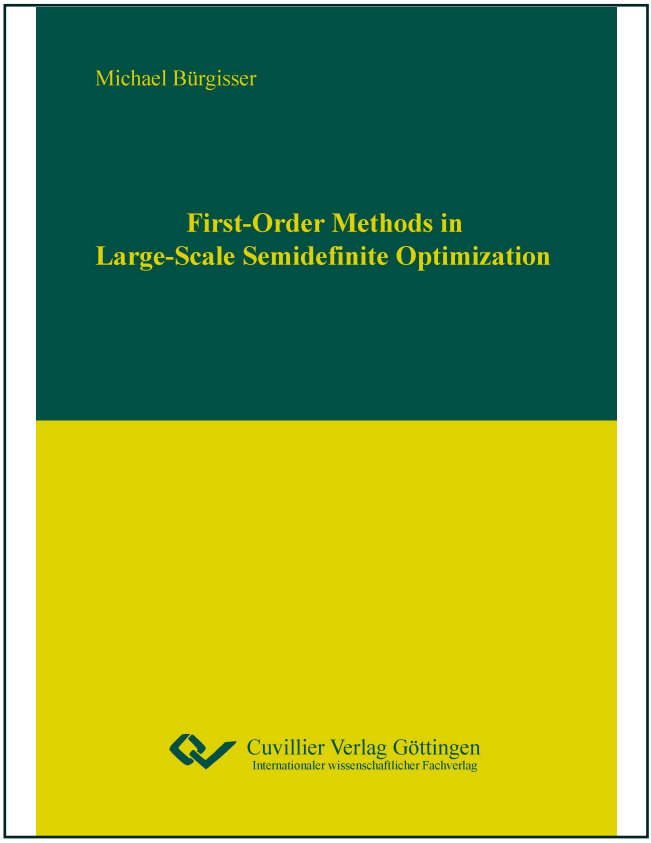

Chapter 1

Introduction

1.1 Motivation

Semidefinite Optimization is generally appreciated as the most exciting innovation in Convex Programming in the 1990's; see for instance [Fre99]. It constitutes a generalization of Linear Programming, where the cone of nonnegative vectors is replaced by the cone of positive semidefinite matrices. That is, we are supposed to minimize a linear function over a set of positive semidefinite matrices, where this set of matrices is described by a collection of linear inequalities.

Nowadays, an impressive amount of real-life optimization problems, from nearly all fields of engineering, can be represented or approximated by semidefinite optimization problems. For instance, Semidefinite Optimization is used in Control [BGFB94], in Structural Design [BTN97], or in Statistics [BV04], only to name a few. Probably the most famous applications are semidefinite relaxations of hard combinatorial problems; see for instance [GW95, NRT99].

Many modern real-life optimization problems – including up-to-date semidefinite optimization problems – are of very large-scale. Generally speaking, the emergence of large-scale optimization problems has two reasons. One the one hand, it has become relatively easy and cheap to collect and store huge amounts of data over the last few years. For instance, we may mention here web-based social platforms, customer bonus cards, or surveillance cameras, where massive amounts of data are collected every day. On the other hand, the real-life systems that we study and try to model as optimization problems are getting more and more complex. The relatively new

field of Systems Biology serves as a prime example here. In Systems Biology, researcher try to analyze, reconstruct, and understand highly complex biological systems using – among others – optimization tools; see for instance [HNTW09, PRP04, TS08].

Motivated by these facts, we study the practical tractability of large-scale semidefinite optimization problems in this thesis. Semidefinite optimization problems that involve matrices of size a few hundred times a few hundred and with a few thousand constraints can be solved up to a very high accuracy by Interior-Point methods [Ali95, NN93]. However, if we go beyond this problem size, it takes Interior-Point methods too long to derive an approximate solution in practice. More formally, Interior-Point methods have a theoretical worst-case running time that is of the order

$$\mathcal{O}\left(\sqrt{m+n}\left[mn^3 + m^2n^2 + m^3\right]\ln\left[(m+n)/\epsilon\right]\right),$$

where m denotes the number of constraints, n the matrix size, and ϵ the target accuracy; see Chapter 6 for a review. The logarithmic dependence on ϵ makes Interior-Point methods a tailored tool for finding highly accurate solutions, whereas the fast – although polynomial – growth in m and n limits the size of problems that can be handled in practice. In order to comply with the modern trends in Semidefinite Optimization, that is, in order to be able to solve large-scale semidefinite optimization problems in practice, the following question arises immediately:

"Assume that we tolerate a moderate complexity increase with respect to the solution accuracy ϵ, say from $\ln[1/\epsilon]$ to $1/\epsilon$, or even to $1/\epsilon^2$. Does there exist an algorithm for solving semidefinite optimization problems with a running time that is below the complexity estimate of Interior-Point methods with respect to the matrix size and the number of constraints?"

This is the opening question of this thesis. Note that a moderate solution accuracy is usually not a barrier in practice, as two or three accuracy digits are typically sufficient for practical applications.

In 2007, Arora and Kale [AK07] introduced an alternative approach for solving slightly structured large-scale semidefinite optimization problems. They perform a Binary Search over the objective function values. At every iteration of this search, they are supposed to answer a feasibility question. The answer to this question is derived by a Matrix Multiplicative Weights Update method. In total, they need to perform, roughly speaking, $\mathcal{O}\left(\ln[1/\epsilon]/\epsilon^2\right)$ iterations of this method in order to find a solution with approximation error $\epsilon > 0$. By "roughly speaking", we refer to the fact that the complexity result depends also on other problem parameters such as the scaling of the problem. At every iteration of this method, the exponentiation of a symmetric matrix and some other computations not exceeding the cost of $\mathcal{O}(mn^2)$ arithmetic

operations are supposed to be carried out. Provided that we compute the exponential through an eigendecomposition of the symmetric matrix, we end up with a method that requires in total, roughly speaking,

$$\mathcal{O}\left([n^3 + mn^2]\ln[1/\epsilon]/\epsilon^2\right)$$

arithmetic operations to compute an approximate solution. In case of sparse matrices, the term mn^2 in the above complexity result can be reduced in accordance to the sparsity pattern of the matrices. Arora and Kale [AK07] also discuss some strategies to replace the exact value of the matrix exponential by a random approximation in their Matrix Multiplicative Weights Update method. For instance, they present a random approximation that is based on the Johnson-Lindenstrauss Lemma [JL84] and on an appropriate truncation of the matrix exponential Taylor series. Using this randomization procedure, we end up with an algorithm whose complexity grows with the order of $\mathcal{O}(\ln[1/\epsilon]/\epsilon^5)$ with respect to solution accuracy $\epsilon > 0$. Because of this fast growth in ϵ, we do not elaborate more on this randomization of their method in this thesis.

The Matrix Multiplicative Weights Update method, which was introduced simultaneously by Arora and Kale [AK07] and by Warmuth et al. [TRW05, WK06], can be seen as a generalization of Multiplicative Weights Update methods to matrices; see [AHK05] for a survey of these methods. The basic concept of Multiplicative Weights Update methods plays a crucial role in the development of algorithms in Machine Learning and in Data Mining, or, more generally, in Computer Science. For instance, AdaBoost [FS97] – one of the top ten Data Mining algorithms (see [WKR$^+$07] for the complete list) – is based on the Hedge algorithm [FS97], which follows the same basic construction as Multiplicative Weights Update methods. Interestingly enough, the Hedge algorithm and (Matrix) Multiplicative Weights Update methods have the same analytical complexity as Dual Averaging schemes [Nes09]: the iteration count of all these methods grows with the order $\mathcal{O}(1/\epsilon^2)$. This observation gives rise to the conjecture that the Hedge algorithm and (Matrix) Multiplicative Weights Update methods are particular instances of Dual Averaging schemes.

Let us now get back to Semidefinite Optimization. When we compare the complexity results of Interior-Point methods and Arora and Kale's scheme for solving slightly structured semidefinite optimization problems, we make the following two observations. On the one hand, the complexity of Arora and Kale's method grows only linearly in the number of constraints. This is in sharp contrast to Interior-Point methods, where the complexity result grows with the power 3.5 with respect to m. On the other hand, there is a tremendous gap (namely, a factor of $1/\epsilon^2$) in the worst-case complexity bounds of

Interior-Point methods and Arora and Kale's scheme. These observations result in the following adaption of the opening question:

"Does there exist a method for solving slightly structured semidefinite optimization problems whose iteration count grows, roughly speaking, with the order $\mathcal{O}(1/\epsilon)$ and whose iteration cost is given by $\mathcal{O}(n^3 + mn^2)$?"

Smoothing Techniques [Nes05] and Mirror-Prox methods [Nem04a] were introduced recently. These powerful methods can be applied to non-differentiable convex problems that have a very specific structure, namely to a huge variety of matrix saddle-point problems. The matrix saddle-point problems that we consider in this thesis correspond all to the problem of minimizing the maximal eigenvalue of convex combinations of symmetric matrices. When we speak of matrix saddle-point problems, we henceforth automatically refer to the maximal eigenvalue minimization problem. Interestingly enough, the iteration count of Smoothing Techniques and Mirror-Prox methods grows with the desired order $\mathcal{O}(1/\epsilon)$, where ϵ denotes the solution accuracy. However, these powerful methods are not directly applicable to generic semidefinite optimization problems, as these problems do not satisfy the structural requirements of the algorithms. Nevertheless, this approach seems to be very promising, because Chudak and Eleutério [CE05] successfully applied these methods to large-scale linear optimization problems with up to millions of variables and constraints. As linear programs constitute a particular subclass of semidefinite optimization problems, it seems to be very natural to extend their approach to the more general class.

In the situation where the decision variables are matrices, Smoothing Techniques and Mirror-Prox methods suffer the same computational bottleneck as the Matrix Multiplicative Weights Update algorithm: they need to determine the exponential of a symmetric $(n \times n)$-matrix at every iteration; see [Nes07] and [Nem04a] for the details. There exists plenty of different ways to determine or to approximate these exponentials; see [ML03] for the classical survey on this topic. Standardly, this exponentiation is performed through an eigendecomposition of the symmetric matrix, requiring $\mathcal{O}(n^3)$ arithmetic operations and, consequently, hampering the resolution of problems with huge matrices. In order to extend the class of matrix saddle-point problems that can be successfully handled by Smoothing Techniques or Mirror-Prox methods in practice, we need to find strategies that allow us to replace this matrix exponential by an approximation that can be computed faster.

1.2 Goals of the thesis

Let us list the goals of this thesis.

A. Apply Smoothing Techniques and Mirror-Prox methods to semidefinite optimization problems

In order to close the dramatic gap between the complexity results of Interior-Point methods and Arora and Kale's scheme [AK07] with respect to the solution accuracy, we want to apply Smoothing Techniques and Mirror-Prox methods to slightly structured semidefinite optimization problems. However, these problems do not match the structural requirements of Smoothing Techniques and Mirror-Prox methods. In a preliminary step, we therefore need to find an appropriate reformulation of semidefinite optimization problems, that is, we are supposed to recast them as matrix saddle-point problems.

B. Reduce the number of iterations of Smoothing Techniques in practice

Smoothing Techniques are a two-stage procedure. In a first step, an appropriate smooth approximation of the objective function is built. This construction exploits the very specific form by which non-differentiability enters the problem. In a second step, we apply an optimal First-Order method [Nes04, Nes05] to the smooth auxiliary problem. At every iteration of this optimal First-Order method, the Lipschitz constant of the gradient of the smoothed objective function is used to determine the next iterate. Clearly, this constant is a global parameter of the problem and might be very pessimistic for the local environment of the algorithm's current iterate. We want to derive a strategy that allows us to replace this global constant by local estimates in the optimal First-order method [Nes04, Nes05].

C. Reduce the iteration cost in Mirror-Prox methods

As pointed out above, both Smoothing Techniques and Mirror-Prox methods require the computation of a matrix exponential at every iteration when applied to matrix saddle-point problems. When we solve problems with huge matrices, this operation becomes critical with respect to the running time of the method. We want to overcome this difficulty by replacing the exact value of the matrix exponential in Mirror-Prox methods by a randomized approximation that can be computed faster. We perform this discussion for Mirror-Prox methods, as this topic was studied in a joint project with Arkadi Nemirovski, the designer of Mirror-Prox methods.

D. Interpret the Hedge algorithm as a Dual Averaging scheme

As shown in [CBL06], the Hedge algorithm can be seen as a particular instance of Mirror-Descent methods [NY83], which are a subclass of Dual Aver-

aging schemes. In this thesis, we want to study this interpretation, point out its severe inconsistencies, and give a complete and consistent discussion of the Hedge algorithm in the context of Dual Averaging schemes. In particular, given the knowledge we gain from this new perspective on the Hedge algorithm, we hope to define alternative versions of this scheme, which perform even more successfully in theory and in practice than the vanilla method.

1.3 Structure of the thesis

This thesis is split in three parts.

Part I: Analytical complexity of solution methods in Convex Optimization

In Part I of this thesis, we lay the theoretical and methodical foundation of this thesis. We introduce general optimization problems, review all properties of both the objective function and the feasible set which are relevant for this thesis, and study the computational tractability of optimization problems in Chapter 2. In Chapter 3, we discuss some Black-Box Optimization methods, namely Dual Averaging schemes, Primal-Dual Subgradient methods, Mirror-Descent algorithms, and optimal First-Order methods. In particular, we present a refinement of optimal First-Order methods which complies with Goal B. We conclude Part I by reviewing Smoothing Techniques, Mirror-Prox schemes, and Interior-Point methods in Chapter 4.

Part II: A new perspective on the Hedge algorithm

Part II consists only of Chapter 5 and addresses exclusively Goal D. In this chapter, we recast the Hedge algorithm in the context of Dual Averaging schemes and derive three new versions of this method, which have theoretical convergence guarantees that are better or at least as good as the convergence result for the vanilla scheme. Numerical results show that all these modified methods perform better than their original counterpart in practice.

Part III: Approximately solving large-scale semidefinite optimization problems

Part III represents the core of this thesis. In Chapter 6, we give an introduction to large-scale Semidefinite Optimization and derive the full complexity result of Interior-Point methods for semidefinite optimization problems. We recast the Matrix Multiplicative Weights Update method in the context of Dual Averaging schemes and discuss the approach of Arora and Kale [AK07]

for solving slightly structured semidefinite optimization problems in Chapter 7. The first part of this chapter can be seen as a generalization of some of the results from Chapter 5 to a matrix setting. In Chapter 8, we achieve Goal A by reformulating slightly structured semidefinite optimization problems in a form to which we can apply not only Primal-Dual Subgradients methods and Mirror-Descent schemes, but also advanced tools such as Smoothing Techniques and Mirror-Prox methods. That is, we recast these problems as matrix saddle-point problems. In the same chapter, we discuss the complexity of Mirror-Descent schemes when applied to these problems. We particularize Smoothing Techniques for matrix saddle-point problems in Chapter 9. In particular, we obtain a procedure that can be used to solve slightly structured semidefinite optimization problems. Importantly, the complexity estimate of this procedure is of the form $\mathcal{O}(1/\epsilon)$ with respect to the solution accuracy $\epsilon > 0$. In Chapter 10, we study the application of Mirror-Prox methods to matrix saddle-point methods and discuss a randomized computation of matrix exponential approximations. We conclude this part by showing some numerical results in Chapter 11.
The conclusions and an outlook are presented in Chapter 12. In the Appendix, we give a short introduction to regular norms and collect some technical proofs.