Chapter 1

Introduction

1.1 Linear Programming, Linear Complementarity and Semidefinite Programming Problems

Whether we formulate them mathematically or not, we solve several optimization problems in the course of an average day: we try to find the shortest route to get to work, to get the best grades possible at school without missing too many parties, to find the cheapest plane ticket for a trip to Europe, to determine the best age to get married, etc.

Not only our daily lives but also the daily operations of manufacturing and service industries involve the solution of decision problems where one tries to optimize (i.e., minimize or maximize) a measure of accomplishment. This measure is called the *objective function*. Furthermore, this task has to be accomplished without violating physical, technical, or logical *constraints*. These decision problems are called *optimization* problems.

Such wide applicability makes optimization a very important component of the decision-making process. This study concentrates on a particular family of solution methods; the *interior-point methods* for the solution of several specially structured classes of *convex* optimization problems. An optimization problem is said to be convex if it involves the minimization of a convex objective function over a convex set.

Following this introductory chapter, we consider the *linear programming* (LP) problem in Chapter 2. LP involves the optimization of a *linear* objective function of a certain number of variables over a set defined by *linear* constraints (which may be equalities or inequalities). Formally, an LP can be represented in the following form:

$$\begin{array}{rrcl} \min_x & c^T x & & \\ & Ax & = & b \\ & x & \geq & 0, \end{array} \tag{1.1}$$

where A is the $m \times n$ *constraint matrix*, b is the m-dimensional *right-hand-side vector*, c is the n-dimensional *cost vector*. A, b, and c are given and x is the n-dimensional *variable vector*. The formulation (1.1) is the standard form of LP.

Linear programming is arguably the best known and the most frequently solved optimization problem. It owes its fame mostly to its great success; real world problems coming from as diverse disciplines as sociology, finance, transportation, economics, production planning, and airline crew scheduling have been formulated and successfully solved as LPs.

The importance and extensive use of LPs also come from the fact that the solution of certain problems that are not linear programs is often carried out by solving a sequence of related linear programs. Such problems include *integer programming* problems where certain indivisibilities are imposed on some of the variables, and *nonlinear programming* problems, where the objective function or the constraints may be nonlinear. In addition, at the modeling stage even when the problem appears to exhibit nonlinearities, the existence of uncertainties in the data and the model often takes away the added accuracy of a nonlinear model and an approximate linear model (which is much easier to solve) may be preferred.

On the other hand, some classes of problems require nonlinear formulations and when these problems have certain special structures they may be almost as easy to solve as linear programs. One of these classes is the *linear complementarity problem* (LCP) that we study in both Chapter 2 and Chapter 3. Given a matrix $M \in \Re^{n \times n}$ and a vector $q \in \Re^n$, the LCP is defined as the problem of finding vectors x and y satisfying

$$\begin{aligned} y - Mx &= q \\ x \geq 0, y \geq 0, x^T y &= 0, \end{aligned} \tag{1.2}$$

or possibly showing that no such solution exists. In general LCPs are more difficult to solve than LPs. However, when the M matrix has special properties such as being positive semidefinite polynomial-time algorithms that are also effective in practice are available. It is easy to see that LP is a special case of LCP. We note that not only linear programming but also quadratic programming problems can be formulated as LCPs, as well as bimatrix game problems, market equilibrium problems, and American option valuation problems.

We consider a slightly more general form of the LCP in Chapter 3, namely the *horizontal linear complementarity problem* (HLCP). This problem seeks two

n-dimensional vectors x and y satisfying

$$\begin{aligned} Ny - Mx &= q \\ x \geq 0, y \geq 0, x^T y &= 0, \end{aligned} \tag{1.3}$$

given $n \times n$ real matrices N, M and an n-dimensional vector q. The term *horizontal* is inspired by the geometric shape of the $n \times 2n$ constraint matrix $[N, -M]$. Clearly, when N is the $n \times n$ identity matrix, the HLCP is the same as the LCP. Also when N is invertible, the first equation of (1.3) can be replaced by $y - N^{-1}Mx = N^{-1}q$, which again leads to an equivalent LCP. However, N is often singular for the HLCPs appearing in practice and they need a separate analysis. HLCPs arise in electrical engineering in the study of resistive piecewise circuits and are also equivalent to systems of piecewise linear equations. Furthermore, HLCPs provide a unifying framework for the study of interior-point methods for certain classes of optimization problems.

In Chapter 4, we consider another class of convex optimization problems, namely the *semidefinite programming* (SDP) problem. SDP is different from the previous classes we mentioned in the sense that the variables in SDP are matrices as opposed to vectors for LP, LCP, and HLCP. These matrix variables are restricted to lie in the cone of positive semidefinite matrices, instead of the non-negative orthant. Yet, SDP is very similar to LP and it can be represented in the following standard form:

$$\begin{aligned} \min_X \quad & C \bullet X \\ A_i \bullet X &= b_i, \quad i = 1, \dots, m \\ X &\succeq 0, \end{aligned} \tag{1.4}$$

where $A_i \in S\mathbb{R}^{n \times n}$, $b \in \mathbb{R}^m$, $C \in S\mathbb{R}^{n \times n}$ are given, and $X \in S\mathbb{R}^{n \times n}$ is the variable matrix. In the formulation above $S\mathbb{R}^{n\times n}$ denotes the space of $n \times n$ symmetric matrices, $X \succeq 0$ indicates that X is a positive semidefinite matrix and $G \bullet K := \mathbf{Tr}\, G^T K$ is the trace of the matrix $G^T K$. Examples of SDPs include matrix norm minimization problems, relaxation of the maximum-cut problem, and smallest or largest eigenvalue problems, among others.

Despite the visual similarity between (1.1) and (1.4), these two problems are different in nature. In particular, SDPs are often nonpolyhedral and nonsmooth optimization problems. However, most interior-point methods developed for LPs can be extended to the framework of SDPs with little extra effort and that is why we concentrate on these problems in Chapter 4.

The last chapter presents the numerical results obtained from the implementation of the infeasible-interior-point potential-reduction algorithm developed in Chapter 2. We solved all of the feasible NETLIB problems succesfully and in a relatively low number of iterations. We also succesfully detect infeasibility in all but one of the infeasible problems in the NETLIB set. As opposed to most other interior-point implementations where detection of infeasibility is usually accomplished by heuristic techniques (and therefore is not always reliable), we generate approximate proofs of infeasibility for these problems using approximate versions of the Farkas' lemma.

1.2 Interior-Point Methods and the Gap between Theory and Practice

The first method developed to solve linear programming problems was the *simplex method* developed by George Dantzig [Dan51,Dan63]. The simplex method not only proved to be very efficient in practice but also facilitated the development of duality theory, post-optimality analysis, and decomposition techniques.

The simplex method remained the champion for solving LPs even after Klee and Minty established that it is not a polynomial-time algorithm [KM72]. This means that for the simplex method it is not possible to bound the computation time required to solve any given LP instance by a polynomial function of the size of a given problem. Indeed, Klee and Minty generated pathological problem instances for a variant of the simplex method where exponentially many iterates are needed for the growing number of variables. Such instances exist for most variants of the method while the question of the existence of a simplex variant with polynomial-time complexity remains open to this date.

As interest in complexity theory grew, it became clear that a "good" algorithm for a problem class should have polynomial complexity. However, there exist several important classes of optimization problems such as the *traveling salesman problem* and the *knapsack problem* for which no polynomial-time algorithm has been developed yet. For LP, the first polynomial-time algorithm was due to Khachiyan [Kha79]. His approach was a modification of the *ellipsoid method* initially designed for nonlinear problems by Shor, Yudin, and Nemirovsky. Although the method has a polynomial-time worst-case complexity it fell far short of the expectation that it would be faster than the simplex method. Today, the ellipsoid method has been abandoned in practice and is used mostly for theoretical considerations, see [GLS86].

A more important development for both the history of linear programming and this dissertation was the publication of Karmarkar's paper in 1984 describing a new polynomial-time *interior-point* algorithm. Interior-point (IP) algorithms are iterative algorithms which start from an initial feasible solution and generate points that satisfy the equality constraints and strictly satisfy the inequality constraints while approaching the optimal solution.

Some of the classical methods developed for nonlinear programming were interior-point methods, namely the *logarithmic barrier function method* of Frisch [Fri55a, Fri55b], and later of Fiacco and McCormick [FM68], the *method of centers* of Huard [Hua67], and the *affine-scaling method* of Dikin [Dik67]. As the above references show, interior-point methods (IPMs) for mathematical programming were fairly developed by the late 60s. However, they were not considered as a viable alternative to the simplex method to solve linear programming problems until Karmarkar's work.

The response of the mathematical programming research community to Karmarkar's paper was impressive. In the following 12 years more than 1500 papers were published on the subject. A comprehensive bibliography of the work done on this subject until 1991 is provided by Kranich [Kra91]. Unsurprisingly, this enormous amount of theoretical work was accompanied by very successful implementations of IPMs. Lustig, Marsten and Shanno [LMS94], who developed the largely acclaimed primal-dual predictor-corrector interior point code OB1, give a history of these implementations and describe how the competition with IPMs led to the improvement of existing simplex codes as well.

Interior-point methods come in different flavors. Although different IPMs share the idea of choosing points iteratively in the relative interior of the feasible region (or the nonnegative orthant, in the case of *infeasible-interior-point methods*) to approach the optimal face, the way they choose the next point given the current point may differ considerably. It is possible to categorize the IPM's on this basis. Den Hertog [dH94] uses the following four categories:

- Path-following methods ;
- Affine-scaling methods ;
- Projective potential-reduction methods ;
- Affine potential-reduction methods.

We note that although this order is historical for the introduction of these methods, their full development has followed a different chronology. It is also important to note that one can distinguish between IPMs according to whether they are primal-only, dual-only algorithms or primal-dual algorithms, and whether they are feasible IPMs or infeasible IPMs.

Affine-scaling methods that do not use any safeguards have not been shown to have polynomial complexity, and despite their initial popularity they are now seldom used in practice. Therefore we will not consider these methods in this dissertation. On the other hand both path-following methods and potential-reduction methods possess the important property of polynomial complexity and are very efficient in practice. Path-following methods generate iterates that follow a certain trajectory in the interior of the feasible region (the so-called *central path*) more or less closely. The limit point of the central path is an optimal solution to the problem, therefore one can approach it by following the central path. On the other hand, potential-reduction methods combine the objective function and the constraints in a certain merit function—similar to the Lagrange function in nonlinear optimization—and they progress to an optimal solution simply by decreasing this merit function as much as possible in every iteration. We will see more of potential-reduction methods in Chapter 2.

One puzzling consequence of the simultaneous work on the theory and implementation of interior-point methods was the existence of significant differences between the algorithms that had the best complexity and the ones with best computational performance. Part of the later research focused on narrowing this gap; the algorithm we describe in Chapter 2 is motivated by this goal, too.

Some of the initial efforts to improve the performance of interior-point implementations concentrated on removing the requirement to start at a feasible interior point, since finding such an initial point is often a difficult task. In theoretical analyses, this issue is usually handled by augmenting the problem in an artificial way (as in the "big M" method), or by solving a feasibility problem first to get a starting feasible interior point (as in the "two phase" method). A more practical alternative is to allow the iterates to violate the equality constraints while they are still forced to satisfy the inequality constraints strictly and then to target feasibility and optimality simultaneously.

Infeasible-interior-point methods (IIPMs) were first introduced into practical implementations and generated a major advancement of the state-of-the-art. One of the earlier references on the success of the IIPMs is a paper by Lustig, Marsten, and Shanno [LMS91] where they describe computational results obtained from their highly successful code OB1[1]. However, until late 1991 these practical methods were not shown to have the theoretical properties of their *feasible* counterparts.

[1]The study that led to OB1 won the Orchard-Hays Prize of the Mathematical Programming Society in 1991. OB1 retired in 1993 after version 6.0. The barrier solver of the popular commercial package CPLEX is based on OB1.

First, Kojima, Megiddo, and Mizuno [KMM93] established the global convergence of IIPMs. Later, Zhang [Zha94] and Mizuno [Miz94] proved polynomial-time convergence results for IIPMs. Several other articles followed these two, improving the complexity bounds and/or relaxing some of the assumptions about the problem or the initial point used by the algorithms. Our understanding of the superior performance of the IIPMs through theoretical analysis was furthered by Wright's demonstration of the superlinear convergence of such methods [Wri94].

Despite this intensive theoretical effort some issues remained unresolved. Most importantly, the algorithms analyzed by the aforementioned authors fell short of the task of efficient detection of infeasibility should they encounter primal or dual infeasible problems. Roughly speaking, this drawback is due to the fact that these algorithms put all their effort into attaining optimality, and can guarantee a detection of infeasibility only when they start from an initial point with huge entries. Such initial points are never used in practical implementations, and this again creates a gap between theory and practice.

On a related issue, indications of infeasibility of the primal or the dual problems provided by most IIPMs are "certificates" that all *optimal* solutions to the primal or the dual problem must have large norm. Such a conclusion is not totally satisfactory since it does not necessarily indicate that the problem is likely to be infeasible, but only that it is likely to be ill-conditioned. A more satisfactory indication of infeasibility would be of the form that primal (or dual) *feasible* solutions, if they exist, must have large norm.

Moreover, most of the IIPMs in the literature are path-following methods. As mentioned above, in this class of methods iterates follow the *central path* some-what closely to an optimal solution. This impractical restriction is introduced to ensure polynomial-time convergence but turns out to be unnecessary in practice and is often eliminated in implementations, contributing to the gap between theory and practice. Additionally, most primal-dual interior-point methods restrict the primal and dual updates to use the same step size in each iteration to obtain theoretical convergence results, while this restriction is also ignored in state-of-the-art implementations.

To resolve these issues, Chapter 2 analyzes potential-reduction methods. In these methods, the choice of the search direction and step sizes, as well as the theoretical complexity analysis are all driven by the potential function. This brings theory and practice much closer. Motivated by this property of the potential-reduction methods we develop an infeasible-interior-point primal-dual potential-reduction algorithm for linear programming problems. This method either finds the solutions for both the primal and dual LPs, or establishes the infeasibility of at least one of these problems without resorting to any regularity assumptions regarding the data (such as the existence of feasible solutions, strictly feasible solutions, or nondegeneracy) or to big M penalty parameters as is customary in the literature.

Additionally, the algorithm we describe ensures a constant reduction (where the constant may depend on the number of variables) in the potential function from any interior-point and also allows the use of *any* heuristic to reduce the potential function further at every iteration. The central-path need not be followed, and different step sizes can be used for primal and dual updates. This algorithmic flexibility coupled with polynomial convergence results narrows the gap between theory and practice for IPMs.

Chapter 5 contains the results of the numerical experiments obtained from a preliminary implementation of the algorithm developed in Chapter 2 in a MATLAB environment. The implementation is fast and robust and the results are encouraging.

1.3 Reduction of Horizontal LCPs to Standard LCPs

One of the basic approaches in mathematics to solving difficult problems is to reduce them to other problems for which solution methods are already available. This practical technique motivates our study in Chapter 3. Namely, we analyze how the harder and more general horizontal linear complementarity problem which is also instrumental in the study of interior-point methods can be reduced to the simpler and better studied (standard) linear complementarity problem.

Although instances of LCPs appeared in publications dating back to the 1940s, their intensive study began in the mid 1960s. The name "linear complementarity problem" was coined by Cottle in 1965, to replace the earlier names including the "fundamental problem" and "complementary pivot problem". As in LP, earlier solution methods for LCPs were pivoting schemes, one of the most popular being Lemke's method [Lem65]. General iterative methods have been developed in parallel to the pivoting methods, and later interior-point methods for LPs were generalized to solve certain LCPs; see [KMNY91] and the references therein.

The horizontal linear complementarity problem was first introduced by Samelson, Thrall, and Wesler [STW58]. Their study originated from a problem in structural engineering. Other sources for HLCPs include electrical engineering and the modeling of static and dynamic systems. Recently, HLCP has received an increasing amount of attention due to its use as a unifying framework for the convergence analysis of infeasible-interior-point algorithms for certain classes of convex programming problems; see [Zha94].

It is easy to perform the reduction of HLCPs to standard LCPs when one of the input matrices N and M is nonsingular. An important observation that we make in Chapter 3 is that if one can form a nonsingular matrix by juxtaposing columns taken from N or M while maintaining the original relative ordering of the columns, then such a reduction can again be carried out by a change of variables. Explicit enumeration of matrices that can be generated by such a construction would then provide a technique for reduction. However, there are 2^n such matrices when N and M are n-dimensional and this exponential growth is not welcome considering that n may be quite large for practical problems.

Instead, we analyze the graphs induced by the sparsity structure of these matrices and observe that certain paths in these graphs correspond to certain exchange operations one can perform between the columns of the matrices N and M to obtain a column rearrangement that will facilitate a reduction of the problem to a standard LCP. We use this idea to develop a polynomial-time algorithm that finds a nonsingular matrix constructed by juxtaposing columns taken from N or M or proves that all such matrices are singular.

Particular properties of a square matrix determine the solution characteristics of the LCPs defined by it. For example, when the matrix M is positive semidefinite then the LCP defined by M and any vector q is solvable whenever it is feasible.

Such properties can be extended to the case of matrix pairs that define horizontal LCPs. A nice feature of our reduction algorithm is that it preserves these matrix properties. This leads to a characterization of the solutions of HLCPs defined by particular matrix pair classes.

In addition to studying these extensions, we try to determine which classes of matrix pairs lead to HLCPs that can be reduced to standard LCPs by means of the algorithm we develop. Our analysis shows that large class of matrix pairs, including P_0-*pairs*, define reducible HLCPs.

1.4 Implementing an Interior-Point Method for SDPs

The ultimate reason for the theoretical development and analysis of algorithms for particular classes of mathematical problems is to use them to solve real problems in these classes. By this token, the *implementation* stage follows the theoretical development. However, it is essential to understand that this is a circular process, i.e., implementation details lead to new problems that require additional analysis and vice versa.

In Chapter 4, we consider an interior-point algorithm developed to solve semidefinite programming problems and discuss how it can be implemented efficiently. As we mentioned above semidefinite programming problems are in a sense similar to LPs and especially so when one tries to solve them using interior-point methods. This similarity motivates us to consider one of the most efficient implementations of interior-point methods for LPs, namely the Mehrotra-type predictor-corrector method, and to use this approach in solving SDPs.

Semidefinite programming problems have a shorter history than LPs but two important developments in recent years attracted tremendous interest in their study and resulted in a vast literature on the subject. One of these developments is the successful extension of interior-point methods from linear programming to semidefinite programming by Nesterov and Nemirovsky [NN90,NN94] and Alizadeh [Ali91,Ali95]. Indeed, for a given closed convex domain that permits a *self-concordant barrier function*, one can associate an interior-point method with this barrier to minimize a linear objective function in polynomial time, see [NN90]. Therefore, polynomial-time IPMs for LP can be extended to certain nonlinear, nonsmooth convex programming problems without difficulty. SDPs fall into this category.

The other important development is the improved approximation results obtained for certain hard combinatorial optimization problems using their semidefinite relaxations, see [GW95]. Although the semidefinite relaxation technique is not new for the solution of combinatorial optimization problems, Goemans and Williamson's novel analysis and attractive results drew attention to the field.

One of the earlier papers on SDPs is by Bellman and Fan [BF63]. Most of the initial algorithmic work related to SDPs focused on the minimization of the maximum eigenvalue of a symmetric matrix (which can be formulated as a semidefinite program, see, e.g., [VB96]). Introduction of interior-point methods for general SDPs gave a boost to the field which has been evolving rapidly since then.

The algorithm we consider was developed by Nesterov and Todd [NT94,NT95]. What their development lacks is a framework to implement this method using the Mehrotra-type predictor-corrector approach. To be more precise, Mehrotra's approach is based on a higher order implementation of Newton's method for solving a nonlinear system and Nesterov and Todd's method has a very different motivation.

Our analysis shows how the Nesterov-Todd method can be seen as a Newton-based approach by considering a reparametrization of the problem which in turn facilitates the development of technical tools for a Mehrotra-type implementation of their method. As a by-product of our analysis we generate some uniqueness results for certain linear systems that arise in the solution of SDPs using interior-point methods. We note that our study has led to an efficient and robust implementation of the Nesterov-Todd approach results of which are documented in [TTT96].