta^\star + 1 = 0$, which ensures $s_r^+ = s_c = 0$.

We now fix $\beta^\star$ and optimize over the relaxation parameter. Observing that g_r^+ is decreasing with α, since $\nabla_\alpha g_r^+(\alpha, \beta^\star, \lambda_{n-s}) = 1/2(-1 + \beta^\star \lambda_{n-s}) < 0$, we conclude that $\alpha^\star$ is the upper bound in (6.35).

For the case where $\lambda_{n-s} \geq |\lambda_1|$ and $\lambda_{n-s} \neq 0$ (Case I), since $\lambda_{n-s} + \lambda_1 \geq 0$ for any choice of β, $\sqrt{\lambda_1^2\beta^2 - 2\beta + 1} - \sqrt{\lambda_{n-s}^2\beta^2 - 2\beta + 1} \leq 0$, the upperbound of (6.35) will be larger than 2. On the other hand, $\alpha \in (0, 2]$. Thus, $\alpha^\star = 2$.

For the case where $|\lambda_1| \geq \lambda_{n-s} > 0$ (Case II), following a similar line of reasoning to the previous case, we obtain

$$\alpha^\star = \frac{4}{2 - (\lambda_{n-s} + \lambda_1)\beta^\star + \sqrt{\lambda_1^2\beta^{\star 2} - 2\beta^\star + 1}} \leq 2.$$

Case III: As before $|\varphi_{2n-s}| = \max\{g_r^+, g_c, 1 - \alpha(1 - \beta)\}$. Given Lemmas 6.9 and 6.10, the minimizer $\beta^\star$ occurs for $g_r^+ = \max\{g_c, 1 - \alpha(1 - \beta)\}$. Supposing that the minimizer occurs for $g_r^+(\alpha, \beta^\star, \lambda_{n-s}) = 1 - \alpha(1 - \beta^\star)$, we obtain $\beta^\star = 1/2$ and $g_r^+(\alpha, \beta^\star, \lambda_{n-s}) = 1 - \alpha/2$. Next we show that $1 - \alpha/2 \geq g_c(\alpha, \beta^\star, \lambda_{n-s})$, which can be rewritten as $0 \geq \frac{\alpha}{2}\lambda_{n-s}(1 - \frac{\alpha}{2}) + s_c$. Recalling from Theorem 6.3 that $s_c = 0$ is a necessary condition for $|\varphi_{n-s}| = g_c$, we set s_c to zero. Since $\lambda_{n-s} \leq 0$, we have that $g_r^+(\alpha, \beta^\star, \lambda_{n-s}) \geq g_c(\alpha, \beta^\star, \lambda_{n-s})$ for $s_c = 0$.

Next we fix $\beta^\star = 1/2$ and optimize over $\alpha \in [1, 2]$. Note that $g_r^+(\alpha, \beta^\star, \lambda_{n-s}) = 1 - \alpha/2$ is decreasing with α and recall that α is constrained to satisfy (6.35). Hence the best parameter $\alpha^\star$ occurs at the boundary of (6.35), i.e., $\alpha^\star = 4/(2 - \lambda_1)$, thus concluding the proof.

6.A.10 Proof of Corollary 6.1

From the proof of Theorem 6.4, when $\lambda_{n-s} > 0$, then $\beta^\star = (1 - \sqrt{1 - \lambda_{n-s}^2})/\lambda_{n-s}^2$ is optimal, and when $\lambda_{n-s} \leq 0$, $\beta^\star = 1/2$ is the minimizer. The result follows by setting $\alpha = 1$ and obtaining corresponding convergence factors that are given by $g_r^+(\alpha = 1, \beta^\star, \lambda_{n-s})$.

6.A.11 Proof of Lemma 6.5

Without loss of generality, consider the optimization problem (6.6) with $\bar{h} = 0$ (see Lemma 6.1) and include the additional constraint $\bar{F}^\top(\bar{E}x + \bar{F})z = 0$. This constraint may be rewritten as $z = -\bar{F}^\dagger \bar{E}x$. Replacing the latter expression in the constraint $\bar{E}x+\bar{F}z = 0$ we obtain $\Pi_{\mathcal{N}(F^\top)}\bar{E}x = 0$, which can be rewritten as $x = Py$ for some $y \in \mathbb{R}^s$. Hence the optimization problem (6.6) is equivalent to

$$\underset{y\in\mathbb{R}^s}{\text{minimize}} \quad \frac{1}{2}y^\top P^\top QPy + q^\top Py - c^\top \bar{F}^\dagger \bar{E}Py$$

The proof follows directly by noting that $P^\top E^\top WEP = P^\top QP$ yields the same optimal solution of the equivalent problem when $Q \succeq 0$ is replaced with $E^\top WE \succ 0$.

6.A.12 Proof of Lemma 6.6

Recall that Assumption 6.1 states that R is chosen so that all solutions to $R(Ex + Fz) = 0$ satisfy $Ex + Fz = 0$. Decomposing Ex as $Ex = \Pi_{\mathcal{N}(F^\top)}Ex + \Pi_{\text{Im}(F)}Ex$, the first equation becomes $R\left(\Pi_{\mathcal{N}(F^\top)}Ex + \Pi_{\text{Im}(F)}(Ex + Fz)\right) = 0$. Since $\mathcal{N}(F^\top)$ and $\text{Im}(F)$ are orthogonal complements, the latter equation can be rewritten as

$$0 = RF\left((F^\top F)^{-1}F^\top Ex + z\right) \tag{6.37a}$$
$$0 = R\Pi_{\mathcal{N}(F^\top)}Ex. \tag{6.37b}$$

The equation (6.37a) admits the same solutions as its unscaled counterpart with $R = I$ if and only if RF has an empty null-space, which is equivalent to have $F^\top WF \succ 0$.

As for equation (6.37b), assuming the latter inequality holds and decomposing $REx = \bar{E}x$ as $\bar{E}x = \Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}x + \Pi_{\text{Im}(\bar{F})}\bar{E}x$, the scaled equations (6.37) can be rewritten as

$$0 = \Pi_{\text{Im}(\bar{F})}\bar{E}x + \bar{F}z \tag{6.38a}$$
$$0 = \Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}x. \tag{6.38b}$$

Solutions to (6.37b) with $R = I$ can be parameterized as $x = Pw \in \mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$, where $P \in \mathbb{R}^{n\times s}$ is an orthonormal basis for $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E)$. Moreover, note that $x = Pw$ is also a solution to (6.38b), yielding $\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}Pw = 0$. Decomposing x as $x = Pw+P_1y$, (6.38b) becomes $\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}(Pw + P_1y) = \Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}P_1y = 0$. Thus, (6.37b) and (6.38b) admit the same solutions $x = Pw$ if and only if $P_1^\top \bar{E}^\top \Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E}P_1 \succ 0$.

The proof concludes by observing that, under Assumption 6.1, (6.37b) with $R = I$ and (6.38b) admit the same solutions.

6.A.13 Proof of Lemma 6.7

First, suppose that $W = R^\top R$ is chosen such that Assumption 6.1 holds, as per Lemma 6.6. Therefore, we have $\mathcal{N}(\Pi_{\mathcal{N}(F^\top)}E) = \mathcal{N}(\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E})$. Note that the unit generalized eigenspace of $(\bar{E}^\top\left(2\Pi_{\text{Im}(\bar{F})} - I\right)\bar{E}, \bar{E}^\top\bar{E})$ is characterized by the solutions of the equation $(\bar{E}^\top\left(2\Pi_{\text{Im}(\bar{F})} - I\right)\bar{E} - \bar{E}^\top\bar{E})v = 0$ and corresponds to $\mathcal{N}(\Pi_{\mathcal{N}(\bar{F}^\top)}\bar{E})$.

Hence, we have $P_1^\top \left(\bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E}\right) P_1 \prec \lambda_{n-s} P_1^\top (\bar{E}^\top \bar{E}) P_1$ and conclude that $\lambda > \lambda_{n-s}$ holds if and only if

$$P_1^\top \left(\bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E} - \lambda \bar{E}^\top \bar{E}\right) P_1 \prec 0. \tag{6.39}$$

Using the Schur lemma, (6.39) can be rewritten as (6.23).

To conclude the proof, we show that a feasible W with $\lambda \leq 1$ does indeed satisfy the conditions in Lemma 6.6. Suppose that (6.39) holds with some $\lambda \leq 1$. The inequality $F^\top W F \succ 0$ is clearly satisfied. As for the condition $P_1^\top \bar{E}^\top \Pi_{\mathcal{N}(\bar{F})} \bar{E} P_1 \succ 0$, note that W satisfies

$$P_1^\top \left(\bar{E}^\top \left(2\Pi_{\mathrm{Im}(\bar{F})} - I\right) \bar{E} - \lambda \bar{E}^\top \bar{E} - (1-\lambda)\bar{E}^\top \bar{E}\right) P_1 \prec 0,$$

since $(1-\lambda)\bar{E}^\top \bar{E} \succeq 0$ for $\lambda \leq 1$. Observing that the latter condition can be rewritten as $2P_1^\top \bar{E}^\top (I - \Pi_{\mathrm{Im}(F)}) \bar{E} P_1 = 2P_1^\top \bar{E}^\top \Pi_{\mathcal{N}(\bar{F})} \bar{E} P_1 \succ 0$ concludes the proof.

6.A.14 Proof of Theorem 6.5

The proof follows from Lemmas 6.5, 6.6, and 6.7.

6.A.15 Proof of Proposition 6.3

From the x_i-update in the ADMM iterations (6.25), we see that the diagonal block of $E^\top W E$ corresponding to node i is given by $\sum_{j \in \mathcal{N}_i} W_{(i,j)}$. Hence, $E^\top W E = \kappa Q$ is met if each agent i ensures that $\sum_{j \in \mathcal{N}_i} W_{(i,j)} = \kappa Q_i$.

6.A.16 Proof of Proposition 6.4

The proof is a direct consequence of the analytical expressions of the eigenvalues of the normalized Laplacian of given in [60, 129] and the relationship $1 - \lambda = \psi$.

6.A.17 Proof of Lemma 6.8

The second last constraint ensures that $\lambda > \lambda_{n-1}$ and follows from a special case of Lemma 6.7, while the last constraint enforces $\lambda_1 > -\lambda$.

Chapter 7
Conclusions and future work

IN this thesis, we have analyzed and optimized the performance of large-scale optimization algorithms. In particular, we addressed a number of open problems in the global convergence of first-order methods. Moreover, we presented novel primal and dual techniques to accelerate the basic gradient iterates via multi-step iterates in the context of network optimization. In the second line of work, we considered the ADMM method and optimized its performance for quadratic problems. We demonstrated how different techniques such as relaxation and constraint scaling can be properly carried out to meliorate the performance of the ADMM algorithm.

In this chapter, we discuss the outcome of each chapter of the thesis. In each case, we also sketch the possible directions to be taken in order to extend the results of the current thesis.

7.1 First-order methods

In Chapter 3, we considered the global convergence of the Heavy-ball method for convex optimization. Previous studies have shown that, when applied to unconstrained optimization problems with strongly convex twice-differentiable cost functions, the method locally converges to the optimum point and demonstrated the advantages of the method compared to alternative first-order techniques [52]. Motivated by these results, we derived the global convergence rate of the Heavy-ball methods for two classes of unconstrained optimization problems.

When the objectives are convex and continuously differentiable with Lipschitz gradient ($f \in \mathcal{F}_L^{1,1}$), we observed that the Césaro-mean of the Heavy-ball iterates converge to the optimum at rate $\mathcal{O}(1/k)$. A similar convergence rate was also proved for the Nesterov's method with constant step-sizes. Our result indicates that if $f \in \mathcal{F}_L^{1,1}$, the gradient, Heavy-ball, and Nesterov's method with constant step-sizes converge at the same rate.

When the objective functions are also strongly convex but not necessarily twice continuously differentiable ($f \in \mathcal{S}_{\mu,L}^{1,1}$), we established linear global convergence rate for the Heavy-ball method along with global stability conditions. The newly derived parameter bounds are wider than the ones for the class $\mathcal{F}_L^{1,1}$ and wider than the previously known bounds for non-convex cost functions with Lipschitz continuous gradients.

7.1.1 Future work

One natural direction to proceed is to extend the results of this chapter by applying the Heavy-ball iterations to the constrained convex optimization problems. Several applications in distributed optimization, multi-agent systems, control, and machine learning motivate such a study.

Moreover, for the class $\mathcal{S}_{\mu,L}^{1,1}$, numerical experiments showed that the Heavy-ball method can be tuned to outperform the alternatives. Deriving sharper convergence bounds for the Heavy-ball method is another interesting future direction.

Finally, for the class $\mathcal{F}_L^{1,1}$, we designed a Heavy-ball based algorithm with time-varying step-sizes in which the individual iterates $x^{(k)}$ converge at rate $\mathcal{O}(1/k)$. Our theoretical and numerical results (see Fig. 3.1) suggest that the Heavy-ball method (at least with currently developed step-sizes) cannot perform at a better convergence rate. This is more noticeable when we consider that Nesterov's iterations with similar time-varying step-sizes converge at rate $\mathcal{O}(1/k^2)$. Proving or disproving a $\mathcal{O}(1/k^2)$ convergence rate for Heavy-ball iterates remains as an open question.

7.2 Multi-step methods for network optimization

In Chapter 4, we studied the multi-step gradient methods for network optimization. Motivated by the Heavy-ball iterates from the literature of unconstrained minimization, we designed primal and dual decomposition based methods to improve the performance of gradient-based algorithms in network optimization. In these applications, a group of decision makers collaborated among each other through an undirected graph to solve joint optimization problems such as resource allocation, global agreement, or network utility maximization problems. By a primal decomposition approach, we studied a weighted gradient method and derived the multi-step counterpart along with its stability conditions, optimal step-size parameters and the associated convergence factor.

The weighting matrices had to obey the sparsity pattern imposed by the underlying graph. In addition, they had to fulfill some extra requirements to maintain the per-iterate feasibility of the primal decision parameters. We also looked into the problem of optimal weight selection for the multi-step methods and proposed convex optimization problems to find such weights. We showed how the technique is applicable for the resource allocation problem and demonstrated the performance benefits of the multi-step method compared to the alternatives.

In general, however, finding the feasible weighting matrices for network optimization problems may not be always possible. Therefore, we proposed a dual-decomposition based multi-step method and derived its optimal step-sizes along with the optimal convergence factor. The applications of this technique was demonstrated numerically in distributed averaging and Internet congestion control applications.

The algorithm parameters for both gradient and multi-step methods are highly dependent on the global information concerning the smoothness of the cost functions and the spectral properties of the communication graphs. Such information may not be locally available to the decision makers and usually are estimated in practice. For this reason, we performed a robustness study to evaluate the sensitivity of the multi-step method to the estimation errors. It turned out that misestimation in algorithm parameters has exactly the same effect on the gradient and the multi-step methods in terms of the stability. Moreover, in most of the cases the performance benefits of the multi-step method compared to the gradient method prevails in the presence of parameter uncertainty.

7.2.1 Future work

The results of Chapter 4 hold under the assumption of having perfect synchronized communication between the agents. In real-world applications, however, decision makers often perform asynchronous updates. The wireless links between the decision makers are most likely asymmetric which indicates that the directed graphs should be considered in order to model the communication mechanism. The effects of packet losses, communication delays, and quantization errors, on the other hand, are unknown on the performance of the multi-step methods. All of these challenges are interesting directions for future investigations.

The second type of interesting extensions are to consider other types of dependencies between the decision makers such as linear inequality constraints. These constraints are usually handled by projected gradient methods. Similar projected multi-step counterparts then can be applied to accelerate the gradient iterates.

Finally, we notice that the effects of adding more memory units into the standard gradient method in order to improve its convergence speed are unknown. An early study by Polyak [78] suggests that such an idea may bring in some extra performance speedup. But the amount of speedup is lower than the two-step method. In a recent study, it is shown that adding more than a single memory unit, does not improve the convergence speed of the linear consensus iterations [132]. A quantitative answer to this question that considers general convex optimization problems is still missing in the literature.

7.3 Accelerating the ADMM algorithm: quadratic problems

Performance optimization of the ADMM algorithm for solving QP problems was discussed in Chapter 5. We first considered ℓ_2-regularized QP problems in which we obtained the optimal ADMM parameters along with the best possible convergence factors. The analysis provided insights about the parameter optimization of the ADMM algorithm and served as a framework to compare the performance of different first-order optimization algorithms.

Then we shifted our attention to the case of QP problems subject to linear inequality constraints. We showed if the cost function is strictly convex and the constraint matrix is surjective (is of full row-rank) then the associated ADMM algorithm converges linearly and globally to the optimal solution. We also suggested the optimal step-size and relaxation parameters that lead into the best achievable performance. Another performance speedup technique was also considered by efficiently preconditioning the constraint matrix. Many engineering problems involve constraint matrices that are not of full-row rank. For such cases, we proposed the heuristic parameters that improve the performance of the ADMM algorithm.

Several numerical examples elucidated our theoretical findings. We compared the performance of gradient, Heavy-ball, and ADMM methods in the context of ℓ_2-regularized QP problem and discussed the benefits of using each of these optimization algorithms. Furthermore, the performance of ADMM method with our tuning rules was compared to a Nesterov based accelerated ADMM method. Finally, a MPC benchmark with 170 QPs was considered in which we tested several parameter configurations for the ADMM method. The numerical results showed the superiority of our optimal or heuristic parameter selection rules compared to alternatives.

7.3.1 Future work

A first natural extension to the results of the chapter is to consider more general cost functions. A recent study shows that the same linear convergence rate and similar tuning rules also hold if we replace QPs with a more general class of strongly convex functions with Lipschitz continuous gradients [133]. A quantitative study that considers the role of parameter selection for the cases when the ADMM method do not converge linearly is an interesting future topic to investigate.

Another generalization that can be made to the results of the chapter is to relax our assumptions. Especially, one can try to find optimal ADMM parameters for the cases when the Hessian is positive semi-definite (instead of positive definite) or the linear constraint matrix is rank deficit. It is noteworthy to mention that a recent paper addresses some of these issues by considering QPs with linear equality and bound constraints [134].

7.4 Accelerating the ADMM algorithm: distributed quadratic problems

Performance optimization of the ADMM algorithm for solving distributed QP problems was discussed in Chapter 6. Distributed unconstrained QPs were cast as equality-constrained quadratic problems, to which the scaled ADMM method was applied. For this class of problems, the network-constrained scaling corresponds to the usual step-size constant, the relaxation parameter, and the edge-weights of the communication graph. By applying amendable scalings on the constraint matrices, for connected communication graphs, we derived analytical expressions for the optimal step-size, relaxation parameter, and the resulting convergence factor. We showed how these parameters depend on the spectral properties of the underlying communication graph.

In particular, we considered two ADMM compliant distributed QP formulations and discussed their optimal parameter selection rules. We noted while node-variable formulation often offers better performance, it imposes more communication overhead compared to edge-variable formulation. Supposing the optimal step-size and relaxation parameter are chosen, the convergence factor was further improved by properly scaling the edge weights. We formulated several convex programing problems to derive such edge weights.

Numerical examples illustrated the theoretical achievements of the chapter. We compared the best performance of ADMM based distributed averaging algorithms to several state-of-the-art techniques. Moreover, we examined the performance of these algorithms in a distributed setup in which the algorithm parameters are chosen based on imperfect locally available information. Significant performance improvements over alternatives were demonstrated for ADMM algorithms with our performance optimization techniques.

7.4.1 Future work

In this chapter, we considered synchronized ADMM algorithms. The works [135, 136, 137] proposed a variety of ADMM algorithms with asynchronous updates. While the first two works consider the distributed setup in which a randomly chosen node (or group of nodes) is allowed to update at each iteration, the work in [137] is a centralized approach in which nodes coordinate with a master node in a star topology. Similar to Chapter 4, investigating the effects of imperfections such as packet losses, quantization and topology changes on the performance of ADMM algorithm are interesting future directions.

As we noticed in Chapter 6, ADMM formulation is naturally compatible with undirected graphs. The recent work [138] presents a distributed optimization framework in the setting of time-varying directed communications. As another future work, we plan to extend the results to account for directed communications among agents.

Appendix A
Notation

Vectors are denoted with small characters, e.g., x, and are column vectors by default. Matrices, however, are denoted with capital letters, e.g., A. Functions are denoted with Latin characters, e.g., f and g. Sets are mostly denoted with calligraphic letters, e.g., $\mathcal{X}$. A list of symbols in this thesis is presented in the following table.

Symbols	$\triangleq$	Interpretations
$\mathbb{R}$		Set of real numbers
$\mathbb{C}$		Set of complex numbers
$\mathbb{R}_+$		Set of positive real numbers
$\mathbb{R}_{++}$		Set of strictly positive real numbers
$\mathbb{N}$		Set of natural numbers
$\mathbb{N}_0$		Set of natural numbers including zero
$\mathbb{R}^n$		Set of real vectors with n components
$\mathbb{R}^{n\times m}$		Set of real $n \times m$ matrices
$\mathcal{S}^n$		Set of $n \times n$ real symmetric matrices
$\mathcal{S}^n_+$		Set of $n \times n$ real positive semi-definite matrices
$\mathcal{S}^n_{++}$		Set of $n \times n$ real positive definite matrices
ϕ		Empty set
$\mathbf{0}$		The (matrix) vector of all 0 entries
$\mathbf{1}$		The (matrix) vector of all 1 entries
I		Identity matrix
I_n		$n \times n$ Identity matrix
$x^{(k)}$		Value of x at iteration k
x_i		The i-th entry of x
A_{ij}		The element on row i and column j in the matrix A
$Z = \mathrm{diag}(z)$		The diagonal matrix with $Z_{ii} = z_i$ and $Z_{ij} = 0$ for $j \neq i$
$A^\top$		Transpose of the matrix A
$\lVert \cdot \rVert_p$		The vector (matrix) p-norm
$\lVert z \rVert_{p,\star}$		Dual norm, $\lVert z \rVert_{p,\star} = \sup\{z^\top x \mid \lVert x \rVert_p \leq 1\}$
$\lVert x \rVert$		Euclidean norm, $\lVert x \rVert = \sqrt{x^\top x}$

$\|A\|$	Spectral norm, $\|A\| = \sup_{x \neq 0} \|Ax\|/\|x\|$
$\lvert\epsilon\rvert$	(Element-wise) absolute value of $\epsilon \in \mathbb{R}$ ($\epsilon \in \mathbb{R}^n$)
$\lvert\mathcal{A}\rvert$	The cardinality of a set $\mathcal{A}$
$\lambda_i(A)$	The i-th smallest in modulus eigenvalue of A
λ-eigenspace of A	The space spanned by all the eigenvectors corresponding to $\lambda(A)$
$r(A)$	The spectral radius, $r(A) \triangleq \max_\lambda \{\lvert\lambda\rvert \mid Ax = \lambda x\}$
$\langle x, y \rangle$	Inner product of x and y, $x^\top y$
$\nabla f(x)$	Gradient of f evaluated at x
$\nabla^2 f(x)$	Hessian of f evaluated at x
$\mathcal{N}(A)$	The null-space of A, $\{x \in \mathbb{R}^m \mid Ax = \mathbf{0}\}$
$\mathrm{Im}(A)$	The range space of A, $\{y \in \mathbb{R}^n \mid y = Ax,\ x \in \mathbb{R}^m\}$
$\dim(\mathcal{X})$	The dimension of a subspace $\mathcal{X}$
$\mathrm{span}(A)$	The subspace spanned by columns of A
$\lambda(B, D)$	Generalized eigenvalues $(B - \lambda D)v = 0$
$A^\dagger \triangleq (A^\top A)^{-1} A^\top$	The pseudo-inverse of full-column rank matrix A
$\Pi_{\mathrm{Im}(A)} \triangleq AA^\dagger$	The orthogonal projector onto $\mathrm{Im}(A)$
$A \succ 0$ ($A \succeq 0$)	Indicates that A is positive definite (semi-definite)
$\mathrm{diag}\left(\{A_i\}_{i=1}^m\right)$	Block-diagonal matrix with $A_i \in \mathbb{R}^{n \times n}$ in its i-th diagonal block
$\otimes$	Kronecker matrix product
$\mathcal{G}$	The (undirected) graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
$\mathcal{V}$	Set of vertices (nodes or agents)
$\mathcal{E}$	Set of edges (or links)
$\mathcal{N}_i$	The neighbor set of node i, $\mathcal{N}_i \triangleq \{j \neq i \mid \{i, j\} \in \mathcal{E}\}$
Sparsity pattern $\mathcal{A}$	$\mathcal{A} \triangleq \{S \in \mathcal{S}^{\lvert\mathcal{V}\rvert} \mid [S]_{ij} = 0 \text{ if } i \neq j \text{ and } \{i, j\} \notin \mathcal{E}\}$

Consider $B, D \in \mathbb{R}^{n \times n}$ with D being invertible. The generalized eigenvalue problem (B, D) is to find pairs of $\lambda_i \in \mathbb{C}$ and nonzero $v_i \in \mathbb{C}^n$ such that $(B - \lambda_i D)v_i = 0$ for $i = 1, \ldots, n$. The pair λ_i and v_i are termed as a generalized eigenvalue and its associated generalized eigenvector, respectively.

Appendix B
Acronyms

ADMM	Alternating Direction Method of Multipliers
IEEE	Institute of Electrical and Electronics Engineers
KKT	Karush-Kuhn-Tucker
MH	Metropolis-Hastings
MPC	Model Predictive Control
MSE	Mean Square Error
NUM	Network Utility Maximization
PD	Proportional Derivative
QEP	Quadratic Eigenvalue Problem
QP	Quadratic Program
RGG	Random Geometric Graph
SDP	Semi-Definite Program

Bibliography

[1] D. Boyle, D. Yates, and E. Yeatman, "Urban sensor data streams: London 2013," *Internet Computing, IEEE*, vol. 17, no. 6, pp. 12–20, November 2013.

[2] "Flight assembled architecture," Gramazio & Kohler and Raffaello D'Andrea Exhibition, 2011. [Online]. Available: http://raffaello.name/projects/flight-assembled-architecture/

[3] A. S. Nemirovski and M. J. Todd, "Interior-point methods for optimization," *Acta Numerica*, pp. 191–234, 2008.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[5] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[6] A. Simonetto, "Distributed estimation and control for robotic networks," Ph.D. dissertation, Technische Universiteit Delft, 2012.

[7] I. Shames, "Formation control and coordination of autonomous agents," Ph.D. dissertation, The Australian National University, 2010.

[8] B. Johansson, "On distributed optimization in networked systems," Ph.D. dissertation, Royal Institute of Technology, 2008.

[9] M. R. Gholami, "Wireless sensor network positioning techniques," Ph.D. dissertation, Chalmers University of Technology, 2013.

[10] P. Park, "Modeling, analysis, and design of wireless sensor network protocol," Ph.D. dissertation, Royal Institute of Technology, 2011.

[11] P. Di Marco, "Modelin and design of wireless protocols for networked control applications," Ph.D. dissertation, Royal Institute of Technology, 2012.

[12] Z. Zou, "Real-time communication in wireless lossy networks," Ph.D. dissertation, Royal Institute of Technology, 2014.

[13] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52 Issue: 7, pp. 1136–1144, July 2004.

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

[15] S. Low and D. Lapsley, “Optimization flow control - I: Basic algorithm and convergence.” *Networking, IEEE/ACM Transactions on*, vol. 7, no. 6, pp. 861–874, December 1999.

[16] P. Soldati, “On cross-layer design and resource scheduling in wireless networks,” Ph.D. dissertation, Royal Institute of Technology, 2009.

[17] M. Awad, V. Mahinthan, M. Mehrjoo, X. Shen, and J. W. Mark, “A dual-decomposition-based resource allocation for ofdma networks with imperfect csi,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 5, pp. 2394–2403, June 2010.

[18] H. Farhadi, “Coordinated transmission for wireless interference networks,” Ph.D. dissertation, Royal Institute of Technology, 2014.

[19] H. Feyzmahdavian, M. Johansson, and T. Charalambous, “Contractive interference functions and rates of convergence of distributed power control laws,” *Wireless Communications, IEEE Transactions on*, vol. 11, no. 12, pp. 4494–4502, December 2012.

[20] C. Fischione, “Fast-Lipschitz optimization with wireless sensor networks applications,” *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2319–2331, October 2011.

[21] D. Feng, C. Jiang, G. Lim, J. Cimini, L.J., G. Feng, and G. Li, “A survey of energy-efficient wireless communications,” *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 167–178, First 2013.

[22] D. Julian, M. Chiang, D. O’Neill, and S. Boyd, “Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, June 2002, pp. 477–486.

[23] J. Hespanha, P. Naghshtabrizi, and Y. Xu, “A survey of recent results in networked control systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, January 2007.

[24] B. Demirel, Z. Zou, P. Soldati, and M. Johansson, “Modular design of jointly optimal controllers and forwarding policies for wireless control,” *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3252–3265, December 2014.

[25] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, “Accelerated gradient methods and dual decomposition in distributed model predictive control,” *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[26] C. A. Larsson, “Application-oriented experiment design for industrial model predictive control,” Ph.D. dissertation, Royal Institute of Technology, 2014.

[27] A. Alam, “Fuel-efficient heavy-duty vehicle platooning,” Ph.D. dissertation, Royal Institute of Technology, 2014.

[28] J. Araújo, “Design, implementation and validation of resource-aware and resilient wireless networked control systems,” Ph.D. dissertation, Royal Institute of Technology, 2014.

[29] A. Teixeira, “Towards cyber-secure and resilient networked control systems,” Ph.D. dissertation, Royal Institute of Technology, 2014.

[30] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods*. MIT Press, Cambridge, 1998.

[31] M. Zhu, S. Wright, and T. Chan, “Duality-based algorithms for total-variation-regularized image restoration,” *Computational Optimization and Applications*, vol. 47, no. 3, pp. 377–400, 2010.

[32] E. Candes and B.Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[33] E. Candes and M. Wakin, “An introduction to compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, March 2008.

[34] Y. Nesterov, “Gradient methods for minimizing composite functions,” *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[35] P. Tseng, “On accelerated proximal gradient methods for convex-concave optimization,” *submitted to SIAM Journal on Optimization*, 2008.

[36] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, “Composite objective mirror descent,” in *Proceedings of the Twenty Third Annual Conference on Computational Learning Theory (COLT)*, 2010, pp. 14–26.

[37] D. P. Bertsekas, “Incremental proximal methods for large scale convex optimization,” *Mathematical Programming*, vol. 129, no. 2, pp. 163–195, 2011.

[38] Y. Nesterov, “Primal-dual subgradient methods for convex problems,” *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2009.

[39] L. Xiao, “Dual averaging methods for regularized stochastic learning and online optimization,” *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596, 2010.

[40] M. H. Wright, “The interior-point revolution in optimization: history, recent developments, and lasting consequences,” *Bull. Amer. Math. Soc. (N.S)*, vol. 42, pp. 39–56, 2005.

[41] M. Hilbert and P. Lopez, “The world’s technological capacity to store, communicate, and compute information,” *Science*, vol. 332, no. 6025, pp. 60–65, April 2011.

[42] "IBM what is big data? - bringing big data to the enterprise," retrieved 2014-12-20. [Online]. Available: www.ibm.com/software/data/bigdata/

[43] M. Wall, "Big Data: Are you ready for blast-off?" BBC News, March 2014.

[44] E. Ghadimi, I. Shames, and M. Johansson, "Multi-step gradient methods for networked optimization," *Signal Processing, IEEE Transactions on*, vol. 61, no. 21, pp. 5417–5429, November 2013.

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

[46] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer New York, 2006.

[47] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[48] C. Meyer and R. Plemmons, "Convergent powers of a matrix with applications to iterative methods for singular linear systems," *SIAM J. Numer. Anal*, vol. 14, pp. 699–705, 1977.

[49] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[50] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[51] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[52] B. T. Polyak, *Introduction to Optimization*. Optimization Software, 1987.

[53] H. Chen and X. Meng, "Accelerating Nesterov's method for strongly convex functions with Lipschitz gradient," MATH301, Stanford University, Tech. Rep., 2011.

[54] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5523 –5537, November 2011.

[55] J. Eckstein, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," RUTCOR Research Report RRR 32-2012, Tech. Rep., December 2012.

[56] ——, "Parallel alternating direction multiplier decomposition of convex programs," *J. Optim. Theory Appl.*, vol. 80, no. 1, pp. 39–62, January 1994.

[57] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice University CAAM Technical Report ,TR12-14, 2012., Tech. Rep., 2012.

[58] P. van Mieghem, *Graph Spectra for Complex Networks.* Cambridge University Press, 2011.

[59] R. Diestel, *Graph Theory*, 4th, Ed. Springer-Verlag, 2010.

[60] F. R. Chung, *Spectral graph theory.* American Mathematical Soc., 1997, vol. 92.

[61] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, January 2009.

[62] Y. C. Ho, L. Servi, and R. Suri, "A class of center-free resource allocation algorithms," *Large Scale Systems*, vol. 1, pp. 51–62, 1980.

[63] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Opt. Theory and Applications*, vol. 129, no. 3, pp. 469–488, 2006.

[64] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[65] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, D. P. Palomar and Y. C. Eldar, Eds. Cambridge University Press, 2010.

[66] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley, "Notes on decomposition methods," Stanford University, Tech. Rep., 2008.

[67] D. Bertsekas., *Nonlinear Programming.* Athena Scientific, 1999.

[68] E. Ghadimi, M. Johansson, and I. Shames, "Accelerated gradient methods for networked optimization," in *American Control Conference (ACC), 2011.* IEEE, 2011, pp. 1668–1673.

[69] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems.* National Bureau of Standards Washington, DC, 1952.

[70] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, pp. 95–110, 1956.

[71] K. J. Arrow, *Studies in Linear and Non-linear Programming.* Stanford mathematical studies in the social sciences, 1958.

[72] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, no. 1-2, pp. 37–75, 2014.

[73] C. Guzman and A. Nemirovski, "On lower complexity bounds for large-scale smooth convex optimization," *submitted to Journal of Complexity*, 2014.

[74] Y. Drori and M. Teboulle, "Performance of first-order methods for smooth convex minimization: a novel approach," *Mathematical Programming, Series A*, vol. 145, pp. 451–482, 2014.

[75] Q. Lin, Z. Lu, and L. Xiao, "An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization," *ArXiv e-prints*, July 2014.

[76] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1232–1243, May 2014.

[77] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[78] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[79] A. Nemirovsky and D. Yudin, *Problem complexity and method efficiency in optimization Problem*, ser. Interscience Series in Discrete Mathematics. John Wiley, 1983.

[80] P. Ochs, T. Brox, and T. Pock, "iPiasco: Inertial proximal algorithm for strongly convex optimization," *Technical Report*, 2014.

[81] H. Wang and P. Miller, "Scaled Heavy-Ball acceleration of the Richardson-Lucy algorithm for 3D microscopy image restoration," *Image Processing, IEEE Transactions on*, vol. 23, no. 2, pp. 848–854, February 2014.

[82] S. Zavriev and F. Kostyuk, "Heavy-ball method in nonconvex optimization problems," *Computational Mathematics and Modeling*, vol. 4, no. 4, pp. 336–341, 1993.

[83] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *eprint arXiv:1408.3595*, 2014.

[84] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer, 1998, vol. 317.

[85] Z. Allen-Zhu and L. Orecchia, "Linear coupling of gradient and mirror descent: A novel, simple interpretation of nesterov's accelerated method," *ArXiv e-prints*, July 2014.

[86] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95 Issue: 1, pp. 215–233, 2007.

[87] S. Kar, J. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: nonlinear observation models and imperfect communication," *Information Theory, IEEE Transactions on*, vol. 58, no. 6, pp. 3575–3605, June 2012.

[88] D. Goodman and N. Mandayam, "Power control for wireless data," *Personal Communications, IEEE*, vol. 7 Issue:2, pp. 48–54, 2000.

[89] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, 1986.

[90] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *44th Annual Allerton Conference on Communication, Control, and Computation*, 2006, pp. 952–959.

[91] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-raphson consensus for distributed convex optimization," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, December 2011, pp. 5917–5922.

[92] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization-I: Algorithm," *Automatic Control, IEEE Transactions on*, vol. 58, no. 9, pp. 2162–2175, 2013.

[93] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, May 2005.

[94] O. Devolder, F. Glineur, and Y. Nesterov, "Double smoothing technique for large-scale linearly constrained convex optimization," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 702–727, 2012.

[95] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[96] P. Maréchal and J. Ye, "Optimizing condition numbers," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 935–947, 2009.

[97] Z. Lu and T. K. Pong, "Minimizing condition number via convex programming," *SIAM J. Matrix Anal. Appl.*, vol. 32, pp. 1193–1211, 2011.

[98] L. Vandenberghe, "Course notes for optimization methods for large-scale systems, EE236C, dual decomposition chapter," UCLA, Tech. Rep., 2012.

[99] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, 1985.

[100] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, June 2003.

[101] B. Oreshkin, M. Coates, and M. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *Signal Processing, IEEE Transactions on*, vol. 58, no. 5, pp. 2850–2865, May 2010.

[102] D. M. Young, “Second-degree iterative methods for the solution of large linear systems,” *Journal of Approximation Theory*, 1972.

[103] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse, “Analysis of accelerated gossip algorithms,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, December 2009, pp. 871–876.

[104] G. H. Golub and R. S. Varga, “Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods,” *Numerische Matematik*, vol. 3, pp. 147–156, 1961.

[105] J. B. H. Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms II.* Springer, 1996.

[106] D. Falcao, F. Wu, and L. Murphy, “Parallel and distributed state estimation,” *Power Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 724–730, May 1995.

[107] A. Nedic, A. Ozdaglar, and P. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 922–938, April 2010.

[108] B. O’Donoghue, G. Stathopoulos, and S. Boyd, “A splitting method for optimal control,” *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 6, pp. 2432–2442, November 2013.

[109] J. Yang and Y. Zhang, “Alternating direction algorithms for l_1-problems in compressive sensing,” *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[110] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, “An ADMM algorithm for a class of total variation regularized estimation problems,” in *16th IFAC Symposium on System Identification*, 2012, pp. 83–88.

[111] M. Figueiredo and J. Bioucas-Dias, “Restoration of poissonian images using alternating direction optimization,” *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3133–3145, 2010.

[112] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based distributed support vector machines,” *J. Mach. Learn. Res.*, vol. 99, pp. 1663–1707, 2010.

[113] S. Joshi, M. Codreanu, and M. Latva-aho, “Distributed SINR balancing for MISO downlink systems via the alternating direction method of multipliers,” in *Proceedings of 11th International Symposium on Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), Tsukuba Science City, Japan*, May 2013, pp. 318–325.

[114] M. Hong and Z.-Q. Luo, “On the linear convergence of the alternating direction method of multipliers,” *ArXiv e-prints*, 2013.

[115] D. Boley, “Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs,” *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2183–2207, 2013.

[116] L. Lasdon, *Optimization theory for large systems.* Courier Dover Publications, 1970.

[117] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, “Optimal scaling of the ADMM algorithm for distributed quadratic programming,” in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec. 2013, pp. 6868–6873.

[118] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the ADMM in decentralized consensus optimization,” *Signal Processing, IEEE Transactions on*, vol. 62, no. 7, pp. 1750–1761, April 2014.

[119] H. Ohlsson, L. Ljung, and S. Boyd, “Segmentation of ARX-models using sum-of-norms regularization,” *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[120] T. Goldstein, B. O’Donoghue, S. Setzer, and R. Baraniuk, “Fast alternating direction optimization methods,” UCLA, Tech. Rep., 2012.

[121] K. Johansson, “The quadruple-tank process: a multivariable laboratory process with an adjustable zero,” *Control Systems Technology, IEEE Transactions on*, vol. 8, no. 3, pp. 456–465, May 2000.

[122] [Online]. Available: https://www.dropbox.com/s/x2w74mpbezejbee/MPC_QP_quadtank_170_Np5_SxQ.mat

[123] F. Farokhi, I. Shames, and K. H. Johansson, “Distributed MPC via dual decomposition and alternative direction method of multipliers,” in *Distributed Model Predictive Control Made Easy*, ser. Intelligent Systems, Control and Automation: Science and Engineering, J. M. Maestre and R. R. Negenborn, Eds. Springer, 2013, vol. 69.

[124] C. Conte, T. Summers, M. Zeilinger, M. Morari, and C. Jones, “Computational aspects of distributed optimization in model predictive control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, December 2012, pp. 6819–6824.

[125] M. Annergren, A. Hansson, and B. Wahlberg, “An ADMM algorithm for solving ℓ_1 regularized MPC,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, December 2012, pp. 4486–4491.

[126] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, “Distributed ADMM for model predictive control and congestion control,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, December 2012, pp. 5110–5115.

[127] F. Tisseur and K. Meerbergen, “The quadratic eigenvalue problem,” *SIAM Review*, vol. 43, no. 2, pp. 235–286, 2001.

[128] E. Jury, *Theory and Application of the z-Transform Method.* Huntington, New York: Krieger Publishing Company, 1974.

[129] S. K. Butler, *Eigenvalues and structures of graphs*. University of California, San Diego, ProQuest, UMI Dissertations Publishing, 2008.

[130] M. Penros, *Random Geometric Graphs*. Oxford Studies in Probability, 2003.

[131] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, March 2000.

[132] A. Sarlette, "Adding a single memory per agent gives the fastest average consensus," *ArXiv e-prints*, December 2014.

[133] P. Giselsson and S. Boyd, "Metric selection in Douglas-Rachford splitting and ADMM," *ArXiv e-prints*, October 2014.

[134] A. U. Raghunathan and S. Di Cairano, "ADMM for convex quadratic programs: Linear convergence and infeasibility detection," *ArXiv e-prints*, November 2014.

[135] E. Wei and A. Ozdaglar, "On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," *ArXiv e-prints*, July 2013.

[136] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," *ArXiv e-prints*, March 2013.

[137] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1701–1709.

[138] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 601–615, March 2015.