McMaster University

Advanced Optimization Laboratory

Title:

Self-Regularity:
A New Paradigm for Primal-Dual Interior-Point Algorithms

This is a sample chapter of the mononograph.

Author:

Jiming Peng, Cornelis Roos and Tamás Terlaky

AdvOl-Report No. 2000/7

This is a sample chapter of the mononograph.

December 2000, Hamilton, Ontario, Canada

Self-Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms

Jiming Peng[† ‡] Cornelis Roos[†] Tamás Terlaky[‡]

December 2000

[†] *Faculty of Information Technology and Systems,*
Delft University of Technology
P.O.Box 5031, 2600 GA Delft, The Netherlands.
Email: C.Roos@its.tudelft.nl, J.Peng@its.tudelft.nl.

[‡] *Department of Computing and Software,*
McMaster University, Hamilton, Ontario, Canada, L8S 4L7.
Email: terlaky@cas.mcmaster.ca, pengj@cas.mcmaster.ca.

Preface

The primary goal of this monograph is to introduce a new framework for the theory of primal-dual interior point methods (IPMs) based on the notion of the self-regularity of a function. Starting from Karmarkar's epoch-making paper [**?**] in 1984, the research on IPMs has dominated the field of continuous optimization for more than 15 years, and over 3000 papers have been published related to IPMs. With the efforts of many excellent experts in the field, particularly the path-breaking work by Nesterov and Nemirovskii [**?**], the theory of IPMs has been developed into a rather mature principle.

An important concept in the IPM literature is the central path of the underlying problem. This was first recognized by Sonnevend [**?**] and Megiddo [**?**]. The primal-dual framework presented in [**?**] laid down the bedrock of many practically efficient IPMs with interesting theoretical properties. These IPMs utilize various strategies to follow the central path approximately. Two contrasting approaches in the analysis and implementation of IPMs are the so-called small-update (with small neighborhood) and large-update (with large neighborhood) methods. Unfortunately, up to now there is still a big gap between the theory and the practical performance of IPMs with respect to these two strategies. As stated by Renegar [**?**, pp. 52], "It is one of the ironies of the IPM literature that algorithms which are more efficient in practice often have somewhat-worse complexity bounds."

The motivation for this work to try and bridge this gap. We deal with linear optimization, nonlinear complementarity problems, semidefinite optimization and second-order conic optimization problems. Our framework also covers large class of linear complementarity problems and convex optimization. The algorithm considered may be interpreted as a path-following method or a constrained potential reduction method. Starting from a primal-dual strictly feasible point, our algorithm chooses a search direction defined by some Newton-type system derived from the self-regular proximity. The iterate is then updated, with the iterates staying in a certain neighborhood of the central path until an approximate solution to the problem is found. By exploring extensively some intriguing properties of the self-regular function, we establish that the complexity of large-update IPMs can come arbitrarily close to the best known iteration bounds of IPMs.

Acknowledgements: The research of the first two authors was mainly supported by the project *High Performance Methods for Mathematical Optimization*, under the Dutch SWON grant 613-304-200. They are grateful for much help from the Faculty of Information Technology and Systems, Delft University of Technology in the Netherlands. Most of this work was done when the first author was visiting the third author in the Department of Computing and Software, McMaster University, Canada. Support from that department is also acknowledged. Both the first and third authors were partially supported by the National Science and Engineering Research Council of Canada, grant 227650-00.

Contents

Chapter 1

Introduction and Preliminaries

Linear Programming is a revolutionary development that permits us, for the first time in our long evolutionary history, to make decisions about the complex world in which we live that can approximate, in some sense, the optimal or best decision.

George B. Dantzig [?]

A short survey about the fields of linear optimization[1] *and interior point methods is presented in this chapter. Based on the simple model of standard linear optimization problems, some basic concepts of interior point methods and various strategies used in the algorithm are introduced. The purpose of this work, as well as some intuitive observations that sparked the authors' research, are described. Several preliminary technical results are presented as a preparation for later analysis and the contents of part I are outlined as well.*

1.1 Historical Background of Interior Point Algorithm

1.1.1 Prelude

There is no doubt that the major breakthroughs in the field of mathematical programming are always inaugurated in linear optimization. Linear optimization, hereafter LO, deals with a simple mathematical model that exhibits a wonderful combination of two contrasting aspects: it can be considered as both a continuous and a combinatorial problem. The continuity of the problem: finding a global minimizer of a continuous linear function over a continuous convex polyhedral constrained set, and its combinatorial character: looking for optimality over a set of vertices of a polyhedron. The Simplex algorithm [?], invented by Dantzig in the mid 1940s, explicitly explores its combinatorial structure to identify the solution by moving

[1] For some historical reason, the name "linear programming" was coined in the early stage of the field of linear optimization. In this monograph we choose its more natural name "linear optimization". This also distinguishes the field from general programming works related to computers.

from one vertex to an adjacent one of the feasible set with improving values of the objective function. Dantzig's Simplex method has achieved tremendous success in both theory and application, and it remains to this day one of the efficient workhorses for solving LO.

The invention of the digital computer has brought dramatic changes to the world, and thus to the area of optimization. By combining computers with efficient methods, we can now solve many large and complex optimization problems that were unsolvable before the advent of computers, or even two decades ago. Inspired by the fast development of computers, scientists began to study the complexity theory of methods in the 1960s and 1970s. An algorithm was termed polynomial if the number of arithmetic operations taken by the algorithm to solve the problem, could be bounded above by a polynomial in the "size" of the problem.[2] Correspondingly such a polynomial algorithm was recognized as an efficient algorithm, and otherwise not. Although the Simplex method with certain pivot rules works very well in practice and it had been shown that the probabilistic computational complexity of the Simplex method is strongly polynomial (see Borgwardt [**?**]), it might take $2^n - 1$ iterations to solve the linear programming problem constructed by Klee and Minty [**?**] in 1972, where n is the number of variables in the problem with $2n$ constraints. Similar 'exponential examples' for different variants of the Simplex method were reported as well. Agitated by these cursed exponential examples for the Simplex method, the researchers in the field became interested in the issue of whether LO problems are solvable in polynomial time?

In 1979, an affirmative answer to the above open question was given by Khachiyan [**?**], who utilized the so-called ellipsoid method to solve the LO problem and proved that the algorithm has a polynomial $\mathcal{O}(n^2L)$ iteration complexity with a total of $\mathcal{O}(n^4L)$ bit operations. Khachiyan's results were immediately hailed in the international press, and the ellipsoid algorithm was subsequently studied intensively by various scholars in both theory and implementation. Unfortunately, in contrast to people's high expectation, even the best known implementation of the ellipsoid method is far from competitive with existing Simplex solvers. In this first challenge of polynomial algorithms to the Simplex method, the latter was the obvious winner. Nevertheless, the sharp contrast between the practical efficiency and the theoretical worse-case complexity of an algorithm set the stage for further exciting developments.

1.1.2 A Brief Review of Modern IPMs

The new era of IPMs started in 1984 when Karmarkar [**?**] proposed his LO algorithm, which enjoyed a polynomial complexity of $\mathcal{O}(nL)$ iterations with $\mathcal{O}(n^{3.5}L)$ bit operations, and made the announcement that his algorithm could solve large-scale linear programming problems much faster than the Simplex method. At that time the name of Karmarkar and his algorithm reached the front-page of New York Times despite the fact that his claim was received with much scepticism by some experts in the field. Nowadays, it is clear that Karmarkar opened a new field: The flourishing field of modern IPMs.

[2] The size of a problem (denoted by L) indicates the length of a binary coding of the input data for the problem. For the standard LO (1.1) on page 4, the length is $L = \sum_{i=0}^{m}\sum_{j=0}^{n}\lceil \log_2(|a_{ij}|+1)\rceil$ with $a_{i0} = b_i$ and $a_{0j} = c_j$. In this work, we derive upper bounds for the number of iterations to obtain an ε-optimal solution of the considered problem, where ε is an accuracy parameter used in the algorithm for termination.

Karmarkar's algorithm in its original form is a primal projective potential reduction method. The potential function was employed to measure the progress of the algorithm and force the iterates to stay in the feasible region. Although Karmarkar considered a nonstandard LO model, it soon turned out that the projective transformation in [?] is not necessary for solving standard LO problems. Shortly after the publication of [?], Gill et al. [?] observed that some simple variants of Karmarkar's algorithm could be tracked back to a very old algorithm in nonlinear optimization: *the logarithmic barrier method*. This observation led to a revival of some old methods for continuous optimization including *the logarithmic barrier function method* by Frisch [?, ?] and Fiacco and McCormick [?], *the center method* by Huard [?], and *the affine scaling method* by Dikin [?]. For example, it was proven by Roos and Vial [?] that the basic logarithmic barrier method for LO has a polynomial complexity. Most of the early work on IPMs in the 1980s followed Karmarkar's primal setting, but focused on more efficient implementation or better complexity bounds. A remarkable work at that time was the algorithm [?] by Renegar, who proposed to use upper bounds on the optimal value of the objective function to form successively smaller subsets of the feasible set, and employ Newton's method to follow the analytic centers of these subsets to get the primal optimal solution. Closely related to the center of a polyhedron is another very important concept in the IPM literature: the so-called *central path* first recognized by Sonnevend [?] and Megiddo [?]. Almost all known polynomial-time variants of IPMs use the *central path* as a guideline to the optimal set, and some variant of Newton's method to follow the central path approximately. These Newton-type methods fall into different groups with respect to the strategies used in the algorithms to follow the central path. One can consult the survey paper by Gonzaga [?] or the monograph of den Hertog [?] for further details.

Among versatile types of IPMs, the so-called primal-dual path-following methods have emerged to be particularly attractive and useful. The primal-dual setting was first suggested by Megiddo [?], Monteiro and Alder [?], Tanabe [?] and Kojima et al. [?] for LO problems, and later extensively investigated for complementarity problems by a Japanese group led by Kojima [?]. In the theoretical aspect, primal-dual IPMs possess the appealing theoretical polynomiality and allow transparent extension to other problems such as convex programming and complementarity problems. In practice they also form the basis of most efficient IPM solvers. A notable work with respect to the practical IPM is due to Mehrotra [?] who described a predictor-corrector algorithm in 1992. Mehrotra's scheme has been proved to be very effective in practice and is employed in most existing successful IPM solvers. Much effort went into investigating preprocessing techniques, warm start, sparse Cholesky factorization, and other implementation issues. By these efforts, the implementation of IPMs was enhanced greatly at both the commercial and the academic level. The paper [?] gives a thorough survey about the implementation and performance of IPMs. As claimed there, compared with the Simplex method, IPMs appear to be a strong rival for solving LO problems of medium size, and the winner for large-scale ones (see also [?]).

The theory of IPMs has been developed into a mature principle during the 1990s. One main contribution to IPM theory in that period came from two mathematicians Nesterov and Nemirovskii [?], who invented the theory of the by-now-well-known self-concordant functions, allowing the algorithms based on the logarithmic barrier function for LO to be transparently extended to more complex problems such as nonlinear convex programming, nonlinear complementarity problems, variational inequalities, and particularly to semidefinite optimization

(SDO) and second-order conic optimization (SOCO). Nesterov and Todd [?, ?] further generalized the primal-dual algorithms to LO on the so-called self-scaled cones, which still include SDO and SOCO as concrete instances. To some extent, SDO became recently the most active area of mathematical programming. SDO involves minimizing the values of a linear function with matrix argument subject to linear equality constraints and requiring that the matrix argument be positive semidefinite. The SDO paradigm includes as special cases LO, quadratic programming (QP), the linear complementarity problem (LCP), and SOCO. It has a great variety of applications in various areas such as optimal control, combinatorics, structural optimization, pattern recognition, etc. We refer to the excellent survey by Vandenberge and Boyd [?] for more details. An extremely important fact is that, while problems such as LO, QP and LCP can also be solved by other methods (and hence IPM is only an alternative choice), IPMs appear to be the first and also most efficient approach for SDO. As such, SDO makes a wonderful advertisement for the power of IPMs, although the theory and implementation of IPMs for SDO are still far from mature, especially the implementation. There is also a great deal of work studying other theoretical properties of IPMs such as local convergence of the algorithm, procedures to get an exact solution from an approximate solution and sensitivity analysis. Interested readers are referred to recent books ([?] by Roos, Terlaky and Vial, [?] by Wright and [?] by Ye) and the references therein. The book edited by Terlaky [?] collects some state-of-art review papers on various topics of IPMs and a large number of related references. For more recent achievements on IPMs we refer to the Interior-Point Methods Online Web site at *www.mcs.anl.gov/otc/InteriorPoint*, where most of the technical reports in the past five years are listed.

1.2 Primal-Dual Path-following Algorithm for Linear Programming

1.2.1 Primal-Dual Model for LO, Duality Theory and the Central Path

In this monograph we are mainly concerned with the complexity theory of IPMs. This is the major theme in the recent IPM literature. To be more specific we need to go into more detail at this stage. We start with the following standard linear programming problem

$$(LP) \qquad \min\{c^T x : Ax = b,\ x \geq 0\}, \tag{1.1}$$

where $A \in \Re^{m\times n}$, $b \in \Re^m$, $c \in \Re^n$, and its dual problem

$$(LD) \qquad \max\{b^T y : A^T y + s = c,\ s \geq 0\}. \tag{1.2}$$

Throughout this work, we assume that A has full row rank, i.e., $\mathrm{rank}(A) = m$. This implies that for a given $s \geq 0$, the vector y is uniquely defined. Hence we may identify a feasible solution of (LD) only by s. The following definitions define some basic concepts for LO.

Definition 1.2.1 *A point x is said to be feasible (or strictly feasible) for (LP) if $Ax = b$ and $x \geq 0$ (or $x > 0$). A point (y, s) is feasible (or strictly feasible) for (LD) if $A^T y + s = c$ and $s \geq 0$ (or $s > 0$). The point (x, y, s) is primal-dual feasible (or strictly feasible) if x and (y, s) are feasible (or strictly feasible) for (LP) and (LD), respectively.*

The relationships between (LP) and (LD) have been well explained by the duality theory of LO. For instance, if (x, y, s) is a primal-dual feasible pair, then there holds $b^T y \leq c^T x$. In other words, the objective value in (LD) gives a lower bound for the objective in (LP), and the objective in (LP) provides an upper bound for that in (LD). The main duality results can be summarized by the following strong duality theorem [**?**, **?**].

Theorem 1.2.2 *For (LP) and (LD) one of the following four alternatives holds:*

(i) (LP) and (LD) are feasible and there exists a primal-dual feasible pair (x^*, y^*, s^*) *such that*

$$c^T x^* = b^T y^*.$$

(ii) (LP) is infeasible and (LD) is unbounded.

(iii) (LD) is infeasible and (LP) is unbounded.

(iv) Both (LP) and (LD) are infeasible.

Hence, solving LO amounts to detecting which of these four cases holds, and in case (i) an optimal solution (x^*, y^*, s^*) must be found. Note that in case (i), the two objective values in (LP) and (LD) coincide with each other at the solution (x^*, y^*, s^*), that is $c^T x^* = b^T y^*$, which further implies

$$(s^*)^T x^* = (c - A^T y^*)^T x^* = c^T x^* - b^T y^* = 0.$$

Observe that since $x^*, s^* \geq 0$, the above equality can also be written as

$$x_i^* s_i^* = 0, \quad i = 1, \ldots, n.$$

An intrinsic property of LO is given by the following result.

Theorem 1.2.3 *Suppose that both (LP) and (LD) are feasible. Then there exists a primal-dual feasible pair* (x^*, y^*, s^*) *such that* $(x^*)^T s^* = 0$ *and* $x^* + s^* > 0$. *A solution* (x^*, s^*) *with this property is called strictly complementary.*

This theorem was first established by Goldman and Tucker [**?**] and later studied by other researchers via different approaches. It plays an important role in the design and analysis of IPMs, particularly in the procedures for getting an exact solution from an approximate solution obtained by IPMs, or detecting the infeasiblity of the problem [**?**, **?**].

Starting from an initial point (x^0, y^0, s^0) with $x^0, s^0 > 0$, all primal-dual interior point algorithms generate a point sequence (x^k, y^k, s^k), with $x^k, s^k > 0$, converging to the set of optimal solutions. If, at each iteration, the point (x^k, y^k, s^k) further satisfies the linear equality constraints, then we call the algorithm a *feasible interior point algorithm*, and otherwise an *infeasible interior point algorithm*. The choice between feasible and infeasible IPMs depends on whether a feasible starting point is available or not. If a strictly feasible point is known, then by starting from this point and carefully updating the iterates, we can keep the iterative sequence feasible.

In this work, the class of feasible IPMs is taken as the framework for our discussions. The reasons for this are twofold. First, theoretically it is more convenient to analyze a feasible IPM rather than an infeasible one. Needless to say, with some extra effort, the analysis for feasible IPMs can usually be extended to its infeasible counterpart. Second, for some problems such as LO, LCP, and SDO, by increasing slightly the size of the problem, we can always use the self-dual embedding model (see Chapter **??** or [**?**, **?**, **?**]) to reformulate the original problem as a new problem for which a strictly feasible starting point is readily available. Hence from now on we will assume without loss of generality that both (LP) and (LD) satisfy *the interior point condition*, i.e., there exists (x^0, s^0, y^0) such that

$$Ax^0 = b,\ x^0 > 0, \quad A^T y^0 + s^0 = c, \quad s^0 > 0. \tag{1.3}$$

Actually, by using the self-dual embedding model (Chapter **??**), we can further assume that $x^0 = s^0 = e$ and $\mu_0 = x^{0T} s^0/n = 1$. Under the interior point condition, we are in case (i) of Theorem 1.2.2, and hence an optimal solution pair always exists. From Theorem 1.2.2 one can see that finding an optimal solution of (LP) and (LD) is equivalent to solving the following system:

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= 0. \end{aligned} \tag{1.4}$$

Here xs denotes the coordinatewise product of the vectors x and s. The basic idea of primal-dual IPMs is to replace the third equation in (1.4), the so-called *complementarity condition* for (LP) and (LD) by the parameterized equation $xs = \mu e$, where e denotes the all-one vector and $\mu > 0$. Thus we consider the system

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ A^T y + s &= c, \quad s \geq 0, \\ xs &= \mu e. \end{aligned} \tag{1.5}$$

The existence of a unique solution to the above system is long known (see McLinden [**?**], Kojima et al. [**?**] and Güler [**?**]).

Theorem 1.2.4 *If the interior point condition (1.3) holds, then for each $\mu > 0$, the parameterized system (1.5) has a unique solution.*

Let us denote this solution by $(x(\mu), y(\mu), s(\mu))$. We call $x(\mu)$ the μ-center of (LP) and $(y(\mu), s(\mu))$ the μ-center of (LD). The set of μ-centers (with μ running through all positive real numbers) gives a homotopy path, which is called *the central path* of (LP) and (LD) (Sonnevend [**?**], Megiddo [**?**]). The limiting behavior of the central path as μ goes to zero has been a hot topic for some time. In [**?**], McLinden investigated the limiting behavior of the path for monotone complementarity problems. The properties of the central path for LO were first considered by Megiddo [**?**], and later by Güler and Ye [**?**]. In the sequel we discuss some of the main properties of the central path. To this end, we first introduce the definition of the analytical center of the optimal set for LO.

Definition 1.2.5 *Let*

$$\mathcal{F} = \{(x, s) : Ax = b,\ A^T y + c = s,\ x \geq 0,\ s \geq 0,\ x^T s = 0\}$$

denote the optimal solution set of primal-dual pair (LP) and (LD). Let

$$\begin{aligned} \mathcal{B}_{\mathcal{F}} &= \{i : \exists (x, s) \in \mathcal{F} \text{ such that } x_i > 0\}, \\ \mathcal{N}_{\mathcal{F}} &= \{i : \exists (x, s) \in \mathcal{F} \text{ such that } s_i > 0\}. \end{aligned}$$

The analytic center of $\mathcal{F}$ is given by

$$(x^*, s^*) = \text{argmax}_{(x,s)\in\mathcal{F}} \prod_{i\in\mathcal{B}_{\mathcal{F}}} x_i \prod_{j\in\mathcal{N}_{\mathcal{F}}} s_j.$$

It follows from Theorem 1.2.3 that the analytical center of $\mathcal{F}$ is a strict complementary solution for LO. A nice property of the central path is that it converges not only to the optimal set $\mathcal{F}$, but also to its analytical center [**?**].

Theorem 1.2.6 *The central path converges to the analytical center of the optimal set $\mathcal{F}$.*

1.2.2 Primal-Dual Newton Method for LO

All the primal-dual path-following algorithms trace the central path approximately. Let us briefly indicate how this goes. To outline a general procedure for IPMs, we first need to define the so-called *neighborhood of the central path*:

$$\mathcal{N}(\tau, \mu) = \{(x, s) > 0 : (x, s) \in \mathcal{F},\ \eta(x, s, \mu) \leq \tau\},$$

where $\eta(x, s, \mu)$ is a so-called *proximity* to measure the distance from the point (x, s) to $(x(\mu), s(\mu))$, and τ is the *radius of the neighborhood*. For the time being, we focus on the structure of the algorithm and leave the discussion of various choices of the proximity $\eta(x, s, \mu)$ to the next section. Most feasible IPMs take the following form.

General Feasible Primal-Dual Newton Method for LO

Input:
A proximity parameter τ;
an accuracy parameter $\varepsilon > 0$;
a fixed barrier update parameter θ, $0 < \theta < 1$;
(x^0, s^0) and $\mu^0 = 1$ such that $\eta(x^0 s^0, \mu^0) \leq \tau$.

begin
$x := x^0$; $s := s^0$; $\mu := \mu^0$;
while $n\mu \geq \varepsilon$ **do**
begin
(Outer Iteration) $\mu := (1-\theta)\mu$;
while $\eta(x, s, \mu) \geq \tau$ **do**
(Inner Iteration)
Solve system (1.6) for $\Delta x, \Delta y, \Delta s$;
begin
Determine step size α by some rules;
$x := x + \alpha \Delta x$;
$s := s + \alpha \Delta s$;
$y := y + \alpha \Delta y$;
Update μ by certain rules.
end
end
end

Remark 1.2.7 *To be consistent with the name used in [?], we refer to each update of μ as an outer iteration and the procedure for updating the primal-dual pair (x, y, s) as an inner iteration. In the algorithm, we use the proximity $\eta(x, s, \mu)$ to control the iterates. Various choices of these proximities are discussed in the next section.*

Without loss of generality we assume that a point (x, y, s) in the neighborhood $\mathcal{N}(\tau, \mu)$ is known for some positive μ. Then we update $\mu := (1-\theta)\mu$ and solve the following Newton system:

$$\begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs. \end{aligned} \tag{1.6}$$

Because A has full row rank, system (1.6) has a unique solution for any $(x, s) > 0$. Thus we get a search direction $(\Delta x, \Delta y, \Delta s)$. Note that if $\theta = 1$, the parameter μ in system (1.6) equals zero. In this case the solution of (1.6) is called the *affine scaling direction* or *predictor direction*. If $\theta = 0$, we obtain the so-called *corrector direction*. By taking a step along the search direction, where the step size α is defined by some line search rules, one constructs a new triple $(x, y, s) = (x_+, y_+, s_+)$. We then update μ by certain rules. If the new iterate

(x, y, s) is in the neighborhood $\mathcal{N}(\tau, \mu)$ of the new center $(x(\mu), y(\mu), s(\mu))$, then μ is reduced again by a factor $1 - \theta$ and we repeat the Newton process until the duality gap is sufficiently small. Otherwise, we need to solve system (1.6) again and get a corrector direction. By taking a suitable step along the corrector direction, we can get a new point that is 'closer' to the center $(x(\mu), y(\mu), s(\mu))$. This inner iteration will be repeated until the iterative point enters the neighborhood $\mathcal{N}(\tau, \mu)$ again. Then we go back to outer iteration, and so on. This process is repeated until an approximate solution to the problem is obtained. Most practical algorithms then construct an exact solution by resorting to a rounding procedure as described by Ye [**?**] (see also Mehrotra and Ye [**?**] or [**?**]), and possibly produce a basic solution via a basis identification procedure or or crossing-over to the Simplex method.

We proceed with a discussion about the update of the parameter μ and line search rules in IPMs. At the outset, it should be mentioned that the update of μ in an outer iteration affects only the search direction in the first inner iteration following the outer iteration. In most IPMs the parameter μ in the inner iteration is defined by $\mu = \frac{x^T s}{n}$ depending on the current point (x, y, s) and updated subsequently after each step. For some IPMs, the step size is chosen very carefully so that the new iteration will always stay in a certain neighborhood of the central path. Hence for such algorithms, only one inner iteration is needed between two successive outer iterations. A more practical way to deal with the step size α is that, in the first inner iteration just after an outer iteration, where usually a large θ is employed, we compute the maximal feasible step size ($\alpha_{\max}$) first and then use $\alpha_{\max}$ up to a certain ratio (for instance $0.9\alpha_{\max}$) as a step size. Then we can update μ and check whether the new point is in the neighborhood of the new center. If the answer is "yes", then we can go back to the outer iteration. Otherwise we commence the inner iteration, and utilize the corrector direction to get a new iterate that is closer to the new center. The inner process is repeated with fixed μ until the iterate enters the neighborhood $\mathcal{N}(\tau, \mu)$ again. Actually, the idea of utilizing a fixed factor of the maximal step size as the working step size has been widely employed in most IPM packages and works extremely well [**?**]. Note that (1.6) gives $\Delta x^T \Delta s = 0$, and whenever the corrector steps are applied,

$$x^T \Delta s + s^T \Delta s = n\mu - x^T s = 0 \quad \text{if} \quad \mu = \frac{x^T s}{n}.$$

The new duality gap is then $(x + \alpha\Delta x)^T (s + \alpha\Delta s) = x^T s$. Hence if $\mu = \frac{x^T s}{n}$ is used for the corrector directions, then the parameter μ remains invariant during the inner process. In such a situation, μ acts as an independent parameter. The above iterative process can therefore be interpreted in the following alternative way, namely the parameter μ is only updated by $\mu = (1 - \theta)\mu$ in the outer iteration and we do not update μ in the inner procedure. Thus, in the inner iteration only the corrector direction (or equivalently the Newton direction for the system $xs = \mu e$) is employed to force the new iterative point to move closer to the target: the new center. This model has been employed for most algorithms in the book by Roos, Terlaky and Vial [**?**]. In this work, we choose this model as the subject for our book. The reason for this is, in the authors' opinion, such a model is very convenient for the complexity analysis of the algorithm. Also, the analysis for such a model where μ is cast as a free parameter can always be adapted to handle the model where μ is defined via the current iterate $\mu = \frac{x^T s}{n}$ without much difficulty. Lastly we would like to mention that the decision on whether μ is treated as an independent parameter in the inner iteration does not have much effect on the practical performance of IPMs .

Since we do not update μ in the inner iteration, to avoid some unnecessary confusing arguments, we describe the classical primal-dual Newton method as follows.

Classical Primal-Dual Newton Method for LO

Input:
A proximity parameter τ;
an accuracy parameter $\varepsilon > 0$;
a fixed barrier update parameter θ, $0 < \theta < 1$;
(x^0, s^0) and $\mu^0 = 1$ such that $\eta(x^0 s^0, \mu^0) \leq \tau$.

begin
$x := x^0$; $s := s^0$; $\mu := \mu^0$;
while $n\mu \geq \varepsilon$ **do**
begin
$\mu := (1 - \theta)\mu$;
while $\eta(x, s, \mu) \geq \tau$ **do**
Solve system (1.6) for $\Delta x, \Delta y, \Delta s$;
begin
Determine a step size α;
$x := x + \alpha \Delta x$;
$s := s + \alpha \Delta s$;
$y := y + \alpha \Delta y$
end
end
end

1.2.3 Strategies in Path-following Algorithm and Motivation

It is clear from the above description that to continue the iteration process, we want to know when a point (x, y, s) is 'close' enough to the central path, so that we can update the parameter μ and go on to the next outer iteration. In the classical primal-dual Newton method, this is done by means of two arguments: the proximity $\eta(x, s, \mu)$ and the constant τ.

Two popular choices for η in the IPM field are the Euclidean norm $\left\| \frac{xs}{\mu} - e \right\|$ and the infinity norm $\left\| \frac{xs}{\mu} - e \right\|_\infty$, while τ is always chosen as a small constant such as $\frac{1}{2}$ [**?**, **?**, **?**]. In the IPM literature, the neighborhood defined by the infinity norm is usually referred to as a large neighborhood, and a small neighborhood if the Euclidean norm is applied.

The choice of the proximity measure is crucial not only for the quality and elegance of the analysis, but also for the performance of the algorithm. An example is that, in practice IPMs with large neighborhood are always more efficient than IPMs with small neighborhood. In sharp contrast, IPMs working with small neighborhood have better worst-case complexity

iteration bounds than IPMs with large neighborhood.

Besides the above-mentioned two proximities, several other proximities have been introduced and used in the IPM literature [?, ?, ?]. These proximities share some desirable properties that are very helpful in the analysis of IPMs. For simplicity, let us first introduce some notation. For any strictly feasible primal-dual pair (x, s) and any positive number μ, we define

$$v := \sqrt{\frac{xs}{\mu}}, \quad v^{-1} := \sqrt{\frac{\mu}{xs}} \tag{1.7}$$

to be the vectors whose i^{th} components are $\sqrt{\frac{x_i s_i}{\mu}}$ and $\sqrt{\frac{\mu}{x_i s_i}}$ respectively. This notation is widely used in the IPM literature to ease the description and analysis of IPMs .

The following are two popular proximity measures for primal-dual IPMs [?, ?, ?]:

$$\delta(x, s, \mu) := \frac{1}{\sqrt{2}} \left\| v - v^{-1} \right\|, \tag{1.8}$$

$$\Phi(x, s, \mu) := \sum_{i=1}^{n} \phi\left(v_i^2\right), \tag{1.9}$$

where

$$\phi(t) = t - 1 - \log t.$$

It is easy to see that each measure vanishes if $v = e$. The last two proximities go to infinity if v approaches the boundary of the nonnegative orthant. The latter case is characterized as the *barrier property* of the proximity. The measure Φ is closely related to the *logarithmic barrier function* with the *barrier parameter* μ; its usefulness has been known already for a long time (cf. Frisch [?], Lootsma [?] and Fiacco and McCormick [?]). The measure δ, up to a factor $\frac{\sqrt{2}}{2}$, was introduced by Jansen et al. [?] and thoroughly used in [?, ?, ?, ?]. Its SDO analogue was also used in the analysis of interior point methods for semidefinite optimization [?]. We note that variants of the proximity $\delta(x, s, \mu)$ were used by Kojima et al. in [?] and Mizuno et al. in [?]. There are also many others proximities used in the IPM literature [?, ?, ?]. Usually these proximities are more or less related to a special class of functions: the so-called *self-concordant* functions introduced by Nesterov and Nemirovskii [?].

Another important ingredient of IPMs is the choice of the parameter θ. Usually, if θ is a constant independent of n the dimension of the problem, for instance $\theta = \frac{1}{2}$, then we call the algorithm a large-update (or long-step) method. If θ depends on the problem such as $\theta = \frac{1}{2\sqrt{n}}$, then the algorithm is named a small-update (or short-step) method. As we stated in the preface, at present there is still a gap between the practical performance and the theoretical worse-case complexity of these two classes of IPMs in the literature. This is especially true for primal-dual large-update methods, which are the most efficient methods in practice (see, e.g. Andersen et al. [?]). In what follows we explain this phenomena in more detail.

We begin by considering a specific small-update IPM with the parameters $\theta = \frac{1}{2\sqrt{n}}$ and $\tau = 1$. Assume that the starting point lies in the neighborhood of the central path $\mathcal{N}(\mu, \tau) = \{(x, s) : \delta(x, s, \mu) \leq \tau\}$. For simplicity, we denote by δ and δ_+ the proximity at the

present iterate and after one update, respectively. We start with an inner iteration. It has been shown (see, for instance, Chapter 7 in [?]) that in this special situation, the full Newton step is feasible and after such a full Newton step, the proximity satisfies $\delta_+ \leq \frac{1}{2}\delta^2 \leq \frac{1}{2}$. Hence we can update μ and check whether the point is in the neighborhood of the new center. Note that, after the update of μ, the proximity δ might increase. However, it has been proven [?] that, if $\delta \leq \frac{1}{2}$ and $\theta = \frac{1}{2\sqrt{n}}$, then the increased proximity still satisfies $\delta_+ \leq 1$, and hence it stays within $\mathcal{N}(\mu, \tau)$. Thus we can dispense with the inner iteration and move to the next outer iteration. Observe that in this process, we reduce the parameter μ by a fixed ratio. It follows immediately if k is a constant such that $(1-\theta)^k \mu_0 \leq \varepsilon$, then the algorithm will terminate and report an ε-solution to the problem. By using some simple calculus, one can show that $k \leq 2\sqrt{n} \log \frac{n\mu_0}{\varepsilon}$ for sufficiently large n (Theorem II.52 in [?]) and this leads to the conclusion that the small-update method has an iteration bound $\mathcal{O}(\sqrt{n} \log \frac{n}{\varepsilon})$, which is to date the best known complexity result for IPMs.

Now let us indicate how a typical large-update method works, for instance $\theta = 0.9$ and $\tau = \sqrt{n}$. Suppose that the current point is in the neighborhood $\mathcal{N}(\mu, \tau)$. Then we can set $\mu = 0.1\mu$. Due to this change of μ, the square of the proximity δ^2 (which can be cast as a potential function) might increase to as large as $\mathcal{O}(n)$. For example, if the current iterate is on the central path, then $v = e$ and $\delta^2 = 0$. However, after the update of μ the new proximity becomes $\delta_+^2 = \frac{1}{2}\sum_{i=1}^n \left(\frac{1}{\sqrt{0.1}} - \sqrt{0.1}\right)^2 = 4.05n$. In this case, the primal-dual pair (x, s) goes far outside the neighborhood $\mathcal{N}(\mu, \tau)$. Thus we need to initialize the inner procedure to reduce the proximity. It has been proven by several researchers [?, ?] that for a suitable step size, $\delta_+^2 - \delta^2 \leq -\beta$, where β is a specific constant. This result implies that the number of inner iterations might be as large as $\mathcal{O}(n)$ to get recentered. Keep in mind that at each outer iteration, we reduce μ by a factor 0.1; hence the number of total outer iterations is bounded above by any constant k satisfying $0.1^k \mu_0 \leq \varepsilon$. It follows immediately that the number of outer iterations is less than $\mathcal{O}\left(\log \frac{n\mu_0}{\varepsilon}\right)$. Multiplying this number by our estimate of the number of inner iterations during two successive outer iterations, we find that the large-update IPM has an $\mathcal{O}(n \log \frac{n}{\varepsilon})$ iteration bound [?, ?, ?]. This result is worse than that for small-update IPMs. However, as demonstrated by Figures 1.1 and 1.2, large-update IPMs perform much more efficiently in practice than small-update methods [?]. This is the gap that Renegar [?] pointed out as one irony of IPM algorithms.

To close this gap, several remedies have been proposed by various authors aiming at interpreting the practical performance of interior-point algorithms or improving the complexity of large update IPMs. For example, instead of the worst-case analysis, Todd [?] and Ye [?] argued that the average-case iteration bounds of large-update IPMs for LO problem instances based on certain probability distribution is $\mathcal{O}(L \log n)$, while small-update IPMs have an average $\mathcal{O}(n^{1/4}L)$ iteration bound. Another attempt to improve the complexity of large-update IPMs was first considered by Monteiro, Adler and Resende, who utilized so-called higher-order methods to improve the complexity of large-update IPMs for linear and quadratic optimization problems. Their ideas were further extended and studied for variants of IPMs for solving more complex problems by Hung and Ye [?], Jansen et al. [?], Zhang and Zhang [?]. For IPMs with higher-order correctors, some additional equation systems that provide a higher-order approximation to the system (1.5) must always be solved at each iteration. Unfortunately neither of these two approaches are very satisfying. Having

Figure 1.1: Performance of a large-update IPM.

Figures 1.1 and 1.2 exhibit the practical performance of an IPM with large-update and small-update for a specific two-dimensional LO problem. These figures are drawn in the v-space.

Figure 1.2: Performance of a small-update IPM.

no other reasonable explanations, most experts in the IPM field then accept unwillingly a vague assertion that small-update IPMs are confined to unacceptably slow progress, while the large-update IPMs are much more adventurous and might have the potential for faster progress in practice.

A natural question arising here is: **Can we improve the complexity bounds of large-update methods by improved analysis without solving any additional systems?**

1.3 Preliminaries and Scope of the work

1.3.1 Preliminary Technical Results

In the last part of this introductory chapter, we first review some of our previous technical results and then describe some intuitive observations that serve to spark the research of the authors. We begin by presenting readers with some mathematical techniques. This is because these technical results are not only completely independent of the algorithm, but also essential in most of the analysis that we carry out for our new IPMs in the later chapters. Further, from the authors' viewpoint, these techniques might also take a key position in future development of the complexity theory of IPMs.

Let us start with a small exercise. Assume that one has a big number, say t_0, which in practice might mean some material. We consider the decreasing behavior of a sequence starting from this number, or in other words the reduction of the material, since the material will be reduced whenever it is used. The question is: when will the material run out? The answer is instinctive if the material is used at a constant frequency with a fixed amount each time. However, this is not the customary way for most people, since whenever the material is abundant, people are likely to lavish more than necessary until they realize a shortage of the material is imminent. Hence a more convincing way to predict the reduction of the material is to assume that it reduces at a speed that depends on the current situation of the material.

The mathematical model for the above situation can be phrased as follows: starting from a positive number t_0, we decrease it to $t_{k+1} = t_k - f(t_k)$, where f is a specific function. The question is: after how many steps will we obtain $t_k \leq 0$? This simple model resembles the inner iterative process in IPMs. Observe that the total iterations of a primal-dual Newton-type algorithm described in the previous section is composed of just two parts: the number of outer and inner iterations. Since we reduce the parameter μ by a constant factor ($\mu_+ = (1-\theta)\mu$) at each outer iteration, the number of outer iterations follows readily. The main effort in the study of IPMs is to figure out an answer to the following question: how many inner iterations are needed to recenter? Note that during the inner process, we always use corrector steps aimed at minimizing certain proximities. It will be desirable if we can reduce the proximity according to its current value. In other words, if the point is far away from the central path, then we hope to move to the path quickly, depending on the present position of the point. Unfortunately, this is not an easy task. For large-update IPMs, which are clearly the best workhorses in IPM implementations [**?**], most of the known results, as we described in Section 1.2.3, show only that the proximity (or the potential function) has at least a constant decrease in each inner iteration, resulting in the worse $\mathcal{O}(nL)$ iteration bounds of

the algorithms [**?**, **?**]. Obviously if we can prove that the decrease of the proximity after one step has a value associated with the present iterate, for instance $\delta_+^2 \leq \delta^2 - \beta\delta^\gamma$ for some $\beta, \gamma > 0$ (which means when the proximity δ^2 is large, it decreases at a rate much greater than just only a constant decrease), then the complexity of large-update algorithms might be improved. Note that in such a situation, the decreasing behavior of the proximity can be simply described by a specific case of the positive decreasing sequence $t_{k+1} = t_k - f(t_k)$ with $f(t) = \beta t^\gamma$.

Let us consider a concrete positive sequence, say $t_0 = 10000$ and $t_{k+1} = t_k - t_k{}^{\frac{1}{2}}$. The problem to be addressed is: after how many iterations do we get $t_k \leq 1$? The behavior of this sequence is demonstrated by Figure 1.3 where the straight line shows the decrease of the sequence defined by $t_{k+1} = t_k - 20$ and another curve illustrates the decreasing behavior of the sequence given by $t_{k+1} = t_k - 2t_k^{\frac{2}{5}}$.

Figure 1.3: The decreasing behavior of three positive sequences.

By a simple program, we quickly find an answer to the above question: after 196 steps one gets $t_{197} \leq 1$ for the sequence generated by $t_{k+1} = t_k - t_k^{\frac{1}{2}}$. We also observe that the sequence given by $t_{k+1} - 2t_k^{\frac{2}{5}}$ also reduces to a very small number after 200 steps, while the straight line is still above the horizontal line $t = 6000$. In what follows we give a mathematical explanation pertinent to the decreasing behavior of these positive sequences. First we present a technical result.

Lemma 1.3.1 *Suppose that* $\alpha \in [0, 1]$. *Then*

$$(1+t)^\alpha \leq 1 + \alpha t, \qquad \forall\ t \geq 0; \tag{1.10}$$

$$(1-t)^\alpha \ \leq 1-\alpha t, \qquad \forall\ t \in [0,1]. \tag{1.11}$$

Proof: Inequality (1.10) is true because for any fixed $\alpha \leq 1$, the function $(1+t)^\alpha - 1 - \alpha t$ is decreasing for $t \geq 0$ and zero if $t = 0$. Similarly we observe that for any fixed $\alpha \leq 1$, the function $(1-t)^\alpha - 1 + \alpha t$ is decreasing with respect to $t \in [0,1]$ and zero if $t = 0$. This gives (1.11). □

Now we are ready to state our main result in this section. This result plays a fundamental role in our complexity analysis of IPMs in the later chapters.

Proposition 1.3.2 *Suppose that $t_0 > 0$ is a constant. Suppose $\{t_k > 0, k = 0, 1, 2, \ldots, \bar{k}\}$ is a sequence satisfying the inequalities*

$$t_{k+1} \leq t_k - \beta t_k^\gamma, \quad k = 0, 1, \ldots, \bar{k} \tag{1.12}$$

with $\gamma \in [0, 1)$ and $t_{\bar{k}+1} < 0$. Then

$$\bar{k} \leq \left\lceil \frac{t_0^{1-\gamma}}{\beta(1-\gamma)} \right\rceil$$

and for any fixed $\rho \geq 0$,

$$t_{k+1} \leq \rho, \quad \textit{for all} \quad k \geq \left\lceil \frac{t_0^{1-\gamma} - \rho^{1-\gamma}}{\beta(1-\gamma)} \right\rceil.$$

Proof: First we note if $\beta \geq t_0^{1-\gamma}$, then one step is sufficient. Hence we can assume without loss of generality that $0 < \beta < t_0^{1-\gamma}$. Let us further assume that at the present step (k-th step) there holds $0 < \beta < t_k^{1-\gamma}$. Then from (1.12) we have

$$t_{k+1}^{1-\gamma} \leq (t_k - \beta t_k^\gamma)^{1-\gamma} = t_k^{1-\gamma}\left(1 - \beta t_k^{\gamma-1}\right)^{1-\gamma} \leq t_k^{1-\gamma}\left(1 - \beta(1-\gamma)t_k^{\gamma-1}\right) = t_k^{1-\gamma} - \beta(1-\gamma),$$

where the second inequality follows from (1.11). The proposition follows immediately. □

We mention that for the preceding example, the theoretical bound on the steps given by Proposition 1.3.2 is 199, which is quite close to the practical 196 steps. This indicates, to some extent, that our theoretical bound is very tight.

We close this section by presenting a technical lemma about the minimal values of certain specific convex functions.

Lemma 1.3.3 *Suppose that $h(t)$ is a twice differentiable convex function with*

$$h(0) = 0, \quad h'(0) < 0.$$

Suppose that $h(t)$ attains its global minimum at its stationary point $t^* > 0$ and $h''(t)$ is increasing with respect to t. Then for any $t \in [0, t^*]$,

$$h(t) \leq \frac{h'(0)t}{2}.$$

Proof: Since $h(0) = 0$, we have

$$\begin{aligned} h(t) &= \int_0^t h'(\xi)d\xi = h'(0)t + \int_0^t \int_0^\xi h''(\zeta)d\zeta d\xi \le h'(0)t + \int_0^t \xi h''(\xi)d\xi \\ &= h'(0)t + (\xi h'(\xi))|_0^t - \int_0^t h'(\xi)d\xi \le h'(0)t - h(t), \end{aligned}$$

where the first inequality is given by the assumption that $h''(t)$ is nonnegative and increasing with respect to $t > 0$, and the second inequality holds because $h'(t) \le 0$ for all $t \in [0, t^*]$. The lemma follows directly. □

Lemma 1.3.3 is elementary in estimating the decrease of our new proximity functions. It is worth noting that the decreasing behavior of some concrete forms of the function $h(t)$, such as $h(t) = t - 1 - \log t$ with $t > 0$, have been well studied in the IPM literature (see [**?**, **?**]).

1.3.2 Observations on the Relations between Proximity and Search Direction

We proceed with a short review of some ideas and observations that inspired our research. Departing from the probabilistic and higher-order approaches, the authors recently proposed [**?**, **?**] a new class of large-update IPMs for solving LO and SDO. To follow the central path more efficiently, we introduced some new search directions that were derived from a new class of proximities for the problem. To describe the algorithms in [**?**, **?**] more concretely, let us introduce the following notation:

$$d_x := \frac{v\Delta x}{x}, \quad d_s := \frac{v\Delta s}{s}; \tag{1.13}$$

$$\tag{1.14}$$

With this notation, one can state the centrality condition in (1.5) as $v = v^{-1} = e$. Let us further denote by $\bar{A} = \frac{1}{\mu}AV^{-1}X, V = \text{diag}\,(v), X = \text{diag}\,(x)$. Then we can rewrite system (1.6) in the v-space as

$$\begin{aligned} \bar{A}d_x &= 0, \\ \bar{A}^T\Delta y + d_s &= 0, \\ d_x + d_s &= v^{-1} - v. \end{aligned} \tag{1.15}$$

Set $d_v = d_x + d_s = v^{-1} - v$. From (1.15) one easily see that d_x, d_s is the orthogonal decomposition of the vector d_v in the null space and row space of $\bar{A}$ respectively. Another interesting observation is that we can always decompose the above system into two systems. One is the system defining the predictor direction, which is obtained by replacing the last term in (1.15) by

$$(d_v)_{Pred} = -v,$$

and the other is the system giving the centering direction by

$$(d_v)_{Cent} = v^{-1}.$$

The centering direction serves the purpose of centering (it points towards the "analytic center" of the feasible set), while the predictor direction aims to decrease the duality gap. It is straightforward to verify that $(d_v)_i \leq 0$ for all the components $v_i \geq 1$ and $(d_v)_i > 0$ for the components $v_i < 1$. This means that if $v_i < 1$ then the classical Newton step increases v_i and decreases it whenever $v_i > 1$ to get closer to the μ-center. It is reasonable to expect that if we can increase the small components and decrease the large components of v more, we might approach our target the μ-center faster, and hence follow the central path more efficiently while staying in a large neighborhood of the path. This straightforward idea is with no doubt the initial point where we launch our novel IPM approach for handling classes of problems.

The new search direction, rooted in the above simple idea and introduced in [**?**], is a slight modification of the standard Newton direction. It is defined by a new system as follows:

$$\begin{aligned} \bar{A}d_x &= 0, \\ \bar{A}^T\Delta y + d_s &= 0, \qquad (1.16) \\ d_x + d_s &= v^{-q} - v, \qquad (1.17) \end{aligned}$$

where $q \geq 1$ is a parameter. That is, instead of the classical centering direction $(d_v)_{Cent} = v^{-1}$, we employ a new kind of centering direction such as $(d_v)_{Cent} = v^{-q}$. At first glance, such a system seems to be out of the way. To extract a relatively deeper and mathematically logical interpretation, let us dwell on this system a little more. For instance, consider the special case that $q = 3$. It is clear that the right-hand side of (1.17) represents the negative gradient of the proximity measure δ^2 in the v-space. When solving this system, we get the steepest descent direction for the proximity measure δ^2, along which the proximity can be driven to zero. It is also of interest to note that the right-hand side, $v - v^{-1}$ of the last term in (1.15) is the negative gradient of the scaled classical barrier function $\frac{1}{2}\Phi$, which means the standard primal-dual Newton method is identical to the steepest descent method for minimizing Φ in the scaled v-space. With a deeper look back at the IPM literature, one can find that, as a matter of fact, most potential reduction methods for solving LO utilize the gradient of the potential barrier function to define a search direction. For instance, the primal-dual potential function in [**?**] (see Chapter 4.3) can be rewritten equivalently as

$$\Phi_1(xs, \mu) = (n + q_0) \log \|v\|^2 - \sum_{i=1}^{n} \log v_i^2, \quad q_0 \geq \sqrt{n},$$

while the search direction satisfies

$$d_x + d_s = v^{-1} - \frac{n + q_0}{\|v\|^2} v.$$

It is straightforward to verify that the search direction considered by Ye is proportional to those defined by (1.15) with v replaced by $\frac{\sqrt{n+q_0}}{\|v\|} v$, which is similar to an update of μ.

From the aforementioned observations, we were naturally led to the idea that whenever another proximity measure is used in the algorithm, one should adapt the search direction correspondingly. The algorithms posed in [**?**] follow such a schedule: they use a special class of potential functions (or proximities) in the path-following method to control the 'distance' of the point to the central path, and then the search direction is obtained by solving system (1.6),

where the right-hand side of (1.17) is represented by the gradient of the potential function in the scaled v-space. Such an idea can be viewed as an elegant combination of the path-following method and the potential reduction method. The functions introduced in [**?**] are named *self-regular* functions, a name inherited from the *self-concordant* function introduced by Nesterov and Nemirovskii [**?**]. As shown in [**?**], *self-regular* functions enjoy certain attractive properties not shared by *self-concordant* functions. Based on versatile appealing properties of *self-regular* functions, we developed some new IPMs with large update for solving LO and SDO, and showed that the complexity of these new algorithms can get arbitrarily close to the best known iteration bounds of IPMs.

1.3.3 Contents and Notational Abbreviations

This work aims at setting up a unified framework for the complexity theory of a class of primal-dual path-following methods for various problems including linear optimization (LO), second-order cone optimization (SOCO), semidefinite optimization (SDO), and nonlinear complementarity problems (NCPs). The work consists of eight chapters. In Chapter 2, we first introduce the definition of *self-regular* functions in $\Re_{++}$. Then we investigate several important properties of self-regular functions such as the growth behavior, the barrier behavior and the relations among the function and its first and second derivatives. Several generating rules to construct a self-regular function are treated as well. Common features and differences between *self-regular* functions and the well-known *self-concordant* functions are discussed. The content of this chapter consists of several conclusions from [**?**, **?**] but we refine and reorganize them slightly differently here. We also postpone the introduction of *self-regular* functions in semidefinite and second-order cones to the corresponding chapters where the considered problem is also described in more detail. It may be possible to build up a unified framework for *self-regular* functions depending on some abstract mathematical theory, but we think that first a simple and concrete introduction and then its more complex evolution will be helpful for most people to understand the concept of *self-regularity* more easily. We also expect such an arrangement will benefit readers who are interested in just one part of this work, the case of LO.

In Chapter 3, we first present the notion of *self-regular proximity* for LO, which is derived from the self-regular function. Self-regular Proximity for LO is defined in the so-called v-space, which we described early in Section 1.2.3. The growth behavior of the proximity as well as its barrier property are studied[3]. Then we introduce some new search directions for path-following algorithms for LO. Using the properties of the proximity, we prove that the large-update path-following method for LO has a polynomial $\mathcal{O}\left(n^{\frac{q+1}{2q}}\log\frac{n}{\varepsilon}\right)$ iteration bound, where $q \geq 1$ is the so-called *barrier degree* of the proximity. When q increases, our result approaches the best known complexity $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\varepsilon}\right)$ iteration bound for interior point methods. Our unified analysis also provides the $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\varepsilon}\right)$ complexity for small-update IPMs. At each iteration, we need to solve only one linear system. The possibility of relaxing some conditions is explored as well.

[3]Throughout this monograph, the growth behavior of a function $f(t)$ is the change of the function as its variable t goes to infinity. If the value of $f(t)$ tends to infinity as t reduces to zero, then we say $f(t)$ has a barrier property. The growth degree and barrier degree of a function are defined in Chapter **??**.

Chapter 4 depicts a direct extension of the algorithms posed in the preceding Chapter 3. The class of complementarity problems (CPs) is chosen as our underlying problems. Many problems in mathematical programming can be reformulated as CPs. To mention a few of them, one can list problems such as LO, QP and general convex programming problems that satisfy certain constraint qualifications. IPMs for solving CPs try to find a primal-dual pair (x, s) in the nonnegative orthant such that both a nonlinear equation[4] and the complementarity condition $xs = 0$ are satisfied. Because of the appearance of nonlinearity, only a specific class of CPs, which always require certain sufficient smoothness conditions (for example the scaled Lipschitz condition, or the self-concordant condition), can be solved efficiently by IPMs. In this chapter, some new sufficient condition are introduced. We show that, if the considered problem satisfies our new sufficient condition, then our new primal-dual path-following method has the same polynomial iteration bounds as its LO analogue.

Having built up the new framework of primal-dual IPMs for LO and NCPs, in Chapter 5 we discuss the generalization of such algorithms to SDO, where the complementarity condition is expressed by $XS = 0$ with both X and S in the positive semidefinite cone $\mathcal{S}_+^{n\times n}$. An important feature for SDO is that, to make the Newton system solvable in the space of $n \times n$ symmetric matrices $\mathcal{S}^{n\times n}$, one has to apply some scaling scheme so that the rescaled Newton system has necessarily a unique symmetric solution. For this we first discuss the proximities based on different scaling techniques. After exploiting some fascinating features of self-regular functions on the positive definite cone $\mathcal{S}_{++}^{n\times n}$, we show that, among others, the NT-scaling is the unique optimal choice if the corresponding proximity is *self-regular*. New search directions for SDO are then proposed rooted in the NT-scaling and *self-regular* proximities in $\mathcal{S}_{++}^{n\times n}$. Certain functions on the cone of general symmetric matrices are discussed and their first and second derivatives are estimated. To the authors' best knowledge, this is the first time that estimates have been obtained for the derivatives of a general function involving matrix functions. The results are meaningful also in a pure mathematical sense. They might be very helpful in the future study of general nonlinear optimization with matrix variables. Finally the complexity of the algorithm is built up on the aforementioned results.

SOCO is a specific class of problems lying between LO and SDO. Theoretically the solution of a SOCO problem can be obtained by finding the optimal solution of an SDO problem relevant to the original SOCO problem. However, for this one usually has to face both a large increase in the problem's size and the loss of certain advantages of the original problem. Thus, with regard to IPMs, the algorithm working on the second-order cone is a favorable choice for solving SOCO. To build up the complexity of the algorithm for SOCO, we first explore in Chapter 6 versatile properties of the functions associated with the second-order cone and estimate their first and second directional derivatives. As a byproduct, these estimates also verify some of our conclusions about matrix functions in Chapter 5. Particular attention is paid to self-regular functions and self-regular proximities on second-order cones. Then, based on these *self-regular* proximities, new search directions are suggested. It is shown that the algorithm for SOCO has the same complexity as its analogue for LO. Let us emphasize again that only one linear system need to be solved at each iteration.

Chapter 7 describes a variety of homogeneous models for classes of problems considered

[4]For LO and LCPs, we need only to identify a primal-dual pair (x, s) satisfying a linear equation and the complementarity certification.