NEW ADAPTIVE INTERIOR POINT ALGORITHMS

Dedicated to:

My wife and son for their love, encouragement, and our families for their constant support

NEW ADAPTIVE INTERIOR POINT ALGORITHMS FOR LINEAR OPTIMIZATION

By

MAZIAR SALAHI, B.Sc., M.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2006) McMaster University
(Mathematics and Statistics) Hamilton, Ontario

TITLE: New Adaptive Interior Point Algorithms for Linear Optimization

AUTHOR: Maziar Salahi
B.Sc. (The University of Guilan)
M.Sc. (Sharif University of Technology)

SUPERVISOR: Professor Tamás Terlaky
Department of Computing and Software

NUMBER OF PAGES: xviii, 190

Abstract

Interior Point Methods (IPMs) have shown their power in solving large scale optimization problems. In this thesis, using the notion of self-regularity, various classes of algorithms are proposed. First we present an adaptive technique to compute the target barrier parameter. Using this adaptive technique a single step large-update algorithm is proposed that enjoys better worst case iteration complexity than the classical large-update IPMs. A new variant of infeasible IPMs, using the same adaptive technique, is proposed that enjoys better worst case iteration complexity than the classical infeasible IPMs. By further use of the adaptive technique a new family of predictor-corrector IPMs is proposed that are different both in the predictor and the corrector steps from their classical counter parts. Worst case iteration complexity analysis reveals significant improvement for some special cases over the classical analogues.

In the second part of thesis we analyze the theoretical behavior of the most widely used algorithm in IPM based software packages, the so called Mehrotra-type predictor-corrector algorithm. This analysis reveals some drawbacks of the algorithm that motivated us to modify it. We combine it with a safeguard that prevents the drawback we observed and enables us to prove strong theoretical results about the new algorithm. Motivated by this drawback, we analyze it from different perspective. In the new approach, contrary to the existing variants in the literature, first we fix the step size and then aim to compute the smallest value of the barrier parameter that ensures the prescribed step size is feasible. After the barrier parameter is computed, the algorithm computes the next iterate by a line-search as usual. The worst case behavior of the proposed algorithms is discussed. Finally, some limited encouraging computational results are reported.

Acknowledgements

I would like to express my kind and sincere thanks to my Ph.D. supervisor, professor Tamás Terlaky, who kindly accepted me to visit him in October 2002 as a visiting scholar from Sharif University of Technology, Tehran, Iran. Later by his suggestion, I started my Ph.D. program under his guidance at McMaster University. I am really indebted to him for bringing me to this field and teaching me. I am also grateful to Dr. Jiming Peng, from our research group (AdvOL) and one of my supervisory committee member, for all of the valuable discussions. We had many fruitful discussion about my research. I would like also thanks the other members of my Ph.D. Supervisory Committee, Dr. Bartosz Protas and Dr. Dmitry Pelinovsky for their guidance during the past three years, for the comments and corrections on early versions of the thesis. The friendly helps of AdvOL members will not be forgotten. I was also lucky to have a warm community of Iranian friends in Hamilton during my studies. I consider it as my most valuable asset which provided me and my family an enjoyable environment in Hamilton. My studying at McMaster financially was supported by McMaster University, MITACS and partially by the Iranian Ministry of Science, Research and Technology. I express my special thanks to all of them.

At last, but not least, special thanks to my wife, son and our families for all their love, constant support and encouragement.

Contents

List of Figures

List of Tables

Notations

A: $m \times n$ coefficient matrix;

A^T: transpose of the matrix A;

e: vector with unit components;

$\mathcal{I}$: index set;

L: the input length of an LO problem;

μ: the barrier parameter or centering parameter;

$\mathcal{R}$: one dimensional Euclidian space;

$\mathcal{R}^n$: the n-dimensional Euclidian space;

$\mathcal{R}^n_+(\mathcal{R}^n_{++})$: nonnegative (positive) orthant in $\mathcal{R}^n$;

x: the vector of the primal variables;

y, s: the vectors of the dual variables;

x_i: the i-th coordinate of the vector x;

x^T: the transpose of vector x;

xs: coordinate-wise (Hadamard) product of the vectors x and s;

x^{-1}: componentwise inverse of vector x;

x^q: componentwise power q of vector x;

x^{-q}: componentwise inverse of vector x^q;

$\frac{x}{s}$: $= xs^{-1}$;

$x \geq 0$: all components of the vector x are nonnegative;

$x > 0$: all components of the vector x are strictly positive;

Δx: the primal search direction;

$\Delta y, \Delta s$: the dual search directions;

d_x: the scaled primal search direction;

d_y, d_s: the scaled dual search directions;

v: the scaled vector defined by $\sqrt{\frac{xs}{\mu}}$;

v_i: the i-th coordinate of v;

$\psi(t)$: a general function from $\mathcal{R} \rightarrow \mathcal{R}$;

$\psi(x)$: $\psi(x) = (\psi(x_1), \cdots, \psi(x_n))^T$;

$\Psi(x)$: $\Psi(x) = \sum_{i=1}^{n} \psi(x_i)$;

$\Psi^{'}(x)$: the first derivative of $\Psi(x)$;

$\Psi^{''}(x)$: the second derivative of $\Psi(x)$;

$\mathcal{N}$: the neighborhood of the central path;

X: the diagonal matrix diag (x) whose i-th diagonal element is x_i;

S: the diagonal matrix diag (s) whose i-th diagonal element is s_i;

τ: the proximity parameter;

μ_g: $= \frac{x^T s}{n}$;

Preface

Linear optimization (LO) problems have numerous applications in real world, in various areas of economics, transportation, VLSI design and many others [10, 14, 39, 59]. The traditional method to solve an LO problem is the famous simplex method and its variants [14, 60]. This method was developed in 1947, by George Dantzig. Since then, hand-in-hand with the development of computing technology, several high performance commercial software packages have been developed based on variants of the simplex method [13, 64]. In spite its high efficiency in computational practice, the exponentiality of the number iteration in number of variables for simplex method was a strong motivation to look for alternative approaches that are polynomial in the number of iterations. The first polynomial algorithm, the ellipsoid method was proposed by Khachian in 1979 [25]. In 1984 Karmarkar published his famous projective algorithm that induced a highly active research area of *Interior Point Methods (IPMs)*. The past two decades of IPMs resulted in hundreds results that changed the state of the art of all areas of optimization significantly.

Due to the notable practical efficiency of IPMs and the ongoing importance of the simplex methods, several high performance software packages based on both algorithm classes have been developed in the last two decades [2, 3, 13, 58, 61, 64, 70]. Due to the simple structure of an LO problem and the existence of many, easy to use reliable, fast and high performance software packages, the application domain of LO expanded rapidly as well. The IPM methodology not only revolutionized the theory and practice of LO, it also allowed to discover many classes of optimization problems, such as *Second*

Order Conic Optimization and *Semidefinite Optimization* [1, 11, 29, 62] that can be solved by IPMs efficiently. Further new application areas, such as data classification [46] are discovered, which certain LO subproblems have to be solved in order to get some useful information about the original model.

A new paradigm of IPMs was recently introduced by Peng et al. [41, 42]. They introduced the notion of *Self-Regular (SR) functions* that enabled them to reduce the discrepancy between theoretical and practical efficiency of IPMs by reducing the iteration complexity of large-update large neighborhood IPMs.

In this thesis, using this tool, we propose several new algorithms that are enjoying better worst case iteration complexity than classical IPMs in the same class. First in Chapter 3 we propose an adaptive scheme that enables us to design a single step large-update algorithm which enjoys an $\mathcal{O}\left(n^{\frac{q+1}{2q}}\log\frac{n}{\epsilon}\right)$ (ϵ is the desired accuracy, n is the dimension of the underlying problem, and q is the barrier degree of the SR function) that is a factor of $\mathcal{O}(n^{\frac{q-1}{2q}})$ better than the complexity of its analogues. Using this adaptive technique a new variant of infeasible IPMs is proposed in Chapter 4. Then, by further using the adaptive scheme, a new variant of predictor-corrector algorithms is developed in Chapter 5. This variant has multiple choices in the predictor and corrector steps using the idea of self-regularity while it is not the case in classical analogues and in certain cases this leads to significant improvement in the iteration complexity compared to its analogues. [38, 45, 49].

In the rest of thesis we analyze theoretical properties of a widely used algorithm that is the backbone of most IPM based software packages. This is the so called Mehrotra-type predictor-corrector algorithm. By a numerical example we show that this algorithm can be very inefficient in practice, therefore it has to be combined with a certain safeguard that gives a lower bound for

the maximum step size at each iteration, which itself implies better polynomial iteration complexity and faster convergence. By slight modification of Merhotra's updating scheme of the barrier (centering) parameter, while keeping the adaptivity, we prove the same order of polynomiality with the superlinear convergence. These results are presented in Section 6.3. Then, in Section 6.4 a second modification of the algorithm is proposed that enjoys further improved iteration complexity.

The identified drawbacks of Mehrotra's algorithm motivated us to analyze this class of practical predictor-corrector IPMs from a different perspective. In the new approach, which is presented in Chapter 7, in contrast to the earlier published variants, first we fix the step size and then aim to compute the smallest value of the barrier parameter that ensures that the chosen step size is feasible. After the barrier parameter is computed it makes the next iterate as usual. The algorithm based on this idea as analyzed in Section 7.1 enjoys the same order of worst case iteration complexity as the algorithm of Section 6.3, namely $\mathcal{O}(n^2 \log \frac{n}{\epsilon})$, without requiring any safeguard scheme. A further modification (see Section 6.3) of this algorithm allows us to prove an $\mathcal{O}(n \log \frac{n}{\epsilon})$ iteration complexity, the same as the algorithm of Section 6.4. Superlinear convergence of both algorithms of Sections 7.1 and 7.2 are established.

We implemented our new algorithms by using the McIPM and LIPSOL software packages [70, 72]. We have modified certain subroutines of these software packages that match to our new algorithms. The test problems for comparison of our algorithms with McIPM and LIPSOL are taken from the standard test set for LO problem, namely the NETLIB test repository [40]. We report some limited computational results in Chapter 8. Our limited computational results show that our new algorithms besides their strong theoretical

properties are competitive with those two state-of-the-art software packages [70, 73].

Chapter 1

Introduction

In this chapter, first we present the standard form *Linear Optimization* (LO) problem [50] that we are dealing with throughout the thesis. Then, for completeness and ease of understanding, we summarize some fundamental concepts and duality results about LO problems. The results presented in this chapter can be found e.g. in [14, 50, 63, 66].

1.1 The Linear Optimization Problem

Throughout this thesis we consider the following standard LO problem:

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

where $A \in R^{m\times n}$ satisfies $\text{rank}(A) = m,\ b \in R^m,\ c \in R^n$ are given vectors and the vector $x \in R^n$ is the vector of variables. The dual LO problem is given by

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

where $y \in R^m$ and $s \in R^n$ are the vectors of unknowns. For simplicity we denote the primal feasible solution by $\mathcal{F}_P$ and the dual feasible solution by

$\mathcal{F}_D$, i.e.,

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

and the set of strictly feasible solution by

$$\mathcal{F}_P^0 = \{x|\ Ax = b,\ x > 0\} \quad \text{and} \quad \mathcal{F}_D^0 = \{(y,s)|\ A^T y + s = c,\ s > 0\},$$

respectively. For ease of reference we use the notation $\mathcal{F} = \mathcal{F}_P \cup \mathcal{F}_D$.

It is worth mentioning that any formulation or representation of an LO problem can be transformed to the standard form, although they can be studied and solved in their original form [50, 63] too.

1.2 Duality Results

In this section we review some fundamental duality results of LO. The following theorem gives the fundamental duality relation between the objective values of problems (P) and (D).

Theorem 1.2.1 (Weak Duality) *Suppose that both $\mathcal{F}_P$ and $\mathcal{F}_D$ are nonempty. Then for any $x \in \mathcal{F}_P$ and $(y,s) \in \mathcal{F}_D$ we have*

$$c^T x \geq b^T y.$$

Proof: If we multiply the constraint of the dual problem by x^T, then we get

$$c^T x = x^T A^T y + x^T s = b^T y + x^T s,$$

where the second inequality follows by using the constraint of (P). Since $x, s \geq 0$, implies $x^T s \geq 0$ and therefore $c^T x \geq b^T y$ that completes the proof. $\square$

The difference $c^T x - b^T y$ is called the *duality gap*. There are several simple consequence of the weak duality theorem. We present those that are needed for our further developments. First let us define the notion of unboundedness for an LO problem.

Definition 1.2.1 *An LO problem (P) is unbounded if there exists a sequence of feasible points $\{x_k\}_{k=1}^{\infty}$ for which the objective value $c^T x_k \longrightarrow -\infty$.*

Corollary 1.2.1 *If the primal problem (P) is unbounded, then the dual problem (D) is infeasible. Analogously, if the dual problem (D) is unbounded, then the primal problem (P) is infeasible.*

Corollary 1.2.2 *If for $x \in \mathcal{F}_P$ and $(y, s) \in \mathcal{F}_D$, $c^T x = b^T y$ holds, then x and (y, s) are optimal solutions of (P) and (D), respectively.*

Now the following question arises after a short introduction:

Is there always a feasible primal-dual solution that have zero duality gap?

This question can be answered by using the famous Farkas Lemma that follow:

Theorem 1.2.2 **[Farkas Lemma, Primal Form]** *Let A be an $m \times n$ matrix and $c \in R^n$. Then exactly one of the following systems has solution:*

- *There exists an x such that $Ax = b,\ x \geq 0$.*
- *There exist a $y \in R^m$ such that $A^T y \leq 0$ implies $b^T y > 0$.*

Proof: Suppose that there exist an x such that $Ax = b$ and $x \geq 0$. If there exists a vector y such that $A^T y \leq 0$, then

$$y^T b = y^T Ax = (y^T A)x \leq 0.$$

Therefore, $A^T y \leq 0$ does not imply $b^T y > 0$. Now let us assume that there is no x such that $Ax = b$ and $x \geq 0$. Define $\mathcal{K} = \{Ax : \ x \geq 0\}$. Note that $\mathcal{K}$ is a closed convex set and $b \notin \mathcal{K}$. Therefore, by Theorem 10.1.2 there exist a vector $y \in R^m$ and a scalar η such that $b^T y > \eta$ and $y^T z \leq \eta$ for any $z \in \mathcal{K}$. Since $0 \in \mathcal{K}$, then $\eta \geq 0$ and so $b^T y > 0$. Also, $\eta \geq y^T Ax = x^T A^T y$ for all $x \geq 0$. Since $x \geq 0$ can be made arbitrary large, the last inequality implies that $A^T y \leq 0$. Therefore, we have constructed a vector $y \in R^m$ such that $A^T y \leq 0$ and $b^T y > 0$ that completes the proof of the theorem. □

Remark 1.2.1 *A vector y, with $A^T y \leq 0$ and $b^T y = 1$ is called a primal infeasibility certificate.*

Theorem 1.2.3 [**Farkas Lemma, Dual Form**] *Let A be an $m \times n$ matrix and $c \in R^n$. Then exactly one of the following systems has solution:*

- $Ax = 0, \ x \geq 0, \ c^T x < 0$ *for some* $x \in R^n$.
- $A^T y \leq c$ *for some* $y \in R^m$.

Proof: Analogous to the proof of Theorem 2.1. □

Remark 1.2.2 *A vector $x \geq 0$, with $Ax = 0$ and $c^T x = -1$, is called a dual infeasibility certificate.*

Corollary 1.2.3 *If (D) is infeasible, then (P) is either infeasible or unbounded.*

Proof: If (D) is infeasible, then there is no y such that

$$A^T y \leq c.$$

Then, by Theorem 1.2.3 there exists an x such that

$$c^T x < 0,\ Ax = 0,\ x \geq 0.$$

If (P) is feasible i.e., there exists an x^0 such that

$$Ax^0 = b,\ x^0 \geq 0.$$

Therefore

$$A(x_0 + rx) = b,\ x_0 + rx \geq 0 \quad \text{and} \quad c^T(x_0 + rx) \to -\infty, \text{when } r \to \infty$$

that completes the proof. □

Theorem 1.2.4 [**Strong Duality Theorem**] *If one of the problems (P) or (D) has an optimal solution, then so does the other and the optimal objective values of the two problems are equal.*

Proof: Suppose that problem (P) has an optimal solution x^*. By Theorem 1.2.3 the dual problem has a feasible solution, otherwise, primal problem is unbounded. Therefore, by Weak Duality Theorem one has $c^T x^* \geq b^T y$ for any dual feasible solution. The goal is to show that there exist a $y \in R^m$ such that $c^T x^* \leq b^T y$. To do so, for any $\zeta < c^T x^*$ we know that the following system does not have any solution, since x^* is the optimal solution of (P).

$$\begin{aligned} c^T x &\leq \zeta \\ Ax &= b \\ x &\geq 0. \end{aligned}$$

Therefore, by Theorem 2.1 the following system is solvable:

$$A^T y - \lambda c \leq 0$$

$$\begin{aligned} b^T y - \lambda\zeta &> 0 \\ \lambda &\geq 0. \end{aligned}$$

Now we have to show that $\lambda \neq 0$. If so, then the following system has a solution:

$$\begin{aligned} A^T y &\leq 0 \\ b^T y &> 0. \end{aligned}$$

By Theorem 1.2.3 this implies that problem (P) is infeasible, which is certainly a contradiction. Finally, by dividing all the inequalities in (1.3) by $\lambda > 0$ we have a feasible dual solution for which $b^T y > \zeta$. Since ζ was an arbitrary value less than $c^T x^*$, then $b^T y > c^T x^*$, which completes the proof of the theorem. □

Therefore, using the Strong Duality Theorem, finding an optimal solutions to (P) and (D) is equivalent to solving the following system of equations, which is referred to the **Optimality Conditions**.

$$\begin{aligned} Ax &= b,\ x \geq 0 \\ A^T y + s &= c,\ s \geq 0 \\ x_i s_i &= 0 \quad \forall i = 1, \cdots, n. \end{aligned} \tag{1.3}$$

As we see the first two set of conditions are the primal and dual feasibility, respectively, and the last set of constraints is called the *complementarity conditions.*

It can happen that the problem (P) or (D) (possibly both) is infeasible. If one of them is infeasible, then we have the following result using Farkas Lemma.

Corollary 1.2.4 *Suppose that problem (P) is feasible. Then its objective function $c^T x$ is bounded below if and only if problem (D) is feasible.*

Similarly, assuming that problem (D) is feasible, then its objective function $b^T y$ is bounded above if and only if problem (P) is feasible.

1.3 From Simplex Methods to IPMs

As it is known, simplex algorithms [14] start from a given feasible basis solution (vertex of the feasible region, which is a polyhedron), then they move to another feasible basis solution with monotonically improving objective value. Geometrically this can be interpreted as moving along the edges of the polyhedron. The process is repeated until the basis solution (vertex) is proved to be optimal or identifies unboundedness of the problem. The Klee-Minty example [27] shows that simplex methods may require an exponential number of steps. This phenomenon has motivated researchers to search for novel algorithms that solve LO problems in a polynomial number of iterations.

The first polynomial-time algorithm for LO is the *ellipsoidal method* proposed by Khachian in 1979 [25]. The algorithm has a polynomial $\mathcal{O}(n^2L)$ iteration complexity with a total $\mathcal{O}(n^4L)$ bit operations, where L is the input size of the LO problem. Despite its favorable complexity, the performance of ellipsoid method in practice was extremely slow, much slower than the simplex method. In contrast to simplex method, the number of iterations of ellipsoid method tented to be comparable to its enormous upper bound. Therefore the simplex method remained the only efficient method to solve LO problems [13, 64].

In 1984, a projective IPM was developed by Karmarkar [26] that had

$\mathcal{O}(nL)$ iteration complexity with a total $\mathcal{O}(n^{3.5}L)$ bit operations. In contrast to the simplex methods which start from a vertex of the feasible set, Karmarkar's method starts somewhere in the interior of the feasible region and moves along a path towards optimality [26]. Karmarkar's epoch-making paper sparked the IPM revolution. Since then thousands of papers have been published and many highly efficient commercial software packages were developed.

The main results of this monumental research efforts are summarized in several books, e.g. [42, 50, 63, 66]. The achievement of computational/software developments can be found in [2, 3, 4, 13, 61]. The impact of IPMs did not remain within the area of LO, it also generalized to many other class of optimization problems such as *Second Order Conic*, *Semidefinite* and *Nonlinear Optimization* [11, 12, 62, 66, 69].

Chapter 2

Interior Point Methods

In this chapter we discuss some of the basic concepts of *Interior Point Methods* (IPMs) for LO.

2.1 The Interior Point Condition

Since IPMs require an interior point to start with, we need to have that both $\mathcal{F}_P^0 \neq \emptyset$ $\mathcal{F}_D^0 \neq \emptyset$. First we introduce this condition and discuss some of its consequences.

Definition 2.1.1 *The problems (P) and (D) satisfy the Interior Point Condition (IPC) if there exist vector x such that*

$$Ax = b, \quad x > 0$$

and vectors y and s such that

$$A^T y + s = c, \ s > 0.$$

The following result relates the IPC to the boundedness of the level set of the primal and dual solution sets.

Theorem 2.1.1 *Suppose that both $\mathcal{F}_P \neq \emptyset$ and $\mathcal{F}_D \neq \emptyset$ hold. If (D) has a strictly feasible point, then any nonempty level set of the set of feasible solution of problem (P) is bounded. Similarly, if the problem (P) has a strictly feasible point, then any nonempty level set of the set of feasible solution of problem (D) is bounded.*

Proof: We prove the first statement, the second one also can be proved analogously. Let us denote the given strictly feasible point of the dual problem by $(\bar{y}, \bar{s})$ and let $x \in \mathcal{F}_P^\gamma = \{x \in \mathcal{F}_P |\ c^T x \leq \gamma\}$. Then one has

$$\bar{s}^T x = \sum_{i=1}^{n} \bar{s}_i x_i = c^T x - b^T \bar{y} \leq \gamma - b^T y.$$

Since all the terms in the summation on the left are nonnegative, we have

$$x_i \leq \frac{1}{\bar{s}_i}(\gamma - b^T y),\ \forall i \Longrightarrow \|x\|_\infty \leq (\gamma - b^T y) \left\|(\bar{s})^{-1}\right\|_\infty,$$

where $(\bar{s})^{-1}$ is the componentwise inverse of the vector $\bar{s}$. Since x was an arbitrary element of $\mathcal{F}_P^\gamma$, we may conclude from this inequality that $\mathcal{F}_P^\gamma$ is bounded, which completes the proof. □

Corollary 2.1.1 *With the assumption of Theorem 2.1.1 the set of optimal solutions is nonempty for both (P) and (D).*

Proof: Existence of the optimal solution follows from the Strong Duality Theorem and boundedness follows from Theorem 2.1.1. □

The IPC ensures the aforementioned appealing properties, however, the IPC does not hold for all LO problems, although they might still be feasible

and may still have finite optimal solutions. As an example let us consider the following problem:

$$\begin{aligned} \min \quad & x_1 \\ & x_1 + x_3 = 0, \\ & x_1,\ x_2 \geq 0 \end{aligned}$$

and its dual

$$\begin{aligned} \max \quad & 0 \\ & y_1 + s_1 = 1 \\ & s_2 = 0 \\ & y_3 + s_3 = 0 \\ & s_1, s_2, s_3 \geq 0. \end{aligned}$$

As we may see any feasible vectors (x, y, s) of the primal and dual problems satisfy $x_1 = x_3 = s_2 = 0$, therefore the interior of both feasible sets are empty, while the problems have optimal solutions and the optimal objective value is zero.

Although many primal-dual methods require strictly feasible starting point, they can be adapted to handle the case in which no such points exist. Infeasible starting point algorithms and the use of self-dual embedding model are two competitive approaches that can handle problems with empty interior, which do not satisfy the IPC. The self-dual embedding model is discussed in details in Section 10.2 of the Appendix [50]. A variant of infeasible starting point algorithms is presented in Chapter 4 of this thesis, one also can consult Chapter 6 of [63] for a classical infeasible algorithm. How to choose a starting

point for practical efficiency is an important concept which is discussed in Section 10.3 of the Appendix.

2.2 Strict Complementarity

Strict complementarity is another important concept in the theory of IPMs that leads to various interesting results in convergence theory and further extensions, like sensitivity analysis [50].

Definition 2.2.1 *A feasible primal-dual pair* (x, y, s) *satisfying the conditions*

$$x_i s_i = 0 \quad \text{and} \quad x_i + s_i > 0, \ i = 1, \cdots, n$$

is said to be a strictly complementary optimal solution.

The following theorem due to *Goldman* and *Tucker* proves the existence of such a solution. This result is crucial to the validity of the self-dual embedding model (see Section 10.2) and prove the local convergence of the primal-dual algorithms.

Theorem 2.2.1 *If both problems (P) and (D) are feasible, then there exist a strictly complementary pair of optimal solutions.*

Proof: See [50]. □

It is also worth mentioning that a given LO problem may have multiple primal-dual solutions, some that are strictly complementary and others that are not i.e.,

$$\min \qquad x_1$$

$$x_1 + x_2 + x_3 = 1$$
$$x \geq 0,$$

whose dual is

$$\begin{aligned} \max \quad & y \\ & y + s1 = 1 \\ & y + s_2 = 0 \\ & y + s_3 = 0 \\ & s \geq 0. \end{aligned} \tag{2.1}$$

Primal-dual solutions are

$$x^* = (0, t, 1, -t),\ y^* = 0,\ s^* = (1, 0, 0) \quad \text{for any} \quad t \in [0, 1].$$

For $t \in (0, 1)$ it is clear that (x^*, y^*, s^*) is a strictly complementary solution. If t takes one of the end points, then the solution is no longer strictly complementary.

2.3 Primal-Dual IPMs

In this section we discuss the origin of the IPMs and some fundamental results. All primal-dual IPMs generate a sequence of points (x^k, y^k, s^k) that satisfy the IPC. Like most iterative algorithms in optimization, primal-dual IPMs have two basic ingredients: a procedure for determining a step and a measure of desirability of each point in the search space. Newton's method also is used to generate the search directions. Specifically, by applying Newton's method

to system (1.3) one has

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -Xs \end{bmatrix}. \tag{2.2}$$

A full step along this direction is not usually permissible, since it would violate the $(x, s) > 0$ constraint. To deal with this requirement, a line search along the Newton direction is performed so that the new iterate is

$$(x^+, y^+, s^+) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$$

for some line search parameter $\alpha \in (0, 1]$ that still satisfies $(x^+, s^+) > 0$. Unfortunately, one often can take only a small step along this direction before violating the IPC. Hence, the pure Newton direction does not allow us to make good progress towards an optimal solution. This motivates us to consider a different approach which is described in the sequel.

Since one of the goals of IPMs is to preserve the IPC for each iterate, the last constraint of (1.3) is replaced by

$$Xs = \mu e,$$

where μ is a positive parameter and e is all one vector. The parameterized system of equations become

$$\begin{aligned} Ax &= b,\ x > 0 \\ A^T y + s &= c,\ s > 0 \\ Xs &= \mu e. \end{aligned} \tag{2.3}$$

This is also done in a different way by Frisch [15, 16]. He removed the non-negativity constraint by augmenting the objective function with a logarithmic barrier function given by:

$$(P_\mu) \qquad \min \left\{ c^T x - \mu \sum_{j=1}^{n} \log x_j : Ax = b, x > 0 \right\}, \tag{2.4}$$

where $\mu > 0$ is a parameter. The logarithmic term forces the variables to stay away from the boundary of the feasible region at an optimal solution, thus if $x(\mu)$ denotes the optimal solution of problem 2.4, then $x(\mu) > 0$ holds. It is easy to check that the objective function of (2.4) is strictly convex, therefore the KKT conditions [9, 15] for problem (2.4) is exactly (2.3). Thus, one can state the following result immediately.

Proposition 2.3.1 *Let* $\text{rank}\,(A) = m$ *and IPC holds, then the Problem (2.3) has a unique solution.*

Equation (2.3) defines a smooth curve (Figure 2.1) $\{(x(\mu), y(\mu), s(\mu)) : \mu > 0\}$ called *the primal-dual central path* that plays an important concept in the theory of IPMs for LO [33, 57]. In addition $\{x(\mu) : \mu > 0\}$ is called the primal central path and $\{(y(\mu), s(\mu)) : \mu > 0\}$ is called the dual central path. It has been shown that the limit of the central path (as μ goes to zero) exists. Because the limit point satisfies the complementarity condition, it naturally

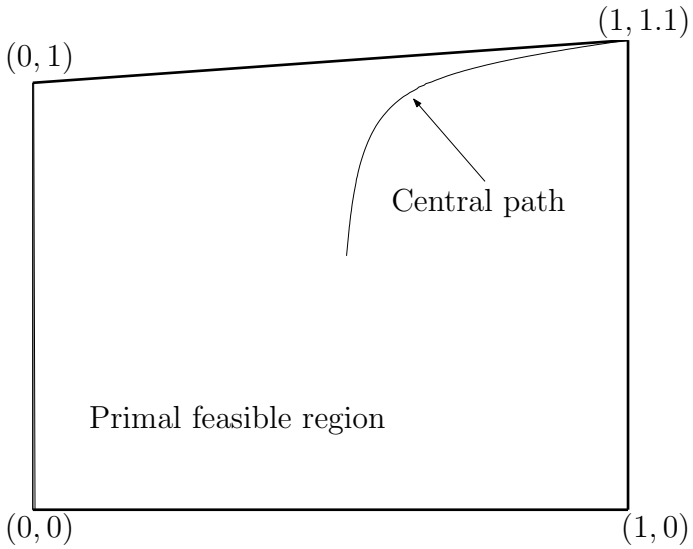

Figure 2.1: The Central path for problem (6.5)

yields optimal solutions for both the primal and the dual problems, respectively [50, 63]. Moreover, the limit point of the central path is a strictly complementary optimal solution. Many other interesting properties of the central path like analyticity also are proved (see [50] and references therein).

2.4 Path-Following IPMs

Path-following IPMs follow the central path in the direction of decreasing of μ parameter values toward to the solution sets of problems (P) and (D). The iterates do not stay exactly on the central path or even too close to it. Rather, they stay in well defined neighborhoods of the central path while steadily reducing the duality gap to zero. Therefore, first we have to define what can be considered as a neighborhood in path-following IPMs.

Definition 2.4.1 *A neighborhood of the central path is defined by*

$$\mathcal{N}(\delta) := \{(x, y, s) \in \mathcal{F}^0 \mid \Phi(x, s, \mu) \leq \delta\}, \tag{2.5}$$

where $\Phi(x, s, \mu)$ is a proximity function to measure the distance of the current point to the central path and δ is the proximity parameter.

Various proximity measures and neighborhoods can be found in the IPMs literature, see also the books [50, 63]. Here we list some of them for the sake of self completeness.

- Let $\theta \in [0, 1]$ and let us define the 2-norm neighborhood as

$$\mathcal{N}_2(\theta) = \{(x, y, s) \in \mathcal{F}^0 \mid \|Xs - \mu e\| \leq \theta\mu\}. \tag{2.6}$$

- Let $\gamma \in (0, 1)$ and let us define the negative infinity norm neighborhood as

$$\mathcal{N}_\infty^-(\gamma) = \{(x, y, s) \in \mathcal{F}^0 \mid x_i s_i \geq \gamma\mu \ \ \forall i = 1, \cdots, n\}. \tag{2.7}$$

These are the most widely used neighborhoods in the development of the theory of IPMs. It is worth to mention that $\mathcal{N}_2(\theta)$ is a small neighborhood, because for an iterate $(x, y, s) \in \mathcal{N}_2(\theta)$ we have

$$\sum_{i=1}^{n} \left(\frac{x_i s_i}{\mu} - 1 \right)^2 \leq \theta^2 < 1.$$

Therefore, even if θ is close to one, the neighborhood $\mathcal{N}_2(\theta)$ contains only a small fraction of the points in $\mathcal{F}^0$. Thus, the algorithm based on this neighborhood will not have much room to operate. On the other hand the neighborhood $\mathcal{N}_\infty^-(\gamma)$ is much larger. When γ is small, it takes up almost the entire strictly feasible set $\mathcal{F}^0$. For this reason most IPMs based software packages are somehow using this neighborhood. In the next section we discuss how a Newton step can be taken while staying in a predefined neighborhood.

2.4.1 The Newton Direction

Let us first assume that the current iterate is strictly feasible[1] and it belongs to a predefined neighborhood. The goal is to move to another point for which the IPC holds while staying in the predefined neighborhood. The traditional way is to apply Newton's method to system (2.3), then move in the directions provided by the Newton method which is given by[2]

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs. \end{aligned} \tag{2.8}$$

The following lemma is an immediate consequence of system (2.8).

Lemma 2.4.1 *Let $(\Delta x, \Delta y, \Delta s)$ be a solution of system (2.8), then*

$$\Delta x^T \Delta s = 0.$$

Proof:

$$\Delta x^T \Delta s = \Delta x^T(-A^T\Delta y) = -(A\Delta x)^T\Delta y = 0.$$

□

It is worth mentioning that the orthogonality property of the displacements leads to much simpler analysis of feasible IPMs compared to the infeasible IPMs for which such a property does not the hold in general.

[1]The infeasible case will be considered later.

[2]Infeasible algorithms also solve such a system with considering the residuals in the primal and dual constraints (see Chapter 4 for more details).

2.4.2 Step Size

After one solved the Newton system (2.8), then a certain step size has to be chosen that preserves the feasibility of the new iterate, i.e.,

$$\alpha^{\max} = \arg\max_{\alpha \geq 0} \{(x, s) + \alpha(\Delta x \ \Delta s) \geq 0\}.$$

In order to guarantee strict feasibility, a damping factor $\lambda \in (0\ 1)$ is needed, then the new iterate becomes

$$(x^+, s^+) = (x, s) + \lambda\alpha(\Delta x, \Delta s).$$

2.4.3 Centering (Barrier) Parameter

Throughout the thesis we call the parameter μ the centering or barrier parameter. Now we discuss how it is chosen during an iterative scheme. Let us denote $\mu_g := \frac{x^T s}{n}$, where n is the dimension of the problem. Then one can define μ by

$$\mu = (1 - \theta)\mu_g,$$

where $\theta \in (0\ 1)$. How to choose the parameter θ is an important issue that has been addressed in several ways in the literature [35, 50, 63]. We will discuss several strategies for choosing θ in coming sections and chapters.

2.4.4 Path-Following Algorithms

The primal-dual path-following algorithm was first proposed by Kojima et al. [28]. This method approaches to the optimal solution through a sequence of the strictly feasible solution. Following the discussion of the previous sections we can outline the general form of the primal-dual path-following algorithm as it follows.

Algorithm 2.4.4: Prima-Dual Path-Following Algorithm

Input:

A proximity parameter δ;

an accuracy parameter $\varepsilon > 0$;

a fixed barrier update parameter $0 < \theta < 1$;

$(x^0, y^0, s^0) \in \mathcal{F}^0$ such that $\Phi(x^0, s^0, \mu^0) \leq \delta$.

begin

$x := x^0;\ y = y^0;\ s := s^0;\ \mu_g = \frac{x^T s}{n}$

while $x^T s \geq \epsilon$ **do**

begin

$\mu := (1 - \theta)\mu_g$;

while $\Phi(x, s, \mu) \geq \delta$ **do**

begin

Solve the Newton system for $\Delta x, \Delta y, \Delta s$;

Determine the maximum step size $\bar{\alpha}$

such that $(x + \bar{\alpha}\Delta x, s + \bar{\alpha}\Delta s) \geq 0$;

Let $\alpha = \arg \min_{\alpha \in (0\ \bar{\alpha}]} \Phi(x + \alpha\Delta x, s + \alpha\Delta s, \mu)$;

Set

$x := x + \alpha\Delta x$;

$s := s + \alpha\Delta s$;

$y := y + \alpha\Delta y$.

end

end

end

As we see the algorithm has inner and outer iterations. At each outer iteration the duality gap is reduced by a factor of $(1-\theta)$. Then it performs inner iterations until $\Phi(x, s, \mu) \leq \delta$. It is worth mentioning that if the parameter θ is independent of n, the dimension of the underlying problem, then the algorithm called large-update (or long-step) algorithm and small-update (or short-step) if θ depends on n.

The worst case iteration bound for small-update methods (Figure 2.2) is $\mathcal{O}(\sqrt{n}\log\frac{n}{\epsilon})$ and for large-update methods (Figure 2.3) is $\mathcal{O}(n\log\frac{n}{\epsilon})$ [50, 63]. Although small-update methods enjoy better iteration bound than large-update methods, but they are not as efficient as the large-update methods in practice. Therefore, this creates a discrepancy between theory and practice. Various approaches have been proposed to remove this discrepancy between theory and practice [23, 43, 42]. Some of them will be discussed later on.

2.5 Predictor-Corrector Methods

In this section we discuss a different approach for large-update and small update methods, namely the predictor-corrector methods. It is worth mentioning that these are the methods used in computational practice. At each iteration, these methods consists of two iterations. The first one is called the predictor step and the second one is called the corrector step. In the predictor step it aims to reduce the duality gap as much as possible while staying in a predefined neighborhood. In the corrector step the aim is to improve the centrality by bringing back the iterate to a slightly smaller neighborhood than the one in the predictor step. Mathematically speaking, in the predictor step it solves system (2.8) with $\mu = 0$ (affine scaling system) and makes the step

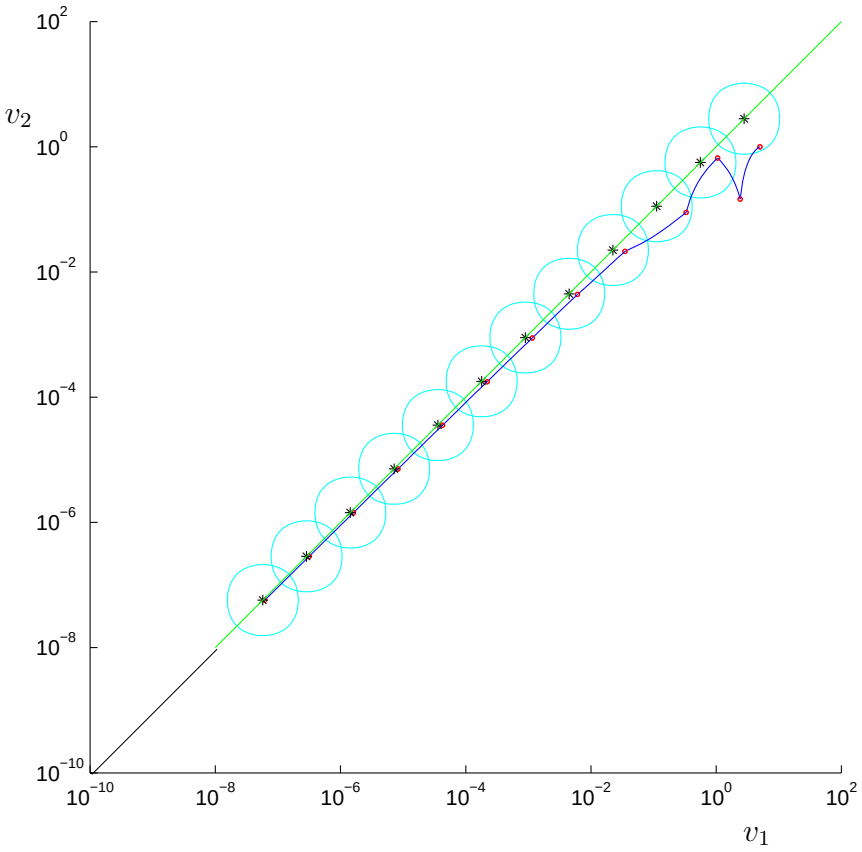

Figure 2.2: Performance of a small-update IPM, $v_i = \sqrt{\frac{x_i s_i}{\mu}}$

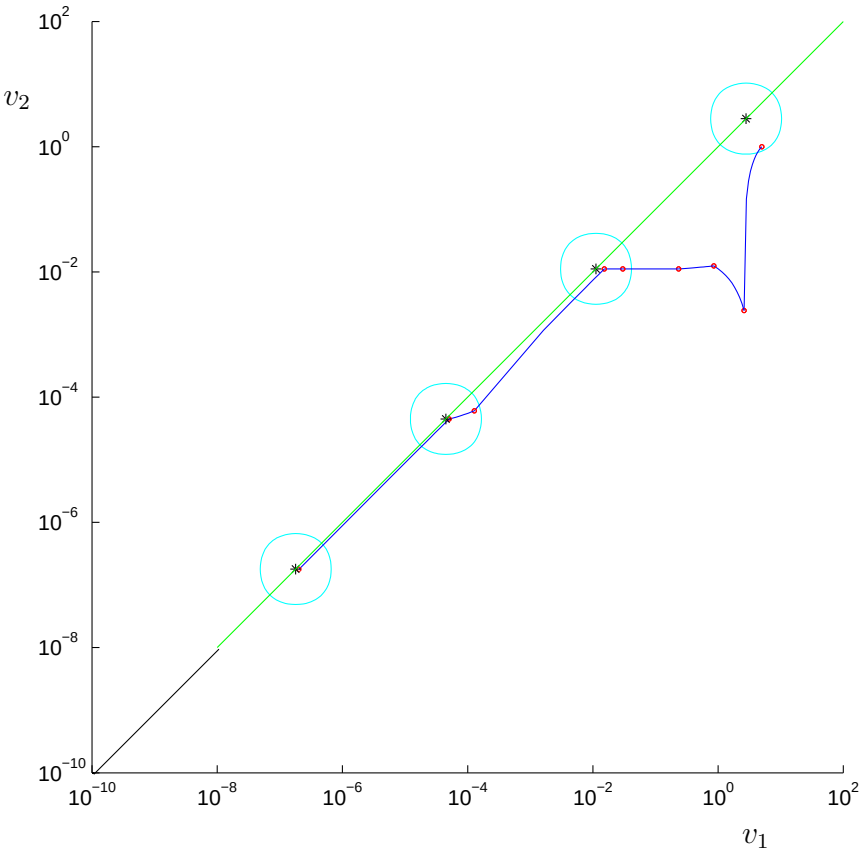

Figure 2.3: Performance of large-update IPM, $v_i = \sqrt{\frac{x_i s_i}{\mu}}$

in the computed directions while staying in a predefined neighborhood. In the corrector step it solves (2.8) with $\mu = \mu_g$ and makes the step in the corresponding direction while the next iterate stays in a smaller neighborhood than the neighborhood in the predictor step. The most famous predictor-corrector algorithm is proposed by Mizuno-Todd-Ye [38] that operates in the $\mathcal{N}_2(0.5)$ and $\mathcal{N}_2(0.25)$ neighborhoods. It starts from a given point in the $\mathcal{N}_2(0.25)$ neighborhood and makes the predictor step while the new iterate is in $\mathcal{N}_2(0.5)$. In the corrector step it brings back the iterate to the $\mathcal{N}_{0.25}(\theta)$ neighborhood. This algorithm enjoys an $\mathcal{O}\left(\sqrt{n}\log\frac{(x^0)^T s^0}{\epsilon}\right)$ worst case iteration complexity. Although it has the best worst case iteration complexity among IPMs, but due to the small neighborhoods, in which the algorithm operates, it is not efficient in practice. There have been some efforts whether it is possible to enlarge the neighborhoods of MTY-type predictor-corrector algorithms or not. Recently Potra [47] enlarged the neighborhoods in the original MTY algorithm, while he preserved the same iteration complexity (the enlargement does not reach the size which is usually used in computational practice). The idea of predictor-corrector methods also extended to large neighborhoods of central path. Anstreicher and Bosch [5] proved that the iteration complexity of a straightforward implementation of the MTY predictor-corrector method in a large neighborhood is $O(n^{\frac{3}{2}}\log\frac{n}{\epsilon})$. Using an elegant analysis, Gonzaga [22] proposed a predictor-corrector algorithm in a wide neighborhood using multiple centering steps in the corrector step. He showed that the maximum number of steps is $\mathcal{O}\left(n\log\frac{n}{\epsilon}\right)$. In [48] Potra proposed a predictor-corrector algorithm for *linear complementarity problem* (LCP) that has the same structure as the MTY algorithm. He proved that under general conditions his algorithm has an $\mathcal{O}(n\log\frac{n}{\epsilon})$ iteration complexity, and quadratic convergence of the duality

gap under the assumption that the LCP is nondegenerate. Recently, Potra and Liu have [49] proposed a predictor-corrector algorithm for sufficient LCPs with the same structure as the MTY method in the wide neighborhood $\mathcal{N}_\infty^-(\rho)$, and proved an $\mathcal{O}\left(n(\kappa+1)\log\frac{n}{\epsilon}\right)$ iteration complexity, where κ is the handicap of the sufficient LCP, and quadratic convergence for nondegenerate problems. Finally, it is worth mentioning that a more practical predictor-corrector algorithm proposed by Mehrotra [34] which is the backbone of several IPMs based software packages. This algorithm will be analyzed in details in Chapter 6.

2.6 Self-Regular IPMs

As we already discussed there exists a discrepancy between theory and practice of IPMs. Algorithms that are efficient in computational practice enjoy higher worst case iteration complexity than those that are performing poorly in practice. In this section we describe the approach proposed by Peng et al. [41, 42] to reduce the discrepancy between theory and practice. This approach is considered for further development in this thesis. Before going further, let us first present the motivation behind this development. To do so, first we need to introduce the notation

$$v := \sqrt{\frac{xs}{\mu}} \quad \text{and} \quad v^{-1} := \sqrt{\frac{\mu e}{xs}},$$

where the i^{th} components of the vector v and v^{-1} are $\sqrt{\frac{x_i s_i}{\mu}}$ and $\sqrt{\frac{\mu}{x_i s_i}}$, respectively. Then, the Newton system for the classical IPMs for LO can be written as:

$$\begin{aligned} \bar{A}d_x &= 0, \\ \bar{A}^T\Delta y + d_s &= 0, \end{aligned} \tag{2.9}$$

$$d_x + d_s = v^{-1} - v,$$

where $\bar{A} = \frac{1}{\mu} A V^{-1} X$ and $V^{-1} = \text{diag}\,(v^{-1})$. Let $d_v = d_x + d_s$. Then one can decompose system (2.9) to two systems of equations: the predictor and the centering system with the difference that the right hand side of the third equation of the first system is $(d_v)_{\text{Pred}} = -v$ and the other one is $(d_v)_{\text{Cent}} = v^{-1}$. The predictor direction aims to decrease the duality gap, while the centering direction is aiming to bring the iterate closer to the central path. It is obvious that $(d_v)_i > 0$ if $v_i < 1$ and $(d_v)_i \leq 0$ if $v_i \geq 1$. Thus, the classical approach decreases the large components and increases the small components in order to get closer to the central path. Therefore, it is reasonable to expect that if we increase the small components of v and decrease the large components, then we might get closer to the central path faster, for example one can consider the following new system:

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

where $q \geq 1$ is a parameter. This means that for the centering direction one has $(d_v)_{\text{Cent}} = v^{-q}$.

To derive a relatively deeper mathematical interpretation, let us look at it a little more. The right hand side of system is the negative gradient of the classical logarithmic barrier function in the $v-$space, which means the standard primal-dual Newton method is identical to the steepest descent method for minimizing the classical logarithmic barrier function. By a deeper look at IPM literature one can observe that most potential reduction methods for

solving LO utilize the gradient of the potential barrier function to define the search direction.

From the aforementioned observations, one is naturally led to the idea that whenever a proximity function is used in the algorithm, then one should adapt the search direction correspondingly. In the next section we introduce the class of SR functions that are developed based on the previous discussions and they will be considered for further developments in this thesis.

2.6.1 Self-Regular Functions

Let us first introduce the class of the so called *Self-Regular* functions and present some of their properties for future use. Then we describe what are the SR-IPMs and we will highlight the major differences between this approach and the classical approach that was discussed in the previous sections.

The class of Self-Regular functions was introduced by Peng et al. [41, 42] as follows.

Definition 2.6.1 *A twice continuously differentiable function $\psi(t) : (0, \infty) \to \mathcal{R}$ is SR if it satisfies the following two conditions:*

SR.1 The function $\psi(t)$ is strictly convex with respect to $t > 0$ and vanishes at its global minimal point $t = 1$, i.e., $\psi(1) = \psi'(1) = 0$. Further, there exist positive constants $\nu_2 \geq \nu_1 > 0$ and $p \geq 1,\ q \geq 1$ such that

$$\nu_1(t^{p-1} + t^{-1-q}) \leq \psi''(t) \leq \nu_2(t^{p-1} + t^{-1-q}), \quad \forall t \in (0, \infty); \quad (2.10)$$

SR.2 For any $t_1, t_2 > 0$,

$$\psi(t_1^r t_2^{1-r}) \leq r\psi(t_1) + (1 - r)\psi(t_2), \quad \forall r \in [0, 1]. \quad (2.11)$$

If $\psi(t)$ is SR, then the parameter q is called the *barrier degree* and parameter p is called the *growth degree* of the SR function $\psi(t)$.

There are two popular families of SR functions. The first family is given by (Figure 2.4)

$$\Upsilon_{p,q}(t) = \frac{t^{p+1}-1}{p(p+1)} + \frac{t^{1-q}-1}{q(q-1)} + \frac{p-q}{pq}(t-1), \quad p \geq 1,\ q > 1, \tag{2.12}$$

with $\nu_1 = \nu_2 = 1$. One can easily derive it by integrating twice the inequalities in (2.10). The second family which is a slight modification of (2.12) is

$$\Gamma_{p,q}(t) = \frac{t^{p+1}-1}{p+1} + \frac{t^{1-q}-1}{q-1}, \quad p \geq 1,\ q > 1, \tag{2.13}$$

with $\nu_1 = \min(p,q)$ and $\nu_2 = \max(p,q)$. For $p, q = 1$ in both cases the classical logarithmic barrier function

$$\Gamma_{11}(t) = \Upsilon_{11}(t) = \frac{t^2-1}{2} - \log t$$

is obtained.

Now, let us present some of the fundamental properties of SR functions. For simplicity here we present only some of the proofs for lemmas and theorems, the interested reader can consult [42] for more details.

Proposition 2.6.1 *If the functions ψ_1 and ψ_2 are SR functions, then any conic combination $\beta_1\psi_1 + \psi_2\beta_2$, where $\beta_1, \beta_2 \geq 0,\ \beta_1 + \beta_2 > 0$ is also SR.*

Let us denote by Ω_1 and Ω_2 the set of functions that satisfy SR.1 and SR.2, respectively.

Lemma 2.6.1 *If $\psi(t) = \psi(t^{-1})$ and $\psi(t) \in \Omega_1$, then $\psi(t)$ is SR.*

The following lemma gives a different characterization of condition SR.2. It has been used for further derivation of new kernel functions in [6, 7].

Lemma 2.6.2 *A function $\psi(t)$ belongs to Ω_2 if and only if the function $\psi(\exp(\zeta))$ is convex in ζ or, equivalently if $\psi^{'}(t) + t\psi^{''}(t) \geq 0$ for $t > 0$.*

Lemma 2.6.3 *Suppose that $\psi(t) \in \Omega_2$. Then for any $\alpha \in \mathcal{R}$, $\psi(t^{\alpha}) \in \Omega_2$.*

The following lemma is used to prove some fundamental results in the sequel which are key elements in the analysis of our new algorithms.

Lemma 2.6.4 *Suppose that $\psi(t) \in \Omega_1$. Then*

$$\begin{aligned} \frac{(t-1)^2}{2} &\leq \frac{\psi(t)}{\nu_1}, \\ \frac{t^{p+1}-1}{p(p+1)} - \frac{t-1}{p} &\leq \frac{\psi(t)}{\nu_1}, \\ \frac{t^{1-q}-1}{q(q-1)} - \frac{t-1}{q} &\leq \frac{\psi(t)}{\nu_1}, \\ \psi(t) &\leq \frac{\psi^{'}(t)^2}{2\nu_1}. \end{aligned}$$

The next lemma gives some lower bounds for $\left|\psi^{'}(t)\right|$ that can also be interpreted as a barrier behavior of the function $\psi^{'}(t)$ and $\psi(t)$. These bounds will be used in the proof of some further results.

Lemma 2.6.5 *Suppose that $\psi(t) \in \Omega_1$. Then*

$$\left|\psi^{'}(t)\right| \geq \frac{\nu_1(t^{-q}-1)}{q}, \ \forall t < 1$$

and

$$\left|\psi^{'}(t)\right| \geq \frac{\nu_1(t^{p}-1)}{p}, \ \forall t > 1.$$

Proof: From condition SR.1, whenever $t < 1$, one has

$$\psi^{'}(t) = \int_1^t \psi^{''}(\zeta)d\zeta \leq \nu_1 \int_1^t (\zeta^{p-1} + \zeta^{-1-q})d\zeta \leq \nu_1 \int_1^t \zeta^{-1-q}d\zeta = \frac{\nu_1}{q}(1-t^{-q}).$$

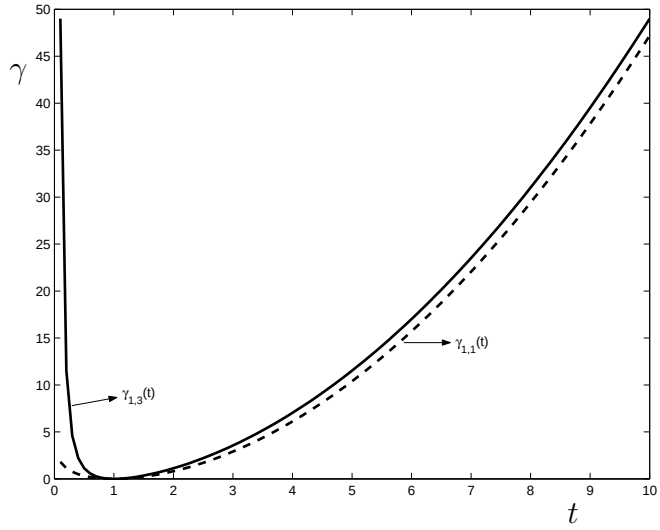

Figure 2.4: Illustration of self-regular functions

This gives the first inequality of the lemma. One can prove the second inequality analogously. □

There are many other significant properties that we do not present in this thesis. The interested reader can consult [42] for more properties.

2.6.2 Self-Regular Proximity Measures

Definition 2.6.2 *A SR-proximity measure* $\Psi(v) : \mathcal{R}^n_{++} \to \mathcal{R}_+$ *is defined by*

$$\Psi(v) := \sum_{i=1}^{n} \psi(v_i), \tag{2.14}$$

where $\psi(\cdot)$ *is SR.*

For notational convenience we use

$$\sigma = \|\nabla\Psi(v)\|. \tag{2.15}$$

The following result establishes a relation between the proximity measure and its gradient norm, and gives lower bounds for the smallest and the largest components of a given vector v. This result will be used frequently in the analysis of our new algorithms.

Proposition 2.6.2 *Let the SR proximity measure $\Psi(v)$ be defined by (2.14). Then*

$$\begin{aligned} \Psi(v) &\leq \frac{\sigma^2}{2\nu_1}, \\ v_{\min} &\geq \left(1 + \frac{q\sigma}{\nu_1}\right)^{\frac{-1}{q}} \\ v_{\max} &\leq \left(1 + \frac{p\sigma}{\nu_1}\right)^{\frac{1}{p}}. \end{aligned}$$

Proof: The first statement follows from the last statement of Lemma 2.6.4. To prove the second inequality, we first note that it holds trivially if $v_{\min} \geq 1$. If $v_{\min} < 1$, then from Lemma 2.6.5 one has

$$\sigma \geq \left|\psi^{'}(v_{\min})\right| \geq \frac{\nu_1}{q}\left(\frac{1}{v_{\min}^q} - 1\right),$$

which implies the second statement of proposition. The last statement can be proved analogously. □

The following theorem gives lower and upper bounds for the smaller and larger components of the vector v, respectively when one uses (2.13) with $p = 1$ as the kernel function in the SR-Proximity measure given by (2.14). These bounds are much better than the bounds in the previous proposition and lead to better results in our development.

Theorem 2.6.1 *Let (2.13) be as the kernel function with $p = 1$ and $q \geq 1$ in the definition of the SR-proximity measure given by (2.14). Then*

$$\begin{aligned} v_{\min} &\geq (1+\sigma)^{\frac{-1}{q}}, \\ v_{\max} &\leq (1+\sigma). \end{aligned}$$

Proof: The statements are trivial when $v_{\min} \geq 1$ and $v_{\max} < 1$. Now let us assume that $v_{\min} < 1$. Then from (2.15) one has

$$\sigma = \left\| v - v^{-q} \right\| \geq \left| v_i^{-q} - v_i \right|, \ \forall i = 1, \cdots, n.$$

This implies

$$\sigma \geq \frac{1}{v_{\min}^q} - v_{\min} \geq \frac{1}{v_{\min}^q} - 1,$$

which implies the first statement of theorem. Now let us consider $v_{\max} > 1$. Then analogous to the previous case one has

$$\sigma \geq v_{\max} - \frac{1}{v_{\max}^q} \geq v_{\max} - 1,$$

which completes the proof of the second statement and finally the proof of theorem. □

2.6.3 Self-Regular Search Directions

In this section we introduce the SR-proximity based search directions. In classical primal-dual IPMs, as we already discussed, one needs to solve (2.8), while in SR-IPMs the Newton system (2.8) is modified. Given a strictly feasible[3]

[3]The infeasible case is discussed in Chapter 4.

point (x, y, s), the Newton system for SR-IPMs for LO is:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mu v \nabla \Psi(v) \end{bmatrix}, \tag{2.16}$$

where $v\nabla\Psi(v) = (v_1\nabla\psi(v_1), \cdots, v_n\nabla\psi(v_n))^T$. Using the notation we introduced in Section 2.5, the Newton system (2.16) can be written as

$$\begin{aligned} \bar{A}d_x &= 0, \\ \bar{A}^T\Delta y + d_s &= 0, \\ d_x + d_s &= -\nabla\Psi(v), \end{aligned} \tag{2.17}$$

where $\bar{A} = \frac{1}{\mu}AV^{-1}X$, and $V^{-1} = \text{diag}\,(v^{-1})$. This representation makes it clear that the SR search direction is nothing else than the projected steepest descent direction for the SR-proximity function.

Therefore, the differences between the classical primal-dual IPMs and the SR-proximity based IPMs are both in the proximity measure used and the Newton system they solve. The most important consequence of using SR-proximity based IPMs is the significant reduction in the iteration bound of the algorithms that are discussed in the sequel.

Small-update SR-proximity based IPMs have an $\mathcal{O}(\sqrt{n}\log\frac{n}{\epsilon})$ worst case iteration bound while large update IPMs have an $\mathcal{O}(qn^{\frac{q+1}{2q}}\log\frac{n}{\epsilon})$ iteration bound. For $q = \frac{\log n}{2}$ they have an $\mathcal{O}(\sqrt{n}\log n\log\frac{n}{\epsilon})$ iteration bound [42], which is the best among all large-update algorithms. This improvement reduces the discrepancy between theory and practice while preserving the practical efficiency [73] of IPMs.

Chapter 3

An Adaptive Self-Regular IPM

In this chapter we propose an adaptive updating scheme algorithm based on a specific family of SR-proximity measures. First we prove some interesting properties of the SR-proximity functions w.r.t. the barrier parameter. Using these properties we define the SR neighborhood that we aim the algorithm to operate n. Then, motivated by practical observations we outline our new algorithm. Finally, its worst case iteration complexity is discussed. The iteration complexity also matches the best iteration complexity obtained so far by using the idea of self-regularity. The results we present in this chapter mostly appeared in [52].

3.1 Proximity Measures and Their Properties

Throughout this chapter we deal with the following family of SR-proximity measures

$$\Phi_q(x, s, \mu) := \Psi_q(v) = \frac{e^T v^2 - n}{2} + \frac{e^T v^{1-q} - n}{q - 1}, \tag{3.1}$$

where $q > 1$. This family of SR-proximity measures using (2.14) are induced by the kernel functions $\Gamma_{1q}(t)$ given by (2.13). Let the current iterate be (x, s), and let $\mu_g := \frac{x^T s}{n}$ denote the parameter value associated with the current duality gap. Next, we consider the behavior of the function $\Phi_q(x, s, \mu)$ w.r.t. μ for a given q. The following result gives the global minimum of the proximity measure.

Proposition 3.1.1 *For any fixed $(x, s) > 0$ and $q > 1$ the proximity function $\Phi_q(x, s, \mu)$, as a function of μ, has a global minimizer at*

$$\mu_q^* = \left(\frac{x^T s}{(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}} \right)^{\frac{2}{q+1}}.$$

Proof: It can be easily proved that for any $q \geq 3$ the proximity function is strictly convex and for any $1 < q < 3$ the proximity function is quasi-convex. Using the strict and quasi-convexity of the proximity function, one can easily show that μ_q^* is the global minimizer of the proximity function. □

In [44], for the special case $q = 3$, the authors used the property $\Phi_q(x, s, \mu_g) = \Phi_q(x, s, \mu_h)$ to simplify the analysis, where μ_h is the harmonic mean of the components of the vectors x and s. In this chapter, for the general case, instead of μ_h we use

$$\mu_q^h = \left(\frac{n}{(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}} \right)^{\frac{2}{q-1}},$$

as a generalized harmonic mean. Unfortunately, in general the relation $\Phi_q(x, s, \mu_g) = \Phi_q(x, s, \mu_q^h)$ does not hold.

The following lemma plays an important role in the definition of SR-proximity based neighborhoods.

Lemma 3.1.1 *For any $\tau \geq 2$, the following statements are equivalent.*
1) $\dfrac{\mu_g}{\mu_q^h} \leq \tau$,
2) $\Phi_q\left(x, s, \dfrac{\mu_g}{\tau}\right) \leq \dfrac{(\tau-1)n}{2}$,
3) $\Phi_q(x, s, \mu_g) \leq \dfrac{\left(\tau^{\frac{q-1}{2}} - 1\right) n}{q-1}$.

Proof: By the assumption of the lemma we can write $\mu_g = \bar{\tau}\mu_q^h$ for some $\bar{\tau} \leq \tau$. It follows that

$$\begin{aligned}\Phi_q\left(x, s, \frac{\mu_g}{\tau}\right) &= \frac{(\tau-1)n}{2} + \frac{\left(\frac{\mu_g}{\tau}\right)^{\frac{q-1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - n}{q-1} \\ &\leq \frac{(\tau-1)n}{2} + \frac{\left((\frac{\bar{\tau}}{\tau})^{\frac{q-1}{2}} - 1\right) n}{q-1} \leq \frac{(\tau-1)n}{2}.\end{aligned}$$

This completes the proof of implication 1) $\Longrightarrow$ 2). One can easily prove the implication 2) $\Longrightarrow$ 1). For implication 1) $\Longrightarrow$ 3) we have

$$\Phi_q(x, s, \mu_g) = \frac{\mu_g^{\frac{q-1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - n}{q-1} \leq \frac{\left(\tau^{\frac{q-1}{2}} - 1\right) n}{q-1}.$$

One can prove analogously implication 3) $\Longrightarrow$ 1), which completes the proof. □

Using Lemma 3.1.1, a SR-proximity based neighborhood of the central path is defined by

$$\mathcal{N}_q(\tau, n) := \{(x, y, s) \in \mathcal{F} \mid (x, s) > 0,\ \Phi_q(x, s, \mu) \leq \eta_q(n, \tau)\}, \tag{3.2}$$

where

$$\eta_q(n, \tau) = \frac{(\tau^{\frac{q-1}{2}} - 1)n}{q-1}.$$

The following crucial lemma determines the relation between μ_g and μ_q^h.

Lemma 3.1.2 *Let the current iterate $(x, y, s) \in \mathcal{N}_q(\tau, n)$, then*

$$\mu_q^h \leq \mu_g.$$

Proof: To prove $\mu_q^h \leq \mu_g$ it suffices to prove

$$(x^T s)^{\frac{q-1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} \geq n^{\frac{q+1}{2}}. \tag{3.3}$$

By the arithmetic-geometric mean inequality we have

$$x^T s \geq n \left(\prod_{i=1}^{n} x_i s_i \right)^{\frac{1}{n}}.$$

Then

$$(x^T s)^{\frac{q-1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} \geq n^{\frac{q-1}{2}} \left(\prod_{i=1}^{n} x_i s_i \right)^{\frac{q-1}{2n}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} \geq n^{\frac{q+1}{2}},$$

where the last inequality follows from the arithmetic-geometric mean inequality for $(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}$. This proves (3.3) and therefore the statement of the lemma. □

Now we proceed to discuss the properties of SR search directions for different updates of μ. Note that, due to the specific choice of the kernel function $\psi(t)$, we can rewrite system (2.16) in the original space ($(\Delta x, \Delta y, \Delta s)$ space) as:

$$\begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu^{\frac{q+1}{2}} (x^{\frac{1-q}{2}}) s^{\frac{1-q}{2}} - xs. \end{aligned} \tag{3.4}$$

Let us denote the solution of system (3.4) by $(\Delta x(\mu), \Delta y(\mu), \Delta s(\mu))$. The following two lemmas discuss the change of the duality gap along the search direction $(\Delta x(\mu), \Delta y(\mu), \Delta s(\mu))$ for $\mu = \mu_q^*$, and for $\mu = \mu_q^h$.

Lemma 3.1.3 *Let $(\Delta x(\mu_q^*), \Delta y(\mu_q^*), \Delta s(\mu_q^*))$ be the solution of system (3.4) with $\mu = \mu_q^*$. Then the relation*

$$x^T \Delta s(\mu_q^*) + s^T \Delta x(\mu_q^*) = 0$$

holds.

Proof: From the third equation of (3.4) we have

$$x^T \Delta s(\mu_q^*) + s^T \Delta x(\mu_q^*) = (\mu_q^*)^{\frac{q+1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - x^T s = 0.$$

This completes the proof. □

The following corollary shows that if the current target is μ_q^*, then the duality gap of the new iterate is the same as the one for the previous iterate.

Corollary 3.1.1 *If the targeted μ parameter is μ_q^*, then the duality gap will not change for any feasible step size α, i.e.,*

$$(x + \alpha \Delta x(\mu_q^*))^T (s + \alpha \Delta s(\mu_q^*)) = x^T s.$$

Lemma 3.1.4 *Let $(\Delta x(\mu_q^h), \Delta y(\mu_q^h), \Delta s(\mu_q^h))$ be the solution of system (3.4) with $\mu = \mu_q^h$. Then the relation*

$$x^T \Delta s(\mu_q^h) + s^T \Delta x(\mu_q^h) = n\mu_q^h - x^T s$$

holds.

Proof: Using the third equation in (3.4) with $\mu = \mu_q^h$ we have

$$s^T \Delta x + x^T \Delta s = (\mu_q^h)^{\frac{q+1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - x^T s = n\mu_q^h - x^T s$$

that completes the proof. □

Corollary 3.1.2 *If the targeted μ parameter is μ_q^h, then the search direction based on our specific SR-proximity function and the standard Newton direction will predict the change of the duality gap in the same way.*

Since we are working with a large neighborhood, we define $\frac{(\tau-1)n}{2}$ as the maximum allowed value of the proximity function w.r.t. μ_q^t, the target μ value. One can see that $\Phi_q(x,s,\mu_q^t)=\frac{(\tau-1)n}{2}$ if and only if μ_q^t satisfies the equation

$$2(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} \mu^{\frac{q+1}{2}} - (2n+\tau(q-1)n)\,\mu + (q-1)x^T s = 0. \tag{3.5}$$

When $\frac{\mu_g}{\mu_q^h} \leq \tau$, this equation has two positive roots. One is less than or equal to μ_q^* and the other is larger than or equal to μ_q^*. In the algorithm we will use the smaller positive root μ_q^t as the target value at each iteration. The following lemma describes the relation between μ_q^t and μ_q^h and μ_q^*.

Lemma 3.1.5 *For any $(x,s) \in \mathcal{N}_q(\tau,n)$ one has*
$1: \mu_q^h \leq \mu_q^*$.
$2: \mu_q^t \leq \mu_q^h$ *where the equality holds when* $\mu_g = \tau\mu_q^h$.

Proof: By definition $\mu_q^* = \left(\mu_g(\mu_q^h)^{\frac{q-1}{2}}\right)^{\frac{2}{q+1}}$. Since for any point in $\mathcal{N}_q(\tau,n)$ by Lemma 3.1.2 one has $\mu_g \geq \mu_q^h$, thus we have proved the first statement of the Lemma. For the second statement, when $\mu_g = \tau\mu_q^h$, from equation (3.5) one has $\mu_q^t = \mu_q^h$. Now let us assume that $\mu_g = \tau_1\mu_q^h < \tau\mu_q^h$. By the definition of the proximity measure we have $\Phi(x,s,\mu_q^h) = \frac{(\tau_1-1)n}{2} < \frac{(\tau-1)n}{2}$. By the quasiconvexity property of the proximity measure w.r.t. μ we have $\mu_q^t < \mu_q^h$. This completes the proof of the second statement and therefore the proof of the lemma. □

The following lemma gives a useful relation between μ_g and μ_q^t which is used frequently in the sequel.

Lemma 3.1.6 *Let μ_q^t be the smaller positive root of equation (3.5). Then*

$$\tau\mu_q^t \leq \mu_g \leq 2\tau\mu_q^t.$$

Proof: If $\mu_g = \tau\mu_q^h$, then $\mu_q^h = \mu_q^t$, thus we are done. If $\mu_g < \tau\mu_q^h$, then the value of the proximity measure at $\mu = \frac{\mu_g}{\tau}$ is

$$\Phi\left(x, s, \frac{\mu_g}{\tau}\right) = \frac{(\tau-1)n}{2} + \frac{\left(\left(\frac{\mu_g}{\tau\mu_q^h}\right)^{\frac{q-1}{2}} - 1\right)n}{q-1}.$$

Since $(x,s) \in \mathcal{N}_q(\tau, q)$, then $\Phi(x, s, \frac{\mu_g}{\tau}) \leq \frac{(\tau-1)n}{2}$. Due to the quasiconvexity of the proximity measure w.r.t. μ one has $\tau\mu_q^t \leq \mu_g$. This proves the left hand side inequality. For the right hand side inequality for the case $q \geq 3$ using (3.5) one has $\mu_g \leq (\tau+1)\mu_q^t$. Now let us assume $1 < q < 3$. For the current iterate $(x,s) \in \mathcal{N}_q(\tau, q)$, one can assume that $\mu_g = \tau_1\mu_q^h$, where $1 \leq \tau_1 \leq \tau$. Therefore, one may write equation (3.5) as

$$2\left(\frac{\tau_1\mu_q^t}{\mu_g}\right)^{\frac{q-1}{2}} + (q-1)\frac{\mu_g}{\mu_q^t} - 2 - (q-1)\tau = 0.$$

Let us assume that the left hand side function is a function of $\dfrac{\mu_q^t}{\mu_g}$ and we denote it by $h\left(\dfrac{\mu_q^t}{\mu_g}\right)$. Then we have $h'(\tau_1^{\frac{1-q}{q+1}}) = 0$, $h'(\frac{\mu_q^t}{\mu_g}) < 0$ for $\frac{\mu_q^t}{\mu_g} < \tau_1^{\frac{1-q}{q+1}}$ and $h'(\frac{\mu_q^t}{\mu_g}) > 0$ for $\frac{\mu_q^t}{\mu_g} > \tau_1^{\frac{1-q}{q+1}}$. One also has

$$\begin{aligned} h(\tau_1^{\frac{1-q}{q+1}}) &= 2\tau_1^{\frac{q-1}{q+1}} + (q-1)\tau_1^{\frac{q-1}{q+1}} - 2 - \tau(q-1) \\ &\leq 2\tau^{\frac{q-1}{q+1}} + (q-1)\tau^{\frac{q-1}{q+1}} - 2 - \tau(q-1) \\ &= (q+1)\tau^{\frac{q-1}{q+1}} - 2 - \tau(q-1) := g(q). \end{aligned}$$

Now the goal is to show that $g(q) < 0$. To do so, one has

$$g'(q) = \tau^{\frac{q-1}{q+1}}\left(1 + \frac{2}{q+1}\log\tau\right) - \tau,$$

which is negative[1] for all $1 \leq q < 3$ and $g(1) = 0$. Therefore, $g(q) < 0$ for all $1 < q < 3$. This proves that $h(\tau_1^{\frac{1-q}{q+1}}) < 0$. To prove the right hand side inequality of the lemma for $1 < q < 3$, it suffices to prove that $h\left(\frac{1}{2\tau}\right) \geq 0$ that follows[2]:

$$h\left(\frac{1}{2\tau}\right) = 2\left(\frac{\tau_1}{2\tau}\right)^{\frac{q-1}{2}} + \tau(q-1) - 2 \geq 2\left(\frac{1}{2\tau}\right)^{\frac{q-1}{2}} + \tau(q-1) - 2 \geq 0.$$

Therefore h has a root greater or equal than $\frac{1}{2\tau}$ because $\frac{1}{2\tau} < \tau_1^{\frac{1-q}{q+1}}$. □

When μ_q^t is the target value, by analogy to Lemma 3.1.3, we can get the following result.

Lemma 3.1.7 *Let $(\Delta x(\mu_q^t), \Delta y(\mu_q^t), \Delta s(\mu_q^t))$ be the solution of system (3.4), where $\mu = \mu_q^t$ is the smaller positive root of equation (3.5). Then one has*

$$x^T \Delta s(\mu_q^t) + s^T \Delta x(\mu_q^t) = (\mu_q^t)^{\frac{q+1}{2}} (x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - x^T s.$$

Therefore, if the targeted parameter is μ_q^t, by Lemma 3.1.7 we have

$$(x + \alpha \Delta x(\mu_q^t))^T (s + \alpha \Delta s(\mu_q^t)) = x^T s \left(1 - \alpha + \frac{(\mu_q^t)^{\frac{q+1}{2}} \alpha}{\mu_g (\mu_q^h)^{\frac{q-1}{2}}}\right). \tag{3.6}$$

Remark 3.1.1 *If $\mu_q^t \sim \mu_q^h$, then (3.6) implies that the search direction based on our specific SR-proximity function and the standard Newton direction will predict the change of the duality gap almost in the same way (If $\mu_q^t = \mu_q^h$, see Corollary 3.1.2). But if $\mu_q^t \ll \mu_q^h$, then the ratio $\dfrac{(\mu_q^t)^{\frac{q+1}{2}}}{\mu_g (\mu_q^h)^{\frac{q-1}{2}}}$ is very small and for the SR search direction the duality gap reduction is much larger than it would be when using the standard Newton direction.*

[1] It follows from the fact that $1 + \log x - x$ is strictly decreasing for $x \geq 1$.

[2] Simple differentiation shows that the last inequality holds for $1 \leq q < 3$ and $\tau \geq 2$.

3.2 Algorithmic Scheme

In this section we consider a specific variant of Algorithm 2.4.4. This variant is more flexible in updating μ and is closer to what is implemented in the McIPM solver [73] than Algorithm 2.4.4, mainly because we use a large-update at each iteration and we do not employ any inner iterations to recenter.

To motivate our design, let us start by considering an implementational issue in the algorithm. Suppose that the present point (x, s) is in a certain neighborhood of the central path. Then we solve the linear system (3.4) for the search direction $(\Delta x, \Delta y, \Delta s)$, from which we can estimate the maximal feasible step size $\alpha_q^{\max}$. A popular heuristic for choosing the step size in IPM solvers is to use a damping factor to $\alpha_q^{\max}$, say $0.995\alpha_q^{\max}$, as a step size. Of course, if the value of the corresponding proximity function is too large for this step size, then we should reduce the step size appropriately so that the value of the proximity function at the new iterate is below a prescribed threshold. Note that it is also possible that the proximity function has a relatively small value for the step size $0.995\alpha_q^{\max}$. In this case, theoretically we can still increase the step size so that the value of the proximity function remains below a prescribed bound. However, in practice this might not be a good idea. Since the step size is already quite close to the maximal feasible step size, even a small increase of the step size may cause numerical problems or drive the iterate too close to the boundary of the feasible region. In this situation, it is better to use the default value $0.995\alpha_q^{\max}$ as the step size. Note that after such a step, we have an iterate (x, s) with a small proximity function value. Motivated by the above observation, we change the procedure of Algorithm 2.4.4. Further, we utilize a parameter $\tau \geq 2$ to keep control on the distance of the iterate to

the central path. Recall the results of Lemma 3.1.1 that quantify the relation between μ_g, μ_q^h and the corresponding proximity values. As specified in (3.2), the definition of the neighborhood $\mathcal{N}_q(\tau, n)$, we force the value of the proximity function (3.1) to satisfy the relation:

$$\Phi_q(x, s, \mu_g) \leq \eta_q(n, \tau). \tag{3.7}$$

In our algorithm, regardless of whether the iterate is close to, or is far away from the central path, we always make a large update of the central path parameter μ.

Remark 3.2.1 *The assumption $\tau \geq 2$ is necessary to keep $\sigma_q = \|v - v^{-q}\| \geq 1$, which is a helpful property in the complexity analysis of our new algorithm.*

For simplicity we also define

$$\mu_g(\alpha) = \frac{x(\alpha)^T s(\alpha)}{n}, \quad \mu_q^h(\alpha) = \left(\frac{n}{(x(\alpha)^{\frac{1-q}{2}})^T s(\alpha)^{\frac{1-q}{2}}} \right)^{\frac{2}{q-1}}$$

and

$$\mu_q^*(\alpha) = \left(\frac{x(\alpha)^T s(\alpha)}{(x(\alpha)^{\frac{1-q}{2}})^T s(\alpha)^{\frac{1-q}{2}}} \right)^{\frac{2}{q+1}}.$$

At each step, we stipulate that the step size should be chosen so that the proximity function $\Phi_q(x(\alpha), s(\alpha), \mu_+)$ has a sufficient decrease while the proximity function w.r.t. $\mu_g(\alpha)$ still satisfies (3.7) at the new iterate. The new algorithm is presented in the scheme Algorithm 3.2.1.

Algorithm 3.2.1: An Adaptive Large-Update SR-IPM

Input:

A proximity parameter $\tau \geq 2$;

an accuracy parameter $\epsilon > 0$;

$(x, s) = (x^0, s^0) \in \mathcal{F}^0$ such that $\frac{\mu_q}{\mu_q^h} \leq \tau$.

begin
while $x^T s \geq \epsilon$ **do**

begin
Solve system (3.4) with $\mu = \mu_q^t$ for $\Delta x, \Delta y, \Delta s$.

begin
Determine the maximum step size $\bar{\alpha}$ such that

$(x + \bar{\alpha}\Delta x, s + \bar{\alpha}\Delta s) \geq 0$;

Let[a]

$\alpha = \max\left\{\alpha \in (0\ \bar{\alpha}] \mid \Phi_q^+ \leq \Phi_q - \frac{\alpha_q^* \sigma_q^2}{4}, (x(\bar{\alpha}), y(\bar{\alpha}), s(\bar{\alpha})) \in \mathcal{N}_q(\tau, n)\right\}$;

Set $x := x(\alpha);\ y := y(\alpha);\ s := s(\alpha)$.
end
end
end

---

[a] $\Phi_q^+ = \Phi_q(x(\alpha), s(\alpha), \mu)$ and $\Phi_q = \Phi_q(x, s, \mu)$.

Remark 3.2.2 *At each iteration, the step size $\bar{\alpha}$ has to be chosen such that the proximity function $\Phi_q(x, s, \mu)$ decreases sufficiently. In the sequel we present a default value for $\bar{\alpha}$ based on the actual value of the proximity function.*

Remark 3.2.3 *For a practical implementation, it would be advantageous to choose the step size $\bar{\alpha}$ so that it minimizes the proximity function*

$\Phi_q(x(\bar{\alpha}), s(\bar{\alpha}), \mu_q^t)$, *while the constraint* $\mu_g(\bar{\alpha}) \leq \tau\mu_q^h(\bar{\alpha})$ *is satisfied. However, this would require an exact line search. Instead of this, in the analysis of Algorithm 3.2.1 we solve the line search problem approximately so that the value of the function* $\Phi_q(x(\alpha), s(\alpha), \mu_t)$ *is decreased sufficiently. Theorem 3.3.1 gives a default value for such a step size.*

3.3 Complexity Analysis

We proceed to analyzing the complexity of Algorithm 3.2.1. The key element of the analysis is to estimate the value of the step size α used in the algorithm that implies sufficient reduction of μ_g, while the new iterate stays in $\mathcal{N}_q(\tau, n)$. This is done in the following subsection.

3.3.1 Estimating the Step Size

In this section, first we give a lower bound for the maximal feasible step size and then we discuss the decreasing behavior of the proximity function $\Phi_q(x(\alpha), s(\alpha), \mu_q^t)$.

Lemma 3.3.1 *Let* $(\Delta x, \Delta y, \Delta s)$ *be the solution of system (3.4) where* $\mu = \mu_q^t$ *is defined by equation (3.5). Then the maximal feasible step size,* $\alpha_q^{\max}$*, satisfies*

$$\alpha_q^{\max} \geq \bar{\alpha}_q = \frac{1}{\sigma_q(1+\sigma_q)^{\frac{1}{q}}}.$$

Proof: We know that

$$v(\alpha_q^{\max}) = (v + \alpha_q^{\max} d_x)^{\frac{1}{2}}(v + \alpha_q^{\max} d_s)^{\frac{1}{2}} = v(e + \alpha_q^{\max} v^{-1} d_x)^{\frac{1}{2}}(e + \alpha_q^{\max} v^{-1} d_s)^{\frac{1}{2}}$$

is nonnegative if

$$e + \alpha_q^{\max} v^{-1} d_x \geq 0, \text{ and } \quad e + \alpha_q^{\max} v^{-1} d_x \geq 0.$$

These inequalities imply

$$\alpha_q^{\max} \geq \frac{1}{\|(v^{-1}d_x, v^{-1}d_s)\|}.$$

We also know that

$$\left\|(v^{-1}d_x, v^{-1}d_s)\right\| \leq \frac{\|(d_x, d_s)\|}{v_{\min}} \leq \sigma_q(1+\sigma_q)^{\frac{1}{q}},$$

where the last inequality follows from the second statement of Theorem 2.6.1. This completes the proof. □

Theorem 3.3.1 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (3.4), where $\mu = \mu_q^t$ is defined by equation (3.5). Then for the step size $\hat{a}_q = \dfrac{\bar{\alpha}_q}{3q}$, the relation*

$$\Phi_q(x(\hat{\alpha}_q), s(\hat{\alpha}_q), \mu_q^t) \leq \Phi_q(x, s, \mu_q^t) - \frac{2^{\frac{q-1}{2q}}\Phi_q(x, s, \mu)^{\frac{q-1}{q}}}{24q}$$

holds.

Proof: Let

$$\begin{aligned} h(\alpha) : &= \Phi_q\left(x(\alpha), s(\alpha), \mu_q^t\right) - \Phi_q(x, s, \mu_q^t) \\ &= \frac{\|v(\alpha)\|^2 - n}{2} + \frac{\left\|v(\alpha)^{\frac{q-1}{2}}\right\|^2 - n}{q-1} - \frac{\|v\|^2 - n}{2} - \frac{\left\|v^{\frac{q-1}{2}}\right\|^2 - n}{q-1}, \end{aligned}$$

where $v(\alpha) = \sqrt{\dfrac{x(\alpha)s(\alpha)}{\mu_q^t}} = (v + \alpha d_x)^{\frac{1}{2}}(v + \alpha d_s)^{\frac{1}{2}}$. One can easily check that $\left\|v(\alpha)^{\frac{1-q}{2}}\right\|^2$ satisfies the condition SR.2 of Definition 2.6.1, then we have

$$\begin{aligned} h(\alpha) \leq & \frac{1}{2}v^T(d_x + d_s)\alpha + \frac{1}{2(q-1)}\sum_{i=1}^{n}(v_i + \alpha(d_x)_i)^{1-q} \\ & + \frac{1}{2(q-1)}\sum_{i=1}^{n}(v_i + \alpha(d_s)_i)^{1-q} - \frac{1}{q-1}\left\|v^{\frac{1-q}{2}}\right\|^2 := h_1(\alpha). \end{aligned}$$

It can be easily shown that

$$h_1(0) = 0, \quad h_1^{'}(0) = -\frac{\sigma_q^2}{2}, \quad \text{and} \quad h_1^{''}(\alpha) \leq \frac{q\sigma_q^2}{2}(v_{\min} - \alpha\sigma_q)^{-1-q}.$$

Then using the fact that $h_1(\alpha) = h_1(0) + h_1^{'}(0)\alpha + \int_0^{\alpha}\int_0^{\zeta} h_1^{''}(\eta)d\eta d\zeta$ one has

$$h(\alpha) \leq -\frac{\alpha\sigma_q^2}{2} + \frac{q\sigma_q^2}{2}\int_0^{\alpha}\int_0^{\zeta}(v_{\min} - \eta\sigma_q)^{-1-q}d\eta d\zeta := h_2(\alpha).$$

Clearly $h_2(\alpha)$ is a convex function and twice differentiable in the interval $[0, \bar{\alpha}_q)$. Let us denote by $\breve{\alpha}_q$ the global minimum of $h_2(\alpha)$ in the interval $[0, \bar{\alpha}_q)$. Then $\breve{\alpha}_q$ is the unique solution of the equation

$$-\sigma_q^2 + \sigma_q\left((v_{\min} - \alpha\sigma_q)^{-q} - (v_{\min})^{-q}\right) = 0.$$

After simplifying we have

$$\breve{\alpha}_q = \frac{v_{\min}}{\sigma_q}\left[1 - (1 + v_{\min}^q\sigma_q)^{-\frac{1}{q}}\right] = \frac{v_{\min}}{\sigma_q}\left[1 - \left(1 - \frac{v_{\min}^q\sigma_q}{1 + v_{\min}^q\sigma_q}\right)^{\frac{1}{q}}\right].$$

Using Lemma 10.1.1 and the second statement of Theorem 2.6.1 one can show that $\breve{\alpha}_q \geq \hat{\alpha}_q := \dfrac{1}{3q\sigma_q(1+\sigma_q)^{\frac{1}{q}}}$. Then by Lemma 10.1.2 we have

$$h(\hat{\alpha}_q) \leq -\frac{\sigma_q^{\frac{q-1}{q}}}{24q}.$$

Finally, by inequality $\sigma_q^2 \geq 2\Phi_q(x, s, \mu)$ of Proposition 2.6.2 we get the desired inequality. □

Now let us discuss under what step size the new iterate stays in the neighborhood $\mathcal{N}_q(\tau, n)$. By Lemma 3.1.1 it suffices to analyze the behavior of $\Phi(x(\alpha), s(\alpha), \mu_g(\alpha))$.

Theorem 3.3.2 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (3.4), where $\mu = \mu_q^t$ is defined by equation (3.5) and $\frac{\mu_g}{\mu_q^h} \leq \left(1 - \frac{1}{3q}\right)^{q+1} \tau$. Then*

$$\Phi_q(x(\hat{\alpha}_q), s(\hat{\alpha}_q), \mu_g(\hat{\alpha}_q)) \leq \eta_q(n, \tau).$$

Proof: It suffices to estimate the interval in which the proximity function satisfies

$$\Phi_q(x(\alpha), s(\alpha), \mu_g(\alpha)) \leq \eta_q(n, \tau).$$

We start by considering the function

$$\begin{aligned} g_1(\alpha) &= \Phi_q(x(\alpha), s(\alpha), \mu_q^*) \\ &= \frac{\mu_q^t}{2\mu_q^*} \|v(\alpha)\|^2 - \frac{n}{2} + \frac{1}{q-1} \left(\frac{\mu_q^*}{\mu_q^t}\right)^{\frac{q-1}{2}} \left\|v(\alpha)^{\frac{1-q}{2}}\right\|^2 - \frac{n}{q-1}, \end{aligned}$$

where $v(\alpha) = \sqrt{\frac{x(\alpha)s(\alpha)}{\mu_q^t}}$. Note that, from the definitions of μ_q^* and μ_q^t, we may derive

$$\frac{\mu_q^t}{\mu_q^*} \|v(0)\|^2 = \left(\frac{\mu_q^*}{\mu_q^t}\right)^{\frac{q-1}{2}} \left\|v(0)^{\frac{1-q}{2}}\right\|^2 = \tau_0^{\frac{q-1}{q+1}} n,$$

where $\tau_0 = \frac{\mu_g}{\mu_q^h}$. On the other hand, by the choice of the search direction we have

$$\|v(\alpha)\|^2 < \|v(0)\|^2.$$

Note that, by using the definitions of μ_q^* and μ_q^t, one can easily prove that

$$\left\|v(\alpha)^{\frac{1-q}{2}}\right\|^2 \leq (1 - \alpha v_{\min}^{-1} \sigma_q)^{1-q} \left\|v^{\frac{1-q}{2}}\right\|^2 \leq \left(1 - \frac{1}{3q}\right)^{1-q} \left\|v^{\frac{1-q}{2}}\right\|^2$$

for all $\alpha \leq \dfrac{1}{3q\sigma_q(1+\sigma_q)^{\frac{1}{q}}}$. Combining the above two inequalities together, we conclude that

$$g_1(\alpha) \leq \frac{\tau_0^{\frac{q-1}{q+1}} n}{2} - \frac{n}{2} + \frac{\left(1 - \frac{1}{3q}\right)^{1-q} \tau_0^{\frac{q-1}{q+1}} n}{q-1} - \frac{n}{q-1} \tag{3.8}$$

for all $\alpha \leq \dfrac{1}{3q\sigma_q(1+\sigma_q)^{\frac{1}{q}}}$. In order to have the next iterate in the SR neighborhood $\mathcal{N}_q(\tau, n)$, it is sufficient to have

$$\frac{\tau_0^{\frac{q-1}{q+1}}n}{2} - \frac{n}{2} + \left(1 - \frac{1}{3q}\right)^{1-q} \frac{\tau_0^{\frac{q-1}{q+1}}n}{q-1} - \frac{n}{q-1} \leq \frac{\tau^{\frac{q-1}{q+1}}n}{2} - \frac{n}{2} + \frac{\tau^{\frac{q-1}{q+1}}n}{q-1} - \frac{n}{q-1}.$$

It is obvious that the previous inequality holds for $\tau_0 \leq (1 - \frac{1}{3q})^{q+1}\tau$. Since $\mu_q^*(\alpha)$ is the global minimum of $\Phi_q(x(\alpha), s(\alpha), \mu)$ w.r.t. μ, from (3.8) we have

$$\Phi_q(x(\alpha), s(\alpha), \mu_q^*(\alpha)) \leq \frac{\tau^{\frac{q-1}{q+1}}n}{2} - \frac{n}{2} + \frac{\tau^{\frac{q-1}{q+1}}n}{q-1} - \frac{n}{q-1}. \tag{3.9}$$

Using (3.9) and the definition of the proximity function $\Psi_q(v)$ we can get $\mu_g(\alpha) \leq \tau\mu_q^h(\alpha)$. This completes the proof of the theorem. □

Now let us discuss the case when $\dfrac{\mu_g}{\mu_q^h} = \tau$.

Theorem 3.3.3 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (3.4) where $\mu = \mu_q^t$ is defined by equation (3.5) and $\dfrac{\mu_g}{\mu_q^h} = \tau$. Then for $\check{\alpha}_q = \dfrac{\hat{\alpha}_q}{\tau}$ the relation*

$$\Phi_q(x(\check{\alpha}_q), s(\check{\alpha}_q), \mu_g(\check{\alpha}_q)) \leq \eta_q(n, \tau).$$

holds.

Proof: From the assumption that $\mu_g = \tau\mu_q^h = \tau\mu_q^t$ one has $\mu_q^* = \tau^{\frac{2}{q+1}}\mu_q^h$. Now, let us define

$$\begin{aligned} g(\alpha) :&= \Phi_q(x(\alpha), s(\alpha), \mu_q^*) - \Phi_q(x, s, \mu_q^*) = \frac{1}{2}\frac{\mu_q^h}{\mu_q^*}v^T(d_x + d_s)\alpha \\ &\quad + \frac{1}{q-1}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}} \left\|v(\alpha)^{\frac{1-q}{2}}\right\|^2 - \frac{1}{q-1}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}} \left\|v^{\frac{1-q}{2}}\right\|^2 \\ &\leq \frac{1}{2}\frac{\mu_q^h}{\mu_q^*}v^T(d_x + d_s)\alpha + \frac{1}{2(q-1)}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}} \left\|(v + \alpha d_x)^{\frac{1-q}{2}}\right\|^2 \end{aligned}$$

$$+\frac{1}{2(q-1)}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}}\left\|(v+\alpha d_s)^{\frac{1-q}{2}}\right\|^2-\frac{1}{q-1}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}}\left\|v^{\frac{1-q}{2}}\right\|^2=g_1(\alpha),$$

where the last inequality follows from the fact that $\left\|v(\alpha)^{\frac{1-q}{2}}\right\|^2$ satisfies the condition SR.2 of Definition 2.6.1. From the definition of v, we have $\left\|v^{\frac{1-q}{2}}\right\|^2 = n$. Using the inequality $\sigma_q^2 \geq 2\Phi(x,s,\mu_q^h) = (\tau-1)n$ of Proposition 2.6.2 further implies

$$\left\|v^{-q}\right\|^2-\left\|v^{\frac{1-q}{2}}\right\|^2\geq 0.$$

This inequality, together with the fact that $\|v\|^2=\tau\left\|v^{\frac{1-q}{2}}\right\|^2$ gives

$$g_1^{'}(0)=\frac{1}{2}\frac{\mu_q^h}{\mu_q^*}v^T(d_x+d_s)-\frac{1}{2}\left(\frac{\mu_q^*}{\mu_q^h}\right)^{\frac{q-1}{2}}(v^{-q})^T(d_x+d_s)\leq-\frac{\sigma_q^2}{2\tau^{\frac{2}{q+1}}}.$$

One also has the following upper bound for the second derivative of $g_1(\alpha)$

$$g_1^{''}(\alpha)\leq\frac{\tau^{\frac{q-1}{q+1}}q\sigma_q^2}{2}(v_{\min}-\alpha\sigma_q)^{-q-1}.$$

Using the facts that

$$g_1(\alpha)=g_1(0)+g_1^{'}(0)\alpha+\int_0^\alpha\int_0^\xi g_1^{''}(\eta)d\eta d\xi$$

and $g_1(0)=0$, one has

$$g(\alpha)\leq-\frac{\sigma_q^2}{2\tau^{\frac{2}{q+1}}}\alpha+\frac{\tau^{\frac{q-1}{q+1}}q\sigma_q^2}{2}\int_0^\alpha\int_0^\xi(v_{\min}-\eta\sigma_q)^{-q-1}d\eta d\xi:=g_2(\alpha).$$

The function $g_2(\alpha)$ is a continuously differentiable function of α. Let us denote its global minimum by $\tilde{\alpha}_q$, then analogous to the proof of Theorem 3.3.1 one can derive the following lower bound for the global minimum:

$$\tilde{\alpha}_q\geq\check{\alpha}_q:=\frac{1}{3\tau q\sigma_q(1+\sigma_q)^{\frac{1}{q}}}.$$

This completes the proof of the theorem. □

Corollary 3.3.1 *By Theorems 3.3.1 and 3.3.3 we call a step good if its step size is at least* $\alpha_q^* = \max\{\hat{\alpha}_q, \check{\alpha}_q\} = \check{\alpha}_q$.

Remark 3.3.1 *As we saw in Theorems 3.3.1 and 3.3.3, in certain parts of the neighborhood* $\mathcal{N}_q(\tau, n)$ *Algorithm 3.2.1 gives a reasonably good reduction of the proximity function that enables us to prove the polynomiality of the algorithm, however it might not give such a good reduction when* $(1 - \frac{1}{3q})^{q+1}\tau < \frac{\mu_g}{\mu_q^h} < \tau$. *Therefore, we call a step good if it falls in one of the cases in Theorem 3.3.1 and 3.3.3, otherwise it is called a bad step. We have to mention that Algorithm 3.2.1 might do at most a bad step after each good step. In the sequel we show that after each possible bad step the algorithm makes a good step again.*

Now let us discuss the case when $\left(1 - \frac{1}{3q}\right)^{q+1} \tau < \frac{\mu_g}{\mu_q^h} < \tau$.

Lemma 3.3.2 *Let* $(\Delta x, \Delta y, \Delta s)$ *be the solution of system (3.4), where* $\mu = \mu_q^t$ *is defined by equation (3.5) and* $\left(1 - \frac{1}{3q}\right)^{q+1} \tau < \frac{\mu_g}{\mu_q^h} < \tau$. *Then, there exists a step size* α *for which the proximity measure decreases with respect to* μ_q^t *and*

$$\Phi_q(x(\alpha), s(\alpha), \mu_g(\alpha)) \leq \eta_q(n, \tau).$$

Proof: As we saw in the proof of Theorem 3.3.2, the only thing that restricts the iterate to proceed is inequality (3.7). We also know by Theorem 3.3.1 that the reduction of the proximity measure with respect to μ_q^t is independent of where the iterates are. Therefore, if the current iterate satisfies the assumption of the lemma, then after making the step, if the step is not a good step, then the new iterate has been stopped with $\frac{\mu_g(\alpha)}{\mu_q^h(\alpha)} = \tau$. This completes the proof of lemma. □

Remark 3.3.2 *After each bad step the algorithm is in the position to make a good step by Theorem 3.3.3.*

3.3.2 Iteration Complexity

To obtain an upper bound for the total number of iterations of Algorithm 3.2.1, we need to estimate the change of the parameter μ_q^t after an iteration. To do so, the following technical lemma is needed.

Lemma 3.3.3 *Let* $v_+ = \dfrac{v}{\sqrt{1-\theta}}$ *for some* $\theta \in (0,1)$. *Then*

$$\Psi_q(v_+) \leq \frac{\Psi_q(v)}{1-\theta} + \frac{n\theta}{2(1-\theta)} + \frac{\theta\tau n}{2(1-\theta)}.$$

Proof: From the definition of the proximity function we have

$$\begin{aligned}
\Psi_q(v_+) &= \frac{\|v_+\|^2 - n}{2} + \frac{\left\|v_+^{\frac{1-q}{2}}\right\|^2 - n}{q-1} \\
&= \frac{\frac{1}{1-\theta}\|v\|^2 - n}{2} + \frac{(1-\theta)^{\frac{q-1}{2}}\left\|v^{\frac{1-q}{2}}\right\|^2 - n}{q-1} \\
&= \frac{1}{1-\theta}\left(\frac{\|v\|^2 - n}{2} + \frac{\left\|v^{\frac{1-q}{2}}\right\|^2 - n}{q-1}\right) + \frac{n\theta}{2(1-\theta)} + \frac{n\theta}{(q-1)(1-\theta)} \\
&\quad + \left((1-\theta)^{\frac{q-1}{2}} - \frac{1}{1-\theta}\right)\frac{\left\|v^{\frac{1-q}{2}}\right\|^2}{q-1} \\
&\leq \frac{\Psi_q(v)}{1-\theta} + \frac{n\theta}{2(1-\theta)} + \frac{\theta}{1-\theta}\left(\frac{n - \left\|v^{\frac{1-q}{2}}\right\|^2}{q-1}\right) \\
&= \frac{\Psi_q(v)}{1-\theta} + \frac{n\theta}{2(1-\theta)} + \frac{n\theta}{1-\theta}\left(\frac{\frac{\mu_g}{\mu_q^t} - \tau}{2}\right),
\end{aligned}$$

where the last equality follows from equation (3.5). By Lemma 3.1.6 one has the desired inequality. □

By applying Lemma 3.3.3 to Theorem 3.3.1, we have the following theorem.

Theorem 3.3.4 *Let $\tau \geq 2$ and $(\Delta x, \Delta y, \Delta s)$ be the solution of system (3.4) as defined in Algorithm 3.2.1 with $\frac{\mu_g}{\mu_q^h} \leq (1 - \frac{1}{3q})^{q+1}\tau$, or $\frac{\mu_g}{\mu_q^h} = \tau$ and let α_q^* be the step size as defined in Corollary 3.3.1. Then*

$$\Phi_q\left(x(\alpha_q^*), s(\alpha_q^*), (1-\theta)\mu_q^t\right) \leq \Phi_q(x, s, \mu_q^t),$$

where

$$\theta = \frac{(\tau-1)^{\frac{q-1}{2q}}}{24q\tau^2 n^{\frac{q+1}{2q}}}.$$

Proof: From Lemma 3.3.3, it can be seen that to prove the theorem it suffices to choose θ satisfying the inequality

$$\Phi_q(x(\alpha_q^*), s(\alpha_q^*), \mu_q^t) + \frac{n\theta}{2} + \frac{n\theta\tau}{2} \leq (1-\theta)\Phi_q(x, s, \mu_q^t).$$

Using Theorem 3.3.1 we conclude that the above inequality will certainly be satisfied if

$$\theta\Phi_q(x, s, \mu_q^t) + \frac{n\theta}{2} + \frac{n\theta\tau}{2} \leq \frac{2^{\frac{q-1}{2q}}\Phi_q(x, s, \mu_q^t)^{\frac{q-1}{2q}}}{24q\tau}. \tag{3.10}$$

Recalling the fact that $\Phi_q(x, s, \mu_q^t) = \dfrac{(\tau-1)n}{2}$, we can rewrite inequality (3.10) as:

$$\theta\left(\frac{(\tau-1)n}{2} + \frac{n}{2} + \frac{n\tau}{2}\right) \leq \frac{(\tau-1)^{\frac{q-1}{2q}} n^{\frac{q-1}{2q}}}{24q\tau}.$$

This relation implies that if we choose

$$\theta = \frac{(\tau-1)^{\frac{q-1}{2q}}}{24q\tau^2 n^{\frac{q+1}{2q}}},$$

then

$$\Phi_q\left(x(\alpha_q^*), s(\alpha_q^*), (1-\theta)\mu_q^t\right) \leq \Phi_q(x, s, \mu_q^t)$$

that completes the proof. □

Remark 3.3.3 *One can prove analogous results for the case when* $\left(1-\frac{1}{3q}\right)^{q+1}\tau < \frac{\mu_g}{\mu_q^h} < \tau$. *Since the* θ *value might not be as good as in the previous case, therefore we do not aim to use it explicitly in the derivation of iteration complexity.*

Now we can proceed to discuss the complexity of Algorithm 3.2.1. By the choice of μ_q^t, we know that the proximity function $\Phi_q(x, s, \mu_q^t)$ remains invariant for all the iterates. Let us denote by $\mu_q^t(\alpha_q^*)$ the target parameter value after one step. Then we have

$$\Phi_q(x, s, \mu_q^t) = \Phi_q(x(\alpha_q^*), s(\alpha_q^*), \mu_q^t(\alpha_q^*)).$$

On the other hand, from Theorem 3.3.4 after a good step we have

$$\Phi_q(x(\alpha_q^*), s(\alpha_q^*), (1-\theta)\mu_q^t) \leq \Phi_q(x(\alpha_q^*), s(\alpha_q^*), \mu_q^t(\alpha_q^*)).$$

Since the proximity function is a quasiconvex function w.r.t. μ, we have

$$\mu_q^t(\alpha_q^*) \leq \left(1 - \frac{(\tau-1)^{\frac{q-1}{2q}}}{24q\tau^2}\right) n^{\frac{-q-1}{2q}} \mu_q^t. \tag{3.11}$$

Now we are ready to give the complexity of Algorithm 3.2.1.

Theorem 3.3.5 *Let* $\tau \geq 2$. *Then after at most*

$$\left\lceil \frac{48q\tau^2 n^{\frac{q+1}{2q}}}{(\tau-1)^{\frac{q-1}{2q}}} \log \frac{2n\tau}{\epsilon} \right\rceil$$

iterations Algorithm 3.2.1 will terminate with a feasible solution satisfying $x^T s \leq \epsilon$.

Proof: Since we do not take into account the reduction of μ_q^t for a bad step, in light of inequality (3.11) we know that after at most

$$\left\lceil \frac{48q\tau^2 n^{\frac{q+1}{2q}}}{(\tau-1)^{\frac{q-1}{2q}}} \log \frac{2n\tau}{\epsilon} \right\rceil$$

iterations we have $\mu_q^t \leq \frac{\epsilon}{2n\tau}$. By using (3.5), we can see that $\mu_g \leq 2\tau\mu_q^t \leq \frac{\epsilon}{n}$, or equivalently $x^T s \leq \epsilon$. □

The following corollary gives the best so far complexity for large-update IPMs [42, 43].

Corollary 3.3.2 *When $q = \frac{\log n}{2}$, then Theorem 3.3.5 provides the following upper bound for the total number of iterations:*

$$O\left(\sqrt{n}\log n \log\frac{n}{\epsilon}\right).$$

In summary in this chapter some properties of the proximity function induced by the family of Self-Regular kernel functions

$$\psi_q(t) = \frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}$$

have been explored. In particular, the properties of the SR-proximity function indicate that no matter if the present iterate is close to, or far from the central path, a large-update algorithm appears to be a natural choice for finding a good search direction and keeping control on the value of the proximity function. Furthermore, in some important cases, this self-regularity based search direction predicts the change of the duality gap along the search direction in the same way as the classical Newton direction does.

Based on these observations an adaptive single step large-update primal-dual SR-IPM for solving LO problems has been presented. We proved that the complexity of the algorithm matches the one for its SR-analogues presented in [42]. A common feature shared by most algorithms implemented in IPM solvers is the use of Mehrotra's predictor-corrector algorithm [35] where the predictor step is never performed but is used only to calculate an adaptive

update, and thus instead of a predictor and a corrector centering step, a single Newton step is made towards the adaptively chosen target (see Chapter 7 for more details). It is worth emphasizing again that our adaptive SR-IPM does not use any inner process to re-center, and analogous to Mehrotra's algorithm, it uses a single Newton step towards an adaptively chosen target.

Chapter 4

An Adaptive Infeasible IPM

In this chapter we present a new *Infeasible IPM (IIPM)* based on the adaptive updating scheme of Chapter 3 for a specific SR-proximity measure. Using the adaptive technique enables us to reduce the best so far worst case iteration complexity of IIPMs. The results we present in this chapter mostly appeared in [51, 55].

4.1 The Proximity Measure and Its Properties

In a feasible algorithm, like the one in Chapter 3, one is required to have a feasible starting point for the algorithm, while this is not the case for most of the practical problems that are modeled as LO problems [31]. Two different widely used approaches have been proposed to deal with this situation. The first one is the so called *self-dual embedding model* that is described in Section 10.2 of the appendix and the second approach is the class of the so called

IIPMs. In this chapter we present a new variant of IIPMs using a specific SR-proximity measure. The algorithm starts from a specific positive point not necessarily feasible. We discuss its worst case iteration complexity analysis and we show that using the idea of self-regularity helps to improve the iteration complexity compared to the classical infeasible algorithm. For practical efficiency, heuristics are used to obtain a starting point (see Section 10.3 of the Appendix).

Mathematically speaking, analogous to the feasible IPMs, by considering infeasibility, IIPMs solve the following Newton system at each iteration:

$$\begin{aligned} A\Delta x &= -r_b, \\ A^T\Delta y + \Delta s &= -r_c, \\ x\Delta s + s\Delta x &= \mu e - xs, \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} r_b &= Ax - b, \\ r_c &= A^T y + s - c \end{aligned}$$

are the residuals of the primal and dual problems.

The choice of parameter μ is similar to the feasible case. Since A has full row rank, for any $x > 0$ and $s > 0$, system (4.1) uniquely defines a Newton search direction $(\Delta x, \Delta y, \Delta s)$ and one can make a step while staying in a certain neighborhood of the central path. For some sufficiently small $\epsilon > 0$ we find an ϵ-optimal solution of problems (P) and (D) if $x^T s \leq \epsilon$ and $\|r_b, r_c\| \leq \epsilon$. (see Section 10.4 of the Appendix for the stopping criteria used in software packages).

As we saw in Chapters 2 and 3, by using the idea of self-regularity we were able to reduce the iteration complexity of feasible IPMs based algorithms

compared to the complexity of classical algorithms. The goal in this chapter is to use the SR approach to reduce the iteration complexity for infeasible algorithms. Throughout this chapter and the next chapter, we deal with the following SR-proximity measure:

$$\Phi_\ell(x,s,\mu) := \Psi_\ell(v) = \frac{e^T v^2 - n}{2} + \frac{e^T v^{-\log n} - n}{\log n}. \tag{4.2}$$

This proximity function is induced by the kernel function $\Gamma_{1q}(t)$, given by (2.13), where $q = 1 + \log n$. The following proposition is a special case of Proposition 3.1.1.

Proposition 4.1.1 *For any fixed $(x,s) > 0$, the proximity measure $\Phi_\ell(x,s,\mu)$, as a function of μ, has a global minimizer at*

$$\mu_\ell^* = \left(\frac{x^T s}{(x^{\frac{-\log n}{2}})^T s^{\frac{-\log n}{2}}} \right)^{\frac{2}{2+\log n}}. \tag{4.3}$$

Following the notation in Chapter 3 we also denote the generalized harmonic mean of the vectors x and s by

$$\mu_\ell^h = \left(\frac{n}{(x^{\frac{-\log n}{2}})^T s^{\frac{-\log n}{2}}} \right)^{\frac{2}{\log n}}. \tag{4.4}$$

Analogous to the feasible case, by considering infeasibility, one has to solve the following SR-proximity based Newton system of equations:

$$\begin{aligned} A\Delta x &= -r_b, \\ A^T\Delta y + \Delta s &= -r_c, \\ x\Delta s + s\Delta x &= -\mu v \nabla \Psi_\ell(v). \end{aligned} \tag{4.5}$$

Following the discussion in the feasible case, we require to that IIPMs operate in a certain neighborhood of the central path too. The widely used neighborhood by most of the IPMs based software packages is the negative infinity

norm neighborhood that is defined by

$$\mathcal{IN}_{\infty}^{-}(\rho)=\left\{(x,s)>0\big|\ \|(r_b,r_c)\|\leq\frac{\beta\|(r_b^0,r_c^0)\|\mu_g}{\mu_g^0},\ \left\|(v^2-e)^-\right\|_{\infty}\leq\rho\right\},\quad(4.6)$$

where $a^-=\min\{a,0\}$, $\beta\geq 1$ is a constant and $\rho\in(0,1)$ is a constant independent of n. We define the SR neighborhood in a way that it contains the neighborhood $\mathcal{IN}_{\infty}^{-}(\rho)$ and these two neighborhoods almost match each other [45]. An infeasible SR neighborhood can be defined as

$$\mathcal{IN}_{\ell}(\tau,\beta)=\left\{(x,s)>0\big|\|(r_b,r_c)\|\leq\frac{\beta\|(r_b^0,r_c^0)\|\mu_g}{\mu_g^0},\ \Phi_{\ell}(x,s,\mu_g)\leq\eta_{\ell}(n,\tau)\right\},\ (4.7)$$

where $\eta_{\ell}(n,\tau)=\frac{\left(\tau^{\frac{\log n}{2}}-1\right)n}{\log n}$ is a positive function that depends on a constant τ and the dimension of the underlying problem. Now we describe how $\eta_{\ell}(n,\tau)$ has been derived. Assume that $(x,s)\in\mathcal{IN}_{\infty}^{-}(\rho)$, then for this (x,s) pair, with $\mu=\mu_g$, we have

$$\Phi_{\ell}(x,s,\mu_g)=\frac{e^Tv^{-\log n}-n}{\log n}\leq\frac{n(1-\rho)^{\frac{-\log n}{2}}-n}{\log n}=\frac{(\tau^{\frac{\log n}{2}}-1)n}{\log n}:=\eta_{\ell}(n,\tau),$$

where $\tau=\frac{1}{1-\rho}$. With this choice of τ the neighborhood $\mathcal{IN}_{\ell}(\tau,\beta)$ contains the neighborhood $\mathcal{IN}_{\infty}^{-}(\rho)$. The reverse inclusion holds for a larger infinity norm neighborhood (the one with smaller parameter), which means that the SR neighborhood is slightly wider than $\mathcal{IN}_{\infty}^{-}(\rho)$, but it can be included in a negative infinity norm neighborhood with a larger parameter ρ.

Now, analogous to Chapter 3, we are interested in determining what is the smallest μ value for which the equality $\Phi_{\ell}(x,s,\mu)=\frac{(\tau-1)n}{2}$ holds. Let us denote this value by $\mu=\mu_{\ell}^t$. It is easy to verify (see Chapter 3) that μ_{ℓ}^t exists whenever $\Phi_{\ell}(x,s,\mu_g)\leq\eta_{\ell}(n,\tau)$ and it is the smaller positive root of the equation

$$f(\mu)=2\left(x^{\frac{-\log n}{2}}\right)^T s^{\frac{-\log n}{2}}\mu^{\frac{2+\log n}{2}}-(2+\tau\log n)\,n\mu+x^Ts\log n=0.\quad(4.8)$$

Since[1] $n \geq 2$ and $(x, s) > 0$, the function $f(\mu)$ is a convex function of μ. Thus, whenever $\Phi_\ell(x, s, \mu_g) \leq \eta_\ell(n, \tau)$, equation (4.8) has two positive real roots, one is less than or equal to μ_ℓ^* and the other one is greater than or equal to μ_ℓ^*. We choose the smaller root of this equation as a default value for μ_ℓ^t. Hence, when $\mu_g \leq \tau\mu_h^\ell$, we have $\mu_\ell^t \leq \mu_\ell^h$, and equality holds if $\mu_g = \tau\mu_\ell^h$. The relation between μ_ℓ^t and μ_g is described in the following lemma, which is analogous to Lemma 3.1.6.

Lemma 4.1.1 *Let $\tau \geq 2$ and $\Phi_\ell(x, s, \mu_g) \leq \eta_\ell(n, \tau)$. Then, we have* $\tau \leq \dfrac{\mu_g}{\mu_\ell^t} \leq \tau + \dfrac{2}{\log n}$.

The following remark is crucial in the design of our new algorithm.

Remark 4.1.1 *In IIPMs it is crucial to keep balance between infeasibility and the complementarity gap. To enhance our ability to maintain this balance we introduce a modified Newton system. When the iterates are not in a certain neighborhood of the central path, then instead of (4.5), our algorithm uses the following system of equations*

$$\begin{aligned} A\Delta x &= -r_b, \\ A^T\Delta y + \Delta s &= -r_c, \\ x\Delta s + s\Delta x &= -\frac{\mu}{2} v\nabla\Psi_\ell(v). \end{aligned} \tag{4.9}$$

The reduction of the right hand side of the third equation in (4.5) is crucial in order to get a reasonable step size which enables us to prove better iteration complexity. Then system (4.5) and (4.9) can be written in a modified form as

[1]The case n=1 is obvious, so for simplicity of the analysis we do not consider this case.

$$\begin{aligned} A\Delta x &= -r_b, \\ A^T\Delta y + \Delta s &= -r_c, \\ x\Delta s + s\Delta x &= -\frac{\mu}{\chi} v\nabla\Psi_\ell(v), \end{aligned} \tag{4.10}$$

where $\chi = 1$ or 2.

Now we proceed to discuss the change of the complementary gap along the SR search direction for different updates of μ. For this, we rewrite (4.10) in the original space. Due to our specific choice of the SR proximity, system (4.10) has the form

$$\begin{aligned} A\Delta x &= -r_b, \\ A^T\Delta y + \Delta s &= -r_c, \\ s\Delta x + x\Delta s &= \frac{1}{\chi}(\mu^{\frac{2+\log n}{2}} x^{\frac{-\log n}{2}} s^{\frac{-\log n}{2}} - xs). \end{aligned} \tag{4.11}$$

Let us denote by $(\Delta x(\mu), \Delta y(\mu), \Delta s(\mu))$ the solution of system (4.11). The following results show how the complementarity gap changes with the choice of the targeted parameter μ.

Lemma 4.1.2 *Let μ_ℓ^*, μ_ℓ^h, and μ_ℓ^t be defined by (4.3), (4.4) and (4.8), respectively. Then*
1) $x^T\Delta s(\mu_\ell^*) + s^T\Delta x(\mu_\ell^*) = 0$,
2) $s^T\Delta x(\mu_\ell^h) + x^T\Delta s(\mu_\ell^h) = \frac{n}{\chi}\left(\mu_\ell^h - \mu_g\right)$,
3) $s^T\Delta x(\mu_\ell^t) + x^T\Delta s(\mu_\ell^t) = \frac{1}{\chi}\left((\mu_\ell^t)^{\frac{2+\log n}{2}}\left(x^{\frac{-\log n}{2}}\right)^T s^{\frac{-\log n}{2}} - x^T s\right)$.

By using the previous lemma one can easily derive the following theorem as.

Theorem 4.1.1 *Let μ_ℓ^h, μ_ℓ^* and μ_ℓ^t be defined by (4.4), (4.3) and (4.8). The following statements hold.*

1) If the target parameter μ is μ_ℓ^*, then for any feasible step size α, we have

$$(x+\alpha\Delta x(\mu_\ell^*))^T(s+\alpha\Delta s(\mu_\ell^*)) = x^Ts\left(1+\alpha^2\frac{\Delta x(\mu_\ell^*)^T\Delta s(\mu_\ell^*)}{x^Ts}\right).$$

2) If the target parameter μ is μ_ℓ^h, then for any feasible step size α, we have

$$\left(x+\alpha\Delta x(\mu_\ell^h)\right)^T\left(s+\alpha\Delta s(\mu_\ell^h)\right) = x^Ts\left(1-\frac{\alpha}{\chi}+\frac{\alpha\mu_\ell^h}{\chi\mu_g}+\alpha^2\frac{\Delta x(\mu_\ell^h)^T\Delta s(\mu_\ell^h)}{x^Ts}\right).$$

3) If the target parameter μ is μ_ℓ^t, then for any feasible step size α, we have

$$\left(x+\alpha\Delta x(\mu_\ell^t)\right)^T\left(s+\alpha\Delta s(\mu_\ell^t)\right) = x^Ts\left(1-\frac{\alpha}{\chi}+\frac{\alpha\left(\mu_\ell^t\right)^{\frac{2+\log n}{2}}}{\chi\mu_g\left(\mu_\ell^h\right)^{\frac{\log n}{2}}}+\alpha^2\frac{\Delta x(\mu_\ell^t)^T\Delta s(\mu_\ell^t)}{x^Ts}\right).$$

4.2 Algorithmic Scheme

In this section a new variant of IIPMs using a specific SR-proximity measure is considered. For simplicity, we use the notations

$$\mu_g(\alpha)=\frac{x(\alpha)^Ts(\alpha)}{n},\qquad \mu_\ell^h(\alpha)=\left(\frac{n}{\left((x(\alpha))^{\frac{-\log n}{2}}\right)^T(s(\alpha))^{\frac{-\log n}{2}}}\right)^{\frac{2}{\log n}}$$

and

$$\mu_\ell^*(\alpha) \;=\; \left(\mu_g(\alpha)(\mu_\ell^h(\alpha))^{\frac{\log n}{2}}\right)^{\frac{2}{2+\log n}}.$$

Using the scaling technique of Chapter 2, the third equation of system (4.10) can be rewritten as

$$d_x+d_s=-\frac{1}{\chi}\nabla\Psi_\ell(v).$$

We also denote the norm of the gradient of $\Psi_\ell(v)$ by σ_ℓ i.e.,

$$\sigma_\ell = \|\nabla\Psi_\ell(v)\| = \|v - v^{-1-\log n}\|. \tag{4.12}$$

In our algorithm we use the neighborhood given by (4.7) and regardless of whether the iterate is close to or far away from the central path, we always make an adaptive large update $\mu_+ = (1-\theta)\mu$ of the central path parameter μ. We utilize a constant $\tau \geq 2$ to keep control on the distance of the iterate to the central path and to force the value of the proximity function to satisfy the following relation

$$\Phi_\ell(x, s, \mu_g) \leq \eta_\ell(n, \tau). \tag{4.13}$$

At each step we stipulate that the step size should be chosen so that the proximity function $\Phi_\ell(x(\alpha),\ s(\alpha),\ \mu)$ has a sufficient decrease, while the new iterate is still in $\mathcal{IN}_\ell(\tau, \beta)$. The new algorithm can be outlined as follows.

Algorithm 4.2: SR-IIPM

Input:

Proximity parameter $\tau \geq 2$ and parameter $\beta \geq 1$;

$\eta_1(n, \tau, \sigma_\ell^k)$, a positive function, is given by (4.44);

neighborhood $\mathcal{IN}_\ell(\tau, \beta)$;

an accuracy parameter $\varepsilon > 0$;

$(x^0, s^0) \in \mathcal{IN}_\ell(\tau, \beta)$; $k = 0$.

begin

while $\max\left\{(x^k)^T s^k, \left\|r_b^k\right\|, \left\|r_c^k\right\|\right\} \geq \epsilon$ **do**

begin

$\mu^k := (\mu_\ell^t)^k$ defined by (4.8);

Let $\chi_k = 1$ when $\frac{\mu_g^k}{\mu^k} \leq \tau + \eta_1(n, \tau_1, \sigma_\ell)$

and $\chi_k = 2$ when $\frac{\mu_g^k}{\mu^k} > \tau + \eta_1(n, \tau_1, \sigma_\ell)$.

Solve system (4.11) for $(\Delta x^k, \Delta y^k, \Delta s^k)$.

begin

Determine the maximum step size $\bar{\alpha}_k$ such that

$(x + \bar{\alpha}_k \Delta x, s + \bar{\alpha}_k \Delta s) \geq 0$;

Let[a]

$\alpha_k = \max\left\{\alpha \in (0\ \bar{\alpha}_k] \middle| \Phi_\ell^+ \leq \Phi_\ell - \frac{(\sigma_\ell^k)^2 \alpha_{\ell k}^*}{4\chi_k}, (x(\alpha), s(\alpha)) \in \mathcal{IN}_\ell(\tau, \beta)\right\}$;

Set $x^{k+1} := x(\alpha_k)$; $y^{k+1} := y(\alpha_k)$; $s^{k+1} := s(\alpha_k)$;

$k = k + 1$.

end

end

end

---

[a] $\Phi_\ell := \Phi_\ell(x^k, s^k, (\mu_\ell^t)^k)$, $\Phi_\ell^+ := \Phi_\ell(x(\alpha), s(\alpha), (\mu_\ell^t)^k)$ and $\alpha_{\ell k}^*$ is given in Lemma 4.3.8 in page 84.

4.3 Complexity Analysis

We proceed to analyze the complexity of Algorithm 4.2. The key element of the analysis is to estimate the value of the step size α in Algorithm 4.2. To do so, we need to estimate the second order term $\frac{\left(\Delta x^k\right)^T \Delta s^k}{\mu^k}$ at the k^{th} iterate. Let

$$\varphi_k = \prod_{i=1}^{k-1} (1 - \alpha_i)\,,\ k = 1, 2, \cdots \tag{4.14}$$

where α_i is the step size at the i^{th} iterate. Because the first two components of system (4.11) are linear, we have

$$(r_b^k, r_c^k) = \varphi_k (r_b^0, r_c^0). \tag{4.15}$$

Since $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$, from (4.14) we derive

$$\varphi_k \frac{\|(r_b^0, r_c^0)\|}{\mu_g^k} = \frac{\left\|(r_b^k, r_c^k)\right\|}{\mu_g^k} \leq \beta \frac{\|(r_b^0, r_c^0)\|}{\mu_g^0}.$$

Provided that $(r_b^0, r_c^0) \neq 0$, it follows from this inequality that

$$\varphi_k \leq \frac{\beta \mu_g^k}{\mu_g^0}. \tag{4.16}$$

To prove polynomial complexity of Algorithm 4.2 one needs to obtain a lower bound on the step size which is an inverse polynomial function of the dimension n. As it is customary in IIPMs [63], we choose the starting point to satisfy

$$(x^0, y^0, s^0) \ = \ (\zeta e, 0, \zeta e), \tag{4.17}$$

where ζ is a scalar for which[2]

$$\|(x^*, s^*)\|_\infty \ \leq \ \zeta \tag{4.18}$$

[2]For practical purpose heuristics are used to obtain starting points (see Section 10.3 of the Appendix).

for some primal-dual optimal solution (x^*, y^*, s^*). Usually we do not know the value $\|(x^*, s^*)\|_\infty$, because we do not know any optimal solution a priori. However, these conditions are still relevant. Theoretically, we can choose $\zeta = O(2^L)$ where L is the input length of the LO problem [63]. Such an initial point with sufficiently large ζ is helpful to determine whether the problem has an optimal solution or it is infeasible.

4.3.1 Some Technical Results

We mention that the orthogonality of the vectors Δx and Δs does not hold in IIPMs. In this section, we estimate the second order term $\frac{\left(\Delta x^k\right)^T \Delta s^k}{\mu^k}$. First we provide a lower bound for $v^k_{\min} = \min\{v^k_i |\ 1 \leq i \leq n\}$, the smallest coordinate of the vector v^k. This result will be used frequently in the rest of this chapter (e.g., in the proof of Lemma 4.3.2).

Lemma 4.3.1 *Suppose the current iterate is in the neighborhood $\mathcal{IN}_\ell(\tau, \beta)$. Let $\mu = (\mu^t_\ell)^k$, $n \geq 2$ and $\tau \geq 2$, then*

$$(\sigma^k_\ell)^2 \geq (\tau - 1)n \geq 2, \tag{4.19}$$

$$v^k_{\min} \geq e^{-1}\left(\tau + \frac{2}{\log n}\right)^{\frac{-1}{\log n}}, \tag{4.20}$$

$$(v^k_{\min})^{1+\log n}\sigma^k_\ell \geq \frac{1}{2}. \tag{4.21}$$

Proof: By Proposition 2.6.2 we have

$$(\sigma^k_\ell)^2 \geq 2\Phi_\ell\left(x^k, s^k, (\mu^t_\ell)^k\right) = (\tau - 1)n$$

that proves the first statement of the lemma. Using Lemma 4.1.1 and the fact that $(\mu^t_\ell)^k \leq (\mu^*_\ell)^k$, we have

$$(v^k_{\min})^{-\log n} \leq \left\|(v^k)^{-\frac{\log n}{2}}\right\|^2 \leq \left\|v^k\right\|^2 = \frac{(x^k)^T s^k}{(\mu^t_\ell)^k} \leq n\left(\tau + \frac{2}{\log n}\right)$$

that implies

$$v_{\min}^k \geq e^{-1}\left(\tau + \frac{2}{\log n}\right)^{\frac{-1}{\log n}}.$$

This way (4.20) is proved. To prove (4.21) we recall Theorem 2.6.1 again and note that inequality (4.19) implies

$$v_{\min}^k \geq (1+\sigma_\ell^k)^{\frac{-1}{1+\log n}} \quad \text{and} \quad \sigma_\ell^k \geq \sqrt{2}.$$

Therefore we have

$$(v_{\min}^k)^{1+\log n}\sigma_\ell^k \geq \frac{\sigma_\ell^k}{1+\sigma_\ell^k} \geq \frac{1}{2}.$$

This completes the proof of the Lemma. □

The following lemma gives an upper bound for $\varphi_k\|(x^k, s^k)\|_1$, where φ_k is defined by (4.14) and this result will be used in the proof of Lemma 4.3.3. The proof is analogous to the proof of Lemmas 6.3 and 6.4 in [63].

Lemma 4.3.2 *Suppose that $n \geq 2, \tau \geq 2$ and the initial point (x^0, y^0, s^0) is chosen so that it satisfies (4.17) and (4.18), and for the current iterate $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$ holds. Then*

$$\zeta\varphi_k\|(x^k, s^k)\|_1 \quad \leq \quad C_1 n\mu_g^k,$$

where $C_1 = 3\beta + 1$.

Let us define $D^k = \text{diag}\left((x^k)^{\frac{1}{2}}(s^k)^{-\frac{1}{2}}\right)$. We also use the matrix norm defined for a matrix $M \in \mathbb{R}^{p\times q}$ as

$$\|M\| = \max_{u\in\mathbb{R}^q:\|u\|=1} \|Mu\|.$$

It is known that $\|Mu\| \leq \|M\|\|u\|$, for all $u \in \mathcal{R}^q$. By the following lemma we give upper bounds for the scaled vectors $(D^k)^{-1}\Delta x^k$ and $D^k\Delta s^k$ that enable us to bound $(\Delta x^k)^T\Delta s^k$.

Lemma 4.3.3 *Suppose that $n \geq 2$, $\tau \geq 2$, the initial point is chosen so that it satisfies (4.17) and (4.18), and for the current iterate $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$ holds. Then*

$$\|(D^k)^{-1}\Delta x^k\| \leq C_2\sqrt{n\mu_g^k}\sigma_\ell^k, \quad \|D^k\Delta s^k\| \leq C_2\sqrt{n\mu_g^k}\sigma_\ell^k, \tag{4.22}$$

where $C_2 = 6C_1 e\tau + 1$.

Proof: Let us define $(\bar{x}, \bar{y}, \bar{s})$ as follows

$$(\bar{x}, \bar{y}, \bar{s}) = \varphi_k[(x^0, y^0, s^0) - (x^*, y^*, s^*)] + (\Delta x^k, \Delta y^k, \Delta s^k),$$

It is not hard to check that $(\bar{x}, \bar{y}, \bar{s})$ satisfies

$$\begin{aligned} A\bar{x} &= 0, \\ A^T\bar{y} + \bar{s} &= 0. \end{aligned}$$

Thus, we have $\bar{x}^T\bar{s} = 0$, i.e.,

$$\left(\Delta x^k + \varphi_k(x^0 - x^*)\right)^T \left(\Delta s^k + \varphi_k(s^0 - s^*)\right) = 0. \tag{4.23}$$

From the third equation of system (4.10), we have

$$\begin{aligned} x^k\left(\Delta s^k + \varphi_k(s^0 - s^*)\right) + s^k\left(\Delta x^k + \varphi_k(x^0 - x^*)\right) \\ = -\frac{\mu^k}{\chi_k}v^k\nabla\Psi_\ell(v^k) + \varphi_k s^k(x^0 - x^*) + \varphi_k x^k(s^0 - s^*). \end{aligned}$$

By multiplying this equation by $\left(X^kS^k\right)^{-\frac{1}{2}}$ and setting $(D^k)^{-1} = (X^k)^{-\frac{1}{2}}(S^k)^{\frac{1}{2}}$ and $D^k = (X^k)^{\frac{1}{2}}(S^k)^{-\frac{1}{2}}$, one has

$$\begin{aligned} (D^k)^{-1}\left(\Delta x^k + \varphi_k(x^0 - x^*)\right) + D^k\left(\Delta s^k + \varphi_k(s^0 - s^*)\right) \\ = -\frac{\sqrt{\mu^k}}{\chi_k}\nabla\Psi_\ell(v^k) + \varphi_k(D^k)^{-1}(x^0 - x^*) + \varphi_k D^k(s^0 - s^*). \end{aligned} \tag{4.24}$$

Then, from (4.23), we have

$$\begin{aligned}\|(D^k)^{-1} \left(\Delta x^k + \varphi_k(x^0 - x^*)\right) + D^k \left(\Delta s^k + \varphi_k(s^0 - s^*)\right) \|^2 \\ = \|(D^k)^{-1} \left(\Delta x^k + \varphi_k(x^0 - x^*)\right) \|^2 + \|D^k \left(\Delta s^k + \varphi_k(s^0 - s^*)\right) \|^2.\end{aligned}$$

By taking squared norm in both sides of (4.24), one has

$$\begin{aligned}\|(D^k)^{-1} \left(\Delta x^k + \varphi_k(x^0 - x^*)\right) \|^2 + \|(D^k)^{-1} \left(\Delta s^k + \varphi_k(s^0 - s^*)\right) \|^2 \\ \leq \left(\frac{1}{\chi_k}\|\sqrt{\mu^k}\nabla\Psi_\ell(v^k)\| + \varphi_k\|(D^k)^{-1}(x^0 - x^*)\| + \varphi_k\|D^k(s^0 - s^*)\|\right)^2.\end{aligned}$$

Let us isolate the first term on the left-hand-side and write

$$\begin{aligned}\left\|(D^k)^{-1} \left(\Delta x^k + \varphi_k(x^0 - x^*)\right)\right\| \leq \frac{1}{\chi_k} \left\|\sqrt{\mu^k}\nabla\Psi_\ell(v^k)\right\| \\ +\varphi_k \left(\left\|(D^k)^{-1}(x^0 - x^*)\right\| + \left\|D^k(s^0 - s^*)\right\|\right).\end{aligned} \tag{4.25}$$

A simple application of the triangle inequality and the addition of an extra term
$\varphi_k \left\|D^k(s^0 - s^*)\right\|$ to the right-hand-side gives

$$\begin{aligned}\left\|(D^k)^{-1}\Delta x^k\right\| \leq \frac{1}{\chi_k} \left\|\sqrt{\mu^k}\nabla\Psi_\ell(v^k)\right\| \\ +2\varphi_k \left(\left\|(D^k)^{-1}(x^0 - x^*)\right\| + \left\|D^k(s^0 - s^*)\right\|\right).\end{aligned} \tag{4.26}$$

Next we show that the magnitude of each term on the right-hand-side of (4.26) is $O\left(\sqrt{\mu_g^k}\right)$. From Lemmas 4.1.1 and 4.3.1, we have

$$\begin{aligned}\|(X^k S^k)^{-\frac{1}{2}}\| = \max_{i=1,\ldots,n} \frac{1}{(x_i^k s_i^k)^{\frac{1}{2}}} \leq \frac{e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}}}{((\mu_\ell^t)^k)^{\frac{1}{2}}} \\ \leq e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} \sqrt{\frac{2 + \tau\log n}{\mu_g^k \log n}}.\end{aligned} \tag{4.27}$$

On the other hand, from the definition of the matrix norm we have

$$\|(D^k)^{-1}\| \;=\; \|(X^kS^k)^{-\frac{1}{2}}S^ke\|_\infty \le \|(X^kS^k)^{-\frac{1}{2}}\|\|s^k\|_1,$$

and similarly $\|D^k\| \le \|(X^kS^k)^{-\frac{1}{2}}\|\|x^k\|_1$. So, we have

$$\begin{aligned}\varphi_k\left(\|(D^k)^{-1}(x^0-x^*)\|+\|D^k(s^0-s^*)\|\right)&\le \varphi_k\zeta\left(\|(D^k)^{-1}e\|+\|D^ke\|\right)\\ &= \varphi_k\zeta\left(\|(X^kS^k)^{-\frac{1}{2}}s^k\|+\|(X^kS^k)^{-\frac{1}{2}}x^k\|\right)\\ &\le \varphi_k\zeta\|(X^kS^k)^{-\frac{1}{2}}\|\|(x^k,s^k)\|_1. \qquad (4.28)\end{aligned}$$

From (4.27), (4.28) and Lemma 4.3.2 we have

$$\begin{aligned}&\varphi_k\left(\|(D^k)^{-1}(x^0-x^*)\|+\|D^k(s^0-s^*)\|\right)\\ &\qquad\le C_1ne\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{2}+\frac{1}{\log n}}\sqrt{\mu_g^k}. \qquad (4.29)\end{aligned}$$

Finally, using inequality (4.29) and relation (4.19), from (4.26) we have

$$\|(D^k)^{-1}\Delta x^k\| \le \frac{\sqrt{\mu_g^k}}{\chi_k}\sigma_\ell^k + 2C_1ne\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{2}+\frac{1}{\log n}}\sqrt{\mu_g^k} \le C_2\sqrt{n\mu_g^k}\sigma_\ell^k,$$

where $C_2 = 6C_1e\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}}+1$. Analogously, one can derive

$$\|(D^k)^{-1}\Delta s^k\| \le C_2\sqrt{n\mu_g^k}\sigma_\ell^k.$$

This completes the proof of the lemma. □

Corollary 4.3.1 *Suppose that the assumptions of Lemma 4.3.3 are satisfied. Then*

$$\frac{\left|\left(\Delta x^k\right)^T\Delta s^k\right|}{\mu_g^k} \;\le\; C_2^2n(\sigma_\ell^k)^2. \qquad (4.30)$$

Proof: By Lemma 4.3.3 we have

$$\left|\left(\Delta x^k\right)^T \Delta s^k\right| = \left|\left((D^k)^{-1}\Delta x^k\right)^T D^k \Delta s^k\right| \;\leq\; C_2^2 \mu_g^k n(\sigma_\ell^k)^2.$$

This completes the proof. □

The following lemma will be used to derive a lower bound for the step size in Algorithm 4.2.

Lemma 4.3.4 *Suppose that $n \geq 2$, $\tau \geq 2$, the initial point is chosen so that it satisfies (4.17) and (4.18), and for the current iterate $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$ holds. Then,*

$$(\sigma_1^k)^2 \leq C_3 n(\sigma_\ell^k)^2, \tag{4.31}$$

where $(\sigma_1^k)^2 := \|d_{x^k}\|^2 + \|d_{s^k}\|^2$ and $C_3 = 2\left(\tau + \frac{2}{\log n}\right) C_2^2 + 1$.

Proof: Using (4.12) we have

$$\begin{aligned}(\sigma_\ell^k)^2 = \chi_k^2 \|d_{x^k} + d_{s^k}\|^2 &= \chi_k^2(\|d_{x^k}\|^2 + \|d_{s^k}\|^2) + 2\chi_k^2 \left(d_{x^k}\right)^T d_{s^k} \\ &= \chi_k^2(\sigma_1^k)^2 + 2\chi_k^2 \frac{\left(\Delta x^k\right)^T \Delta s^k}{(\mu_\ell^t)^k}. \end{aligned} \tag{4.32}$$

Thus, by (4.30) we have

$$\begin{aligned}(\sigma_1^k)^2 &= \frac{(\sigma_\ell^k)^2}{\chi_k^2} - 2\frac{\left(\Delta x^k\right)^T \Delta s^k}{(\mu_\ell^t)^k} \leq \frac{(\sigma_\ell^k)^2}{\chi_k^2} + 2\frac{\left|\left(\Delta x^k\right)^T \Delta s^k\right|}{(\mu_\ell^t)^k} \\ &\leq \frac{(\sigma_\ell^k)^2}{\chi_k^2} + 2\left(\tau + \frac{2}{\log n}\right)\frac{\left|\left(\Delta x^k\right)^T \Delta s^k\right|}{\mu_g^k} \\ &\leq \frac{(\sigma_\ell^k)^2}{\chi_k^2} + 2\left(\tau + \frac{2}{\log n}\right) C_2^2 n(\sigma_\ell^k)^2 \leq C_3 n(\sigma_\ell^k)^2,\end{aligned}$$

where the second inequality follows from Lemma 4.1.1 and the third inequality follows from Corollary 4.3.1. This completes the proof of the lemma. □

4.3.2 Estimating the step size

In this section we discuss in detail how to derive a lower bound for the step size in the specified neighborhood that leads us to the polynomiality of the algorithm. The following theorem provides a lower bound for the maximal feasible step size and gives a certain reduction for the proximity function.

Theorem 4.3.1 *Let $n \geq 2$, $\tau \geq 2$, $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$, and $(\Delta x^k, \Delta y^k, \Delta s^k)$ be the solution of system (4.11) with $\mu^k = (\mu_\ell^t)^k$ and χ_k is chosen as in Algorithm 4.2. Then, the maximal feasible step size, $\alpha_\ell^{\max}$ satisfies*

$$\alpha_\ell^{\max} \geq \bar{\alpha}_{\ell k} = \frac{v_{\min}^k}{\sigma_1^k} \geq \frac{1}{e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} \sigma_1^k}.$$

Moreover, for any step size

$$\alpha \leq \hat{\alpha}_{\ell k} = \frac{4\sigma_\ell^k}{2e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} \sigma_1^k \log n \left(4\chi_k \sigma_1^k + \sigma_\ell^k\right)} \tag{4.33}$$

the following inequality holds:

$$\Phi_\ell\left(x(\alpha), s(\alpha), \left(\mu_\ell^t\right)^k\right) \leq \Phi_\ell\left(x^k, s^k, \left(\mu_\ell^t\right)^k\right) - \frac{(\sigma_\ell^k)^2}{4\chi_k}\alpha. \tag{4.34}$$

Proof: The proof of the first part is analogous to Lemma 3.3.1 by using Lemma 4.3.1. To prove the second part, for simplicity we drop the subscript and the superscript k. Now, let us define

$$f(\alpha) = \Psi_\ell(v(\alpha)) - \Psi_\ell(v),$$

where $v(\alpha) = \sqrt{\frac{x(\alpha)s(\alpha)}{\mu_\ell^t}} = \sqrt{(v + \alpha d_x)(v + \alpha d_s)}$. Then by the definition of the proximity function (4.2) one has

$$f(\alpha) = \frac{1}{2}v^T(d_x + d_s)\alpha + \frac{\alpha^2}{2}d_x^T d_s + \frac{\left\|v^{-\frac{\log n}{2}}(\alpha)\right\|^2}{\log n} - \frac{\left\|v^{-\frac{\log n}{2}}\right\|^2}{\log n}.$$

Since $\left\|v^{-\frac{\log n}{2}}(\alpha)\right\|^2$ satisfies condition SR.2 of Definition 2.6.1,

$$\begin{aligned} f(\alpha) \leq \frac{1}{2}v^T(d_x + d_s)\alpha \ &+\frac{\alpha^2}{2}d_x^T d_s \\ &+\frac{1}{2\log n}\sum_{i=1}^{n}[(v_i + \alpha(d_x)_i)^{-\log n} + (v_i + \alpha(d_s)_i)^{-\log n}] \\ &-\frac{\left\|v^{\frac{-\log n}{2}}\right\|^2}{\log n} := f_1(\alpha). \end{aligned}$$

The first and second derivatives of $f_1(\alpha)$ are

$$\begin{aligned} f_1'(\alpha) \ &= \frac{1}{2}v^T(d_x + d_s) + \alpha d_x^T d_s \\ &\quad -\frac{1}{2}\sum_{i=1}^{n}[(v_i + \alpha(d_x)_i)^{-\log n - 1}(d_x)_i + (v_i + \alpha(d_s)_i)^{-\log n - 1}(d_s)_i] \end{aligned}$$

and

$$\begin{aligned} f_1''(\alpha) \ &= d_x^T d_s \\ &+\frac{(1+\log n)}{2}\sum_{i=1}^{n}[(v_i + \alpha(d_x)_i)^{-2-\log n}(d_x)_i^2 + (v_i + \alpha(d_s)_i^2)^{-2-\log n}(d_s)_i] \\ &\leq \frac{\sigma_1^2}{2}\left((1+\log n)(v_{\min} - \alpha\sigma_1)^{-2-\log n} + 1\right). \end{aligned} \tag{4.35}$$

By the definition of f and f_1 one also has

$$f(0) = f_1(0) = 0; \quad f_1'(0) = -\frac{\sigma_\ell^2}{2\chi}.$$

Using the fact that

$$f_1(\alpha) = f_1(0) + f_1^{'}(0)\alpha + \int_0^\alpha \int_0^\xi f_1^{''}(\eta)d\eta d\xi$$

we may write

$$f_1(\alpha) \leq -\frac{\sigma_\ell^2 \alpha}{2\chi} + \frac{\sigma_1^2}{2}\int_0^\alpha \int_0^\xi \left((1+\log n)(v_{\min} - \eta\sigma_1)^{-2-\log n} + 1\right) d\eta d\xi \\ := f_2(\alpha).$$

The function $f_2(\alpha)$ is convex and twice continuously differentiable in the interval $[0, \bar{\alpha}_{\ell k})$. Let us denote by $\check{\alpha}_{\ell k}$ the global minimizer of $f_2(\alpha)$ in the interval $[0, \bar{\alpha}_{\ell k})$. Then it is the unique solution of the equation

$$-\frac{\sigma_\ell^2}{2\chi} + \frac{\sigma_1}{2}(v_{\min} - \alpha\sigma_1)^{-1-\log n} - \frac{\sigma_1}{2} v_{\min}^{-1-\log n} + \frac{\alpha\sigma_1^2}{2} = 0,$$

or

$$-\frac{\sigma_\ell^2}{\chi} + \sigma_1(v_{\min} - \alpha\sigma_1)^{-1-\log n} - \sigma_1 v_{\min}^{-1-\log n} + \alpha\sigma_1^2 = 0. \tag{4.36}$$

Now let us define

$$\omega_1(\alpha) = -\frac{\sigma_\ell^2}{2\chi} + \alpha\sigma_1^2,$$

$$\omega_2(\alpha) = -\frac{\sigma_\ell^2}{2\chi} + \sigma_1(v_{\min} - \alpha\sigma_1)^{-1-\log n} - \sigma_1 v_{\min}^{-1-\log n}.$$

It is obvious that ω_1 and ω_2 are increasing function w.r.t. α on the interval $[0, \bar{\alpha}_{\ell k})$. The root α_1^* of $\omega_1(\alpha)$ is $\frac{\sigma_\ell^2}{2\chi\sigma_1^2}$ and the root of $\omega_2(\alpha)$ is

$$\alpha_2^* = \frac{v_{\min}}{\sigma_1}\left(1 - \left(\frac{2\chi\sigma_1}{2\chi\sigma_1 + \sigma_\ell^2 v_{\min}^{1+\log n}}\right)^{\frac{1}{1+\log n}}\right). \tag{4.37}$$

Now, by using Lemma 10.1.1 we have

$$\left(\frac{2\chi\sigma_1}{2\chi\sigma_1 + \sigma_\ell^2 v_{\min}^{1+\log n}}\right)^{\frac{1}{1+\log n}} = \left(1 - \frac{\sigma_\ell^2 v_{\min}^{1+\log n}}{2\chi\sigma_1 + \sigma_\ell^2 v_{\min}^{1+\log n}}\right)^{\frac{1}{1+\log n}} \\ \leq 1 - \frac{\sigma_\ell^2 v_{\min}^{1+\log n}}{(1+\log n)\left(2\chi\sigma_1 + \sigma_\ell^2 v_{\min}^{1+\log n}\right)}.$$

By substituting this in (4.37) we conclude that

$$\alpha_2^* \geq \frac{\sigma_\ell^2 v_{\min} v_{\min}^{1+\log n}}{(1+\log n)\sigma_1 \left(2\chi\sigma_1 + v_{\min}^{1+\log n}\sigma_\ell^2\right)}.$$

Using Lemma 4.3.1, one has

$$\alpha_2^* \geq \frac{\sigma_\ell}{2e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} \sigma_1 \log n \left(4\chi\sigma_1 + \sigma_\ell\right)}.$$

Since, (4.36) is equivalent to $\omega_1(\alpha) + \omega_2(\alpha) = 0$, then

$$\check{\alpha}_{\ell k} \geq \min(\alpha_1^*, \alpha_2^*) \geq \frac{\sigma_\ell}{2e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} \sigma_1 \log n \left(4\chi\sigma_1 + \sigma_\ell\right)}.$$

Finally, for any $\alpha \leq \check{\alpha}_{\ell k}$ we have

$$f(\alpha) \leq f_1(\alpha) \leq f_2(\alpha) \leq \frac{f_2'(0)}{2}\alpha = -\frac{\sigma_\ell^2}{4\chi}\alpha,$$

where the last inequality follows from Lemma 10.1.2. □

Corollary 4.3.2 *Suppose that all the assumptions of Theorem 4.3.1 are satisfied. Then*

$$\check{\alpha}_{\ell k} = \frac{1}{10\chi_k e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n \sigma_\ell^k \log n} \tag{4.38}$$

is strictly feasible and (4.34) also holds for $\alpha = \check{\alpha}_{\ell k}$.

Proof: By (4.31) one has

$$\hat{\alpha}_{\ell k} \geq \check{\alpha}_{\ell k} = \frac{1}{10\chi_k e\left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n \sigma_\ell^k \log n}.$$

This completes the proof. □

In what follows we further analyze whether $\check{\alpha}_{\ell k}$ can be used as the default step size, or if it has to be somehow reduced. To do so, we need to derive an upper bound for $\sigma_{\ell k}$. This bound also allows us to find lower and upper bounds for the complementarity gap, which is important to control the iterates and to derive the iteration bound. For simplicity we omit the superscripts and subscripts k in this lemma.

Lemma 4.3.5 *Let $n \geq 2,\ \tau \geq 2$. If $\Psi_\ell(v) \leq \dfrac{(\tau-1)\,n}{2}$, then*

$$\sigma_\ell \leq \frac{3}{2} e \tau^{1+\frac{1}{\log n}} n \log n. \tag{4.39}$$

Proof: To derive an upper bound for σ_ℓ we need to find the optimal value of the problem:

$$\begin{aligned} \max \quad & \sigma_\ell^2 \\ \text{s.t.} \quad & \Psi_\ell(v) \leq \frac{(\tau-1)\,n}{2}. \end{aligned}$$

This can be done by solving the following system of optimality conditions:

$$\begin{aligned} \psi_\ell'(v_i)\,(\psi_\ell''(v_i) - \lambda) &= 0, \quad i = 1, \ldots, n, \\ \Psi_\ell(v) &\leq \frac{(\tau-1)\,n}{2}, \\ \lambda \left(\Psi_\ell(v) - \frac{(\tau-1)\,n}{2} \right) &= 0. \end{aligned} \tag{4.40}$$

It is obvious that $\lambda = 0$ if and only if $v = e$, which implies $\sigma_\ell = 0$. Now, if $v \neq e$, and therefore $\lambda \neq 0$, then the following two cases may happen in the solution of system (4.40).

(i) $v_1 = \ldots = v_{n-j} = 1,\ v_{n-j+1} = \ldots = v_n > 1$,

(ii) $v_1 = \ldots = v_{n-j} = 1,\ v_{n-j+1} = \ldots = v_n < 1$.

For case (i) we have

$$\frac{j v_n^2 - j}{2} - \frac{n}{\log n} \quad \leq \quad \Psi_\ell(v) = \frac{(\tau-1)\,n}{2}.$$

This implies $v_n^2 \leq \dfrac{n\log n(\tau-1)+2n}{j\log n}+1$. Thus,

$$\sigma_\ell^2 = \sum_{i=1}^{n}\left(v_i - v_i^{-1-\log n}\right)^2 = j\left(v_n - v_n^{-1-\log n}\right)^2 \leq jv_n^2 \leq \frac{n\tau\log n+2n}{\log n}.$$

For case (ii) we have

$$\frac{jv_n^{-\log n}-j}{\log n} - \frac{n}{2} \leq \Psi_\ell(v) = \frac{(\tau-1)\,n}{2},$$

that implies $v_n^{-\log n} \leq \frac{n\tau\log n+2j}{2j}$. Thus,

$$\begin{aligned}\sigma_\ell^2 &= j\left(v_n - v_n^{-1-\log n}\right)^2 \leq jv_n^{-2-2\log n} = j\left(v_n^{-\log n}\right)^{\frac{2+2\log n}{\log n}} \\ &\leq j\left(\frac{n\tau\log n+2j}{2j}\right)^{\frac{2+2\log n}{\log n}} \leq \left(\frac{n\tau\log n+2}{2}\right)^{\frac{2+2\log n}{\log n}}.\end{aligned}$$

This upper bound is larger than the bound derived in (i); therefore we have the following upper bound for σ_ℓ,

$$\sigma_\ell \leq \frac{3}{2}e\tau^{1+\frac{1}{\log n}}n\log n.$$

This completes the proof. □

In the sequel, we will see that in the worst case one might not be able to determine a unique lower bound for the step size in the entire neighborhood i.e., in some parts of the neighborhood one can have much better lower bound of the step size than the other parts. It will be also motivated why for an iterate in $\mathcal{IN}_\ell(\tau,\beta)$ we use system (4.11) with $\chi_k = 1$, if $\dfrac{\mu_g^k}{\left(\mu_\ell^t\right)^k} \leq \tau + \eta_1(n,\tau,\sigma_\ell)$, and with $\chi_k = 2$ if $\dfrac{\mu_g^k}{\left(\mu_\ell^t\right)^k} > \tau + \eta_1(n,\tau,\sigma_\ell)$.

Lemma 4.3.6 *Let* $n \geq 2,\ \tau \geq 2,\ (x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta),\ \dfrac{\mu_g^k}{(\mu_\ell^h)^k} \leq 0.99\tau$ *and* $(\Delta x^k, \Delta y^k, \Delta s^k)$ *be the solution of system (4.11) with* $\mu = (\mu_\ell^t)^k$ *and* χ_k *is chosen as in Algorithm 4.2. Then*

$$\Phi_\ell(x(\check{\alpha}_{\ell k}), s(\check{\alpha}_{\ell k}), \mu_g(\check{\alpha}_{\ell k})) \leq \eta_\ell(n, \tau).$$

Proof: For $\alpha = \check{\alpha}_{\ell k}$ let

$$g_1(\alpha) = \Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*) = \frac{1}{2}\frac{\mu_\ell^t}{\mu_\ell^*}\|v(\alpha)\|^2 - \frac{n}{2} + \frac{1}{\log n}\left(\frac{\mu_\ell^*}{\mu_\ell^t}\right)^{\frac{\log n}{2}} \left\|v(\alpha)^{\frac{-\log n}{2}}\right\|^2 - \frac{n}{\log n}.$$

Note that, from the definition of μ_ℓ^* and μ_ℓ^t, we know that

$$\frac{\mu_\ell^t}{\mu_\ell^*}\|v(0)\|^2 = \left(\frac{\mu_\ell^*}{\mu_\ell^t}\right)^{\frac{\log n}{2}} \left\|v(0)^{\frac{-\log n}{2}}\right\|^2 = n(\tau_0)^{\frac{\log n}{2+\log n}},$$

where $\tau_0 := \frac{\mu_g}{\mu_\ell^h}$. On the other hand, we have

$$\|v(\alpha)\|^2 = \|v\|^2 + \alpha v^T(d_x + d_s) + \alpha^2 d_x^T d_s \leq \|v\|^2 + \alpha^2 d_x^T d_s,$$

because $v^T(d_x + d_s) = \left\|v^{\frac{-\log n}{2}}\right\|^2 - \|v\|^2 \leq 0$. It is easy to verify that

$$\left\|v(\alpha)^{\frac{-\log n}{2}}\right\|^2 \leq (1 - \alpha v_{\min}^{-1}\sigma_1)^{-\log n}\left\|v^{\frac{-\log n}{2}}\right\|^2.$$

We also have $(1-\alpha v_{\min}^{-1}\sigma_1)^{-\log n} \leq 1.0005$ and $\dfrac{\alpha^2\left|d_x^T d_s\right|}{2} \leq 10^{-4}$. Then we have

$$g_1(\alpha) \leq \frac{n(\tau_0)^{\frac{\log n}{2+\log n}}}{2} - \frac{n}{2} + 10^{-4} + \frac{1.0005(\tau_0)^{\frac{\log n}{2+\log n}}n}{\log n} - \frac{n}{\log n}.$$

Since $\mu_\ell^*(\alpha)$ is the global minimizer of $\Phi_\ell(x(\alpha), s(\alpha), \mu)$, then one has $\Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*(\alpha)) \leq \Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*)$. This implies that the next iterate satisfies the statement of the lemma if

$$\frac{(\tau_0)^{\frac{\log n}{2+\log n}}}{2} + 10^{-4} + \frac{1.0005(\tau_0)^{\frac{\log n}{2+\log n}}}{\log n} \leq \frac{\tau^{\frac{\log n}{2+\log n}}}{2} + \frac{\tau^{\frac{\log n}{2+\log n}}}{\log n}.$$

This implies that $\tau_0 = \tau_2\tau$, where $\tau_2 \leq 0.99$ that completes the proof. $\square$

The following lemma deals with the case when $\mu_g^k = \tau(\mu_\ell^h)^k$.

Lemma 4.3.7 *Let $n \geq 2,\ \tau \geq 2,\ (x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta),\ \dfrac{\mu_g^k}{(\mu_\ell^h)^k} = \tau$ and let $(\Delta x^k, \Delta y^k, \Delta s^k)$ be the solution of system (4.11) with $\mu = (\mu_\ell^t)^k$ and χ_k is chosen as in Algorithm 4.2. Then $\tilde{\alpha}_{\ell k} = \dfrac{1}{10\chi_k e\tau\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n\sigma_\ell^k \log n}$ is strictly feasible and*

$$\Phi_\ell(x(\tilde{\alpha}_{\ell k}), s(\tilde{\alpha}_{\ell k}), \mu_g(\tilde{\alpha}_{\ell k})) \leq \eta_\ell(n, \tau).$$

Proof: From the assumption that $\mu_g = \tau\mu_\ell^h = \tau\mu_\ell^t$ one has $\mu_\ell^* = \tau^{\frac{2}{2+\log n}}\mu_\ell^h$. Now, let us define

$$\begin{aligned}
g(\alpha) :&= \Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*) - \Phi_\ell(x, s, \mu_\ell^*) \\
&= \frac{1}{2}\frac{\mu_\ell^h}{\mu_\ell^*}\left(v^T(d_x + d_s)\alpha + \alpha^2 d_x^T d_s\right) + \left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} \frac{\left\|v(\alpha)^{\frac{-\log n}{2}}\right\|^2}{\log n} \\
&\qquad - \left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} \frac{\left\|v^{\frac{-\log n}{2}}\right\|^2}{\log n} \\
&\leq \frac{1}{2}\frac{\mu_\ell^h}{\mu_\ell^*}\left(v^T(d_x + d_s)\alpha + \alpha^2 d_x^T d_s\right) + \frac{1}{2\log n}\left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} \left\|(v + \alpha d_x)^{\frac{-\log n}{2}}\right\|^2 \\
&\quad + \frac{1}{2\log n}\left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} \left\|(v + \alpha d_s)^{\frac{-\log n}{2}}\right\|^2 - \left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} \frac{\left\|v^{\frac{-\log n}{2}}\right\|^2}{\log n} := g_1(\alpha),
\end{aligned}$$

where the last inequality follows from the fact that $\left\|v(\alpha)^{\frac{-\log n}{2}}\right\|^2$ satisfies condition SR.2 of Definition 2.6.1. From the definition of v, we have $\left\|v^{\frac{-\log n}{2}}\right\|^2 = n$. The inequality $\sigma_\ell^2 \geq 2\Phi_\ell(x, s\mu_\ell^t) = (\tau - 1)n$ from Proposition 2.6.2 further implies

$$\left\|v^{-1-\log n}\right\|^2 - \left\|v^{\frac{-\log n}{2}}\right\|^2 \geq 0.$$

This inequality, together with the fact

$$\|v\|^2 = \tau \left\| v^{\frac{-\log n}{2}} \right\|^2,$$

gives

$$g_1^{'}(0) = \frac{1}{2}\frac{\mu_\ell^h}{\mu_\ell^*} v^T(d_x + d_s) - \frac{1}{2}\left(\frac{\mu_\ell^*}{\mu_\ell^h}\right)^{\frac{\log n}{2}} (v^{-1-\log n})^T(d_x + d_s) \leq -\frac{\sigma_\ell^2}{2\chi_k \tau^{\frac{2}{2+\log n}}}.$$

Analogous to Theorem 4.3.1 one has the following upper bound for the second derivative of $g(\alpha)$

$$g_1^{''}(\alpha) \leq \frac{\sigma_1^2}{2}\left(\tau^{\frac{\log n}{2+\log n}}(1+\log n)(v_{\min} - \alpha\sigma_1)^{-2-\log n} + \tau^{\frac{-2}{2+\log n}}\right).$$

Using the facts that

$$g_1(\alpha) = g_1(0) + g_1^{'}(0)\alpha + \int_0^\alpha \int_0^\xi g_1^{''}(\eta) d\eta d\xi$$

and $g_1(0) = 0$ one has

$$\begin{aligned} g_1(\alpha) \ &\leq \frac{-\sigma_\ell^2 \alpha}{2\chi_k \tau^{\frac{2}{2+\log n}}} \\ &+ \frac{\sigma_1^2}{2}\int_0^\alpha \int_0^\xi \left(\tau^{\frac{\log n}{2+\log n}}(1+\log n)(v_{\min} - \eta\sigma_1)^{-2-\log n} + \tau^{\frac{-2}{2+\log n}}\right) d\eta d\xi \\ &:= g_2(\alpha). \end{aligned}$$

The function $g_2(\alpha)$ is a continuously differentiable function of α. Let us denote its global minimum by $\alpha_{\ell k}$; then analogous to the proof of Theorem 4.3.1 one can derive the following lower bound for the maximum feasible step size

$$\alpha_{\ell k} \geq \tilde{\alpha}_{\ell k} = \frac{1}{10\chi_k e\tau \left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n \sigma_\ell^k \log n}.$$

This completes the proof of the lemma. □

In the following Lemma we show that for a certain upper bound of the step size the inequality corresponding to the residuals in the definition of $\mathcal{IN}_\ell(\tau,\beta)$ holds. The proof of the lemma explains why the Newton system is changed in a certain part of the neighborhood.

Lemma 4.3.8 *Let $n \geq 2,\ \tau \geq 2,\ (x^k,s^k) \in \mathcal{IN}_\ell(\tau,\beta)$ and $(\Delta x^k, \Delta y^k, \Delta s^k)$ be the solution of system (4.11) with $\mu = \left(\mu_\ell^t\right)^k$, and χ_k is chosen as specified in Algorithm 4.2. Then for $\alpha_k \leq \alpha_{\ell k}^* := \frac{1}{10e\tau\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n\sigma_\ell^k \log n}$ as the step size one has*

$$\frac{\left\|(r_b^{k+1}, r_c^{k+1})\right\|}{\mu_g^{k+1}} \leq \beta \frac{\|(r_b^0, r_c^0)\|}{\mu_g^0}. \tag{4.41}$$

Proof: Since after each step with α_k as the step size one has

$$r_b^{k+1} = (1-\alpha_k) r_b^k \quad \text{and} \quad r_c^{k+1} = (1-\alpha_k) r_c^k,$$

it suffices to show that

$$\mu_g(\alpha_k) \geq (1-\alpha_k)\mu_g^k.$$

Let us consider the case when $\alpha_k = \alpha_{\ell k}^*$. Then from the third part of Theorem 4.1.1, we know that

$$\mu_g(\alpha_{\ell k}^*) = \mu_g^k \left(1 - \frac{\alpha_{\ell k}^*}{\chi_k} + \frac{\alpha_{\ell k}^* \left(\left(\mu_\ell^t\right)^k\right)^{\frac{2+\log n}{2}}}{\chi_k \mu_g^k \left(\left(\mu_\ell^h\right)^k\right)^{\frac{\log n}{2}}} + \left(\alpha_k^*\right)^2 \frac{(\Delta x^k)^T \Delta s^k}{n\mu_g^k} \right). \tag{4.42}$$

Let us define

$$Z(\alpha) := \frac{\left(\left(\mu_\ell^t\right)^k\right)^{\frac{2+\log n}{2}}}{\chi_k \mu_g^k \left(\left(\mu_\ell^h\right)^k\right)^{\frac{\log n}{2}}} + \alpha \frac{(\Delta x^k)^T \Delta s^k}{n\mu_g^k}. \tag{4.43}$$

Using (4.8) we have

$$Z(\alpha_k) \ \geq \ \frac{(2+\tau\log n)\left(\mu_\ell^t\right)^k - \mu_g^k \log n}{2\chi_k \mu_g^k} - \alpha_k \frac{|(\Delta x^k)^T \Delta s^k|}{n\mu_g^k}.$$

Applying the assumptions of the lemma and relations (4.30) and (4.39) to this inequality with $\chi_k = 1$ one has

$$Z(\alpha_k) \geq 0,$$

if $\dfrac{\mu_g^k}{(\mu_\ell^t)^k} \leq \tau + \eta_1(n, \tau, \sigma_\ell^k)$, where

$$\eta_1(n, \tau, \sigma_\ell^k) = \frac{10e\tau \left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n \log n - \tau C_2^2 \sigma_\ell^k}{5e\tau \left(\tau + \frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n (\log n)^2 + C_2^2 \sigma_\ell^k}. \tag{4.44}$$

This implies the statement of the theorem when $\dfrac{\mu_g^k}{(\mu_\ell^t)^k} \leq \tau + \eta_1(n, \tau, \sigma_\ell^k)$. Now let us consider the case when $\tau + \eta_1(n, \tau, \sigma_\ell^k) < \dfrac{\mu_g^k}{(\mu_\ell^t)^k}$. By letting $\chi_k = 2$ in (4.42) and using (4.43) one has

$$\mu_g(\alpha_k) = \mu_g^k \left(1 - \frac{\alpha_k}{2} + \alpha_k Z(\alpha_k)\right).$$

Therefore, to show that $\mu_g(\alpha_k) \geq (1 - \alpha_k)\mu_g^k$, it suffices to have $Z(\alpha_k) \geq -\frac{1}{2}$. Since $\dfrac{\mu_g^k}{(\mu_\ell^t)^k} \leq \tau + \dfrac{2}{\log n}$, the inequality $Z(\alpha_k) \geq -\frac{1}{2}$ follows easily from (4.30), (4.39) and from the definition of α_k. Finally, for any $\alpha_k \leq \alpha_{\ell k}^*$ one has

$$\begin{aligned} Z(\alpha_k) &\geq \frac{(2 + \tau \log n)\,(\mu_\ell^t)^k - \mu_g^k \log n}{2\chi_k \mu_g^k} - \alpha_k \frac{|(\Delta x^k)^T \Delta s^k|}{n\mu_g^k} \\ &\geq \frac{(2 + \tau \log n)\,(\mu_\ell^t)^k - \mu_g^k \log n}{2\chi_k \mu_g^k} - \alpha_{\ell k}^* \frac{|(\Delta x^k)^T \Delta s^k|}{n\mu_g^k}. \end{aligned}$$

that completes the proof of the lemma. □

Remark 4.3.1 *From now on, we call a step good if the maximum step size α_k is greater than or equal to $\alpha_{\ell k}^*$ in $\mathcal{IN}_\ell(\tau, \beta)$ for which inequality (4.34) holds.*

Remark 4.3.2 *If after a step the new iterate is in $\mathcal{IN}_\ell(\tau,\beta)$ neighborhood with $\frac{\mu_g^k}{(\mu_\ell^h)^k} \leq 0.99\tau$ or $\frac{\mu_g^k}{(\mu_\ell^h)^k} = \tau$, then the results of Lemmas 4.3.6, 4.3.7 and 4.3.8 give an estimate of the step size that leads to the polynomiality of the algorithm. However the algorithm might get to a point for which $0.99\tau < \frac{\mu_g^k}{(\mu_\ell^h)^k} < \tau$, because if we would make a larger step toward the boundary of the neighborhood, then inequality (4.34) might be violated before we hit the boundary. In this case the algorithm might take a slightly shorter step that will be analyzed in the sequel.*

In what follows we discuss the cases where $0.99\tau < \frac{\mu_g^k}{(\mu_\ell^h)^k} < \tau$.

Case 1: If the maximum step size of the next iterate is greater than or equal to $\alpha_{\ell k}^*$. Then the algorithm makes a good step.

Case 2: The algorithm might make a step with $\alpha_k < \alpha_{\ell k}^*$ for which $\frac{\mu_g^k(\alpha_k)}{\mu_\ell^h(\alpha_k)} \leq 0.99\tau$. Then by Lemma 4.3.5 we know that in the next iterate, we can make a good step.

Case 3: The algorithm might make a step with $\alpha_k < \alpha_{\ell k}^*$ for which $\frac{\mu_g^k(\alpha_k)}{\mu_\ell^h(\alpha_k)} = \tau$. Then by Lemma 4.3.6 we know that the next step will be again a good one.

Case 4: The algorithm might make a step with $\alpha_k < \alpha_{\ell k}^*$ for which $0.99\tau < \frac{\mu_g^k(\alpha_k)}{\mu_\ell^h(\alpha_k)} < \tau$.

Our next lemma shows that Case 4 can not happen.

Lemma 4.3.9 *Suppose that for the current iterate in $\mathcal{IN}_\ell(\tau,\beta)$, $0.99\tau < \frac{\mu_g^k}{(\mu_\ell^h)^k} < \tau$. If the algorithm does a step with $\alpha_k < \alpha_{\ell k}^*$, then Case 4 can not happen.*

Proof: Let us suppose that the algorithm makes a step with $\alpha_k < \alpha_{\ell k}^*$. Then by inequality (4.34) and Lemma 4.3.8 the only thing that can restrict the step size to be less than $\alpha_{\ell k}^*$ is inequality (4.13). In this situation one can conclude

that equality holds in (4.13) that implies $\frac{\mu_g^k(\alpha_k)}{\mu_\ell^h(\alpha_k)} = \tau$. Therefore, Case 4 can not happen. This completes the proof of the lemma. □

4.3.3 Iteration Complexity

The following Lemma is used to estimate the reduction of μ_ℓ^t after a good step. The proof is a direct consequence of Theorem 4.3.1 and Lemma 4.3.1.

Lemma 4.3.10 *Suppose that all the assumptions of Theorem 4.3.1 are satisfied, then*

$$\begin{aligned}\Phi_\ell\left(x(\alpha_{\ell k}^*), s(\alpha_{\ell k}^*), \left(\mu_\ell^t\right)^k\right) &\leq \Phi_\ell\left(x^k, s^k, \left(\mu_\ell^t\right)^k\right) \\ &- \frac{\sqrt{\tau-1}}{80e\tau\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n^{\frac{1}{2}}\log n}. \quad (4.45)\end{aligned}$$

By summarizing the previous results, we have:

Theorem 4.3.2 *Let $n \geq 2$, $\tau \geq 2$, $(x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$, and $(\Delta x^k, \Delta y^k, \Delta s^k)$ be the solution of system (4.11) with $\mu = (\mu_\ell^t)^k$ and χ_k is chosen as specified in Algorithm 4.2. Then, for the good step, the default step size $\alpha_{\ell k}^*$ is strictly feasible and $(x(\alpha_{\ell k}^*), s(\alpha_{\ell k}^*)) \in \mathcal{IN}_\ell(\tau, \beta)$. In the worst case each good step might be followed by at most one short step for which there exists a positive step size $\alpha_k \leq \alpha_{\ell k}^*$ such that $(x(\alpha_k), s(\alpha_k)) \in \mathcal{IN}_\ell(\tau, \beta)$.*

To obtain an upper bound for the total number of iterations of Algorithm 4.2, we need to investigate the growth behavior of $\Phi_\ell(x^k, s^k, (\mu_\ell^t)^k)$ w.r.t. $(\mu_\ell^t)^k$. Since the step size in a short step might be much smaller than the step size in a good step, therefore, we do not take into account how much μ_ℓ^t reduces at a short step. We just use the fact that it reduces at each short step as well.

The following technical lemma is an important tool to prove the polynomial complexity of our new algorithm. For general q we had a similar lemma in Chapter 3, but here the upper bound is sharpened.

Lemma 4.3.11 *Let* $v_+ = \dfrac{v}{\sqrt{1-\theta}}$ *for some* $\theta \in (0,1)$. *Then we have:*

$$\Psi_\ell(v_+) \leq \frac{\Psi_\ell(v)}{1-\theta} + \frac{2n\theta}{(1-\theta)}.$$

Proof: From the definition of the proximity function we have

$$\begin{aligned}
\Psi_\ell(v_+) &= \frac{\|v_+\|^2 - n}{2} + \frac{\left\|v_+^{\frac{-\log n}{2}}\right\|^2 - n}{\log n} \\
&= \frac{\frac{1}{1-\theta}\|v\|^2 - n}{2} + \frac{(1-\theta)^{\frac{\log n}{2}}\left\|v^{\frac{-\log n}{2}}\right\|^2 - n}{\log n} \\
&= \frac{1}{1-\theta}\left(\frac{\|v\|^2 - n}{2} + \frac{\left\|v^{\frac{-\log n}{2}}\right\|^2 - n}{\log n}\right) + \frac{n\theta}{2(1-\theta)} \\
&\quad + \frac{n\theta}{\log n(1-\theta)} + \left((1-\theta)^{\frac{\log n}{2}} - \frac{1}{1-\theta}\right)\frac{\left\|v^{\frac{-\log n}{2}}\right\|^2}{\log n} \\
&\leq \frac{\Psi_\ell(v)}{1-\theta} + \frac{2n\theta}{(1-\theta)}.
\end{aligned}$$

This completes the proof of the lemma. □

The following theorem ensures the sufficient reduction of μ_ℓ^t at a good step.

Theorem 4.3.3 *Let* $n \geq 2,\ \tau \geq 2,\ (x^k, s^k) \in \mathcal{IN}_\ell(\tau, \beta)$, *and* $(\Delta x^k, \Delta y^k, \Delta s^k)$ *be the solution of system (4.11) with* $\mu = (\mu_\ell^t)^k$ *and* χ_k *chosen as it is specified in Algorithm 4.2. Then after each good step one has*

$$\Phi_\ell\left(x(\alpha_{\ell k}^*), s(\alpha_{\ell k}^*), (1-\theta_k)\left(\mu_\ell^t\right)^k\right) \leq \Phi_\ell\left(x^k, s^k, \left(\mu_\ell^t\right)^k\right),$$

where

$$\theta_k = \frac{\sqrt{\tau-1}}{40e\tau(\tau+3)\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n^{\frac{3}{2}} \log n}. \tag{4.46}$$

Proof: Using Lemma 4.3.11, it suffices to choose θ_k satisfying the following inequality

$$\Phi_\ell\left(x(\alpha_k^*), s(\alpha_k^*), \left(\mu_t^\ell\right)^k\right) + 2n\theta_k \leq (1-\theta_k)\Phi_\ell\left(x^k, s^k, \left(\mu_t^\ell\right)^k\right).$$

Considering (4.45), this inequality is certainly satisfied if

$$\theta_k \Phi_\ell\left(x^k, s^k, \left(\mu_t^\ell\right)^k\right) + 2n\theta_k \leq \frac{\sqrt{\tau-1}}{80e\tau\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n^{\frac{1}{2}} \log n}.$$

By using the fact that $\Phi_\ell\left(x^k, s^k, \left(\mu_t^\ell\right)^k\right) = \frac{(\tau-1)n}{2}$, we can rewrite this inequality as:

$$\theta_k\left(\frac{(\tau-1)n}{2} + 2n\right) \leq \frac{\sqrt{\tau-1}}{80e\tau\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n^{\frac{1}{2}} \log n},$$

or

$$\theta_k \leq \frac{\sqrt{\tau-1}}{40e\tau(\tau+3)\left(\tau+\frac{2}{\log n}\right)^{\frac{1}{\log n}} C_3 n^{\frac{3}{2}} \log n},$$

which completes the proof of the theorem. □

By the choice of $(\mu_\ell^t)^k$ we know that the proximity function value $\Phi_\ell(x^k, s^k, (\mu_\ell^t)^k)$ remains invariant for all iterates. Let us denote by $\mu_\ell^t(\alpha_k)$ the value of $(\mu_\ell^t)^k$ after one step, thus

$$\Phi_\ell(x^k, s^k, (\mu_\ell^t)^k) = \Phi_\ell\left(x(\alpha_k), s(\alpha_k), \mu_\ell^t(\alpha_k)\right) = \frac{(\tau-1)n}{2}. \tag{4.47}$$

Corollary 4.3.3 *Under the assumptions of Theorem 4.3.3, we have*

$$\mu_\ell^t(\alpha_{\ell k}^*) \;\leq\; (1-\theta_k)\,(\mu_\ell^t)^k,$$

where θ_k is given by (4.46).

Proof: Using (4.47), Theorem 4.3.3 and the quasiconvexity of the proximity measure w.r.t. μ, one can easily conclude the statement. □

Corollary 4.3.4 *By Theorem 4.3.1 one has a certain reduction of the proximity measure independent of where the iterates are in $\mathcal{IN}_\ell(\tau,\beta)$. Therefore, one can get analogous reduction of $\mu_\ell^t(\alpha_k)$ after each short step. Since we do not use it explicitly in the final iteration number derivation, we omitted it here.*

Now we are ready to give the complexity of Algorithm 4.2.

Theorem 4.3.4 *Let $n \geq 2,\ \tau \geq 2,\ (x^0,s^0) \in \mathcal{N}_\ell(\tau,\beta)$, and $t_0 = \max\left\{1, \frac{\|(r_b^0,r_c^0)\|}{\mu_g^0}\right\}$. Then after at most*

$$O\left(n^{\frac{3}{2}}\log n \log \frac{\beta t_0\left(\tau+\frac{2}{\log n}\right)n}{\epsilon}\right)$$

iterations Algorithm 4.2 terminates with a solution satisfying $(x^k)^T s^k \leq \epsilon$ and $\|(r_b^k,r_c^k)\| \leq \epsilon$.

Proof: Using Lemma II.17 [50], Corollary 4.3.3 and Corollary 4.3.4 we need at most

$$\left\lceil \frac{2}{\theta}\log \frac{\beta t_0\left(\tau+\frac{2}{\log n}\right)n}{\epsilon}\right\rceil$$

iterations to have $(\mu_\ell^t)^k \leq \frac{\epsilon}{\beta t_0 n\left(\tau+\frac{2}{\log n}\right)}$, where θ is given by (4.46). Now, Lemma 4.1.1 implies that $\mu_g^k \leq \left(\tau+\frac{2}{\log n}\right)(\mu_\ell^t)^k \leq \frac{\epsilon}{n\beta t_0}$. Thus, we have $(x^k)^T s^k \leq \frac{\epsilon}{\beta t_0} \leq \epsilon$. Using Lemma 4.3.8, we also have $\|(r_b^k,r_c^k)\| \leq \epsilon$. □

Chapter 5

An Adaptive Predictor-Corrector IPM

In this chapter we present a feasible adaptive SR-proximity based predictor-corrector (SR-PC) algorithm. The new algorithm differs both in predictor and corrector steps from its analogues. For certain values of q, the barrier degree, we improve the best so far iteration complexity for predictor-corrector algorithms. The quadratic convergence of the algorithm also is proved. Limited computational result based on our algorithm using the McIPM software package [73] will be reported in Chapter 8. The results we present in this chapter mostly appeared in [54].

5.1 The Proximity Measure and Its Properties

Throughout this chapter the proximity measure is the same as the one in Chapter 4, namely the function given by (4.2). The main difference between our adaptive algorithm and the traditional predictor-corrector algorithms are the updating strategy which follows the same scheme that we explained in Chapter 3 and multiple choices in the predictor step that we will discus in the sequel. For simplicity of analysis we propose a feasible algorithm. The algorithm operates in a SR neighborhood that almost matches a negative infinity norm neighborhood as it is defined in Chapter 4 by excluding infeasibility. Namely, we have the neighborhood

$$\mathcal{N}_\ell(\tau, n) := \left\{(x, y, s) \in \mathcal{F}^0 \middle| \ \Phi_\ell(x, s, \mu_g) \leq \eta_\ell(n, \tau)\right\}, \tag{5.1}$$

where $\tau = \dfrac{1}{1-\rho}$, $\rho \in (0, 1)$ is the parameter of the neighborhood $\mathcal{N}_\infty^-(\rho)$ and $\eta_\ell(n, \tau) = \dfrac{\left(\tau^{\frac{\log n}{2}} - 1\right) n}{\log n}$ is a positive function that depends on a constant τ and the dimension of the underlying problem. As proved in Chapter 4, for this specific parameter one has $\mathcal{N}_\infty^-(\rho) \subseteq \mathcal{N}_\ell(\tau, n)$, and the reverse inequality also holds for infinity norm neighborhood with larger parameter.

In the original predictor-corrector methods one uses the primal-dual affine scaling direction in the predictor step, which is the solution of the following system of equations:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= -xs. \end{aligned} \tag{5.2}$$

It is well known that close to optimality, the step size of the affine scaling direction converges to one [50]. This fact implies that close to optimality the affine scaling direction is the best choice. In what follows in the first iterations we change this scheme and we use SR direction for the predictor step. To be more precise, we compare the duality gap reduction of the affine scaling direction with the warranted duality gap reduction of the SR direction and then we move in the direction that gives the larger duality gap reduction.

For the SR direction in the predictor step we use the same adaptive scheme that is presented in Section 4.1. The following simple assumption is crucial for the rest of this chapter.

Without loss of generality we may always assume that after a predictor step the new iterate is on the boundary of the SR neighborhood, or equivalently $\Phi_\ell(x, s, \mu_g) = \eta_\ell(n, \tau)$.

The relation between μ_g and μ_ℓ^* given by (4.3) plays an important role in the analysis of our new algorithm that is given in the following lemma.

Lemma 5.1.1 *Let* μ_ℓ^* *be the global minimum of the proximity measure as it is defined in (4.3). If* $\Phi_\ell(x, s, \mu_g) = \eta_\ell(n, \tau)$, *then* $\mu_\ell^* = \tau^{\frac{-\log n}{2+\log n}} \mu_g$.

Proof: From the definition of μ_ℓ^* one has

$$\mu_\ell^* = \left(\mu_g (\mu_\ell^h)^{\frac{\log n}{2}} \right)^{\frac{2}{2+\log n}}.$$

Since by the assumption of the lemma $\Phi_\ell(x, s, \mu_g) = \eta_\ell(n, \tau)$, then we have $\mu_g = \tau \mu_\ell^h$. This implies that

$$\mu_\ell^* = \tau^{\frac{-\log n}{2+\log n}} \mu_g$$

that completes the proof of the lemma. □

Unlike the previous chapters here we have more possibilities to choose the search direction rather than the one which is induced by the proximity function (4.2). The search directions throughout this chapter are based on the SR-proximity function

$$\Psi_q(v) = \Phi_q(x, s, \mu) := \frac{e^T v^2 - n}{2} + \frac{e^T v^{1-q} - n}{q-1}, \tag{5.3}$$

where $1 \leq q \leq 1 + \log n$.

In predicting the reduction of duality gap the following technical lemma is useful.

Lemma 5.1.2 *Let $1 < q \leq 1 + \log n$, then the global minimum of $\Psi_q(v)$ in (5.3), μ_q^*, is decreasing w.r.t. q.*

Proof: By the definition of μ_q^* from Proposition 3.1.1 on page 36, the first derivative of μ_q^* as a function of q is equal to

$$\left(\mu_q^*(q)\right)' = -2\mu_q^* \frac{\log\left(\frac{x^T s}{\left((x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)}\right)}{(q+1)^2} + \mu_q^* \frac{\left(\sum_{i=1}^n (x_i s_i)^{\frac{1-q}{2}} \log(x_i s_i)\right)}{\left((q+1)(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)}.$$

It suffices to prove that $\left(\mu_q^*(q)\right)' \leq 0$ for all $q > 1$, which is equivalent to

$$\frac{q+1}{2} \frac{\left(\sum_{i=1}^n (x_i s_i)^{\frac{1-q}{2}} \log(x_i s_i)\right)}{\left((x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)} \leq \log\left(\frac{x^T s}{\left((x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)}\right),$$

which is equivalent to

$$\frac{q+1}{1-q} \frac{\left(\sum_{i=1}^n (x_i s_i)^{\frac{1-q}{2}} \log(x_i s_i)^{\frac{1-q}{2}}\right)}{\left((x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)} \leq \log\left(\frac{x^T s}{\left((x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}\right)}\right). \tag{5.4}$$

By using Jensen's inequality for the function $u\log(u)$ for the left hand side of (5.4) one has

$$\frac{q+1}{1-q}\frac{\left(\sum_{i=1}^{n}(x_is_i)^{\frac{1-q}{2}}\log(x_is_i)^{\frac{1-q}{2}}\right)}{\left((x^{\frac{1-q}{2}})^Ts^{\frac{(1-q)}{2}}\right)} \leq \frac{q+1}{1-q}\log\left(\frac{\left((x^{\frac{1-q}{2}})^Ts^{\frac{1-q}{2}}\right)}{n}\right).$$

Therefore, (5.4) holds if

$$\frac{q+1}{1-q}\log\left(\frac{\left((x^{\frac{(1-q)}{2}})^Ts^{\frac{(1-q)}{2}}\right)}{n}\right) \leq \log\left(\frac{x^Ts}{\left((x^{\frac{(1-q)}{2}})^Ts^{\frac{(1-q)}{2}}\right)}\right),$$

which is equivalent to $\mu_q^h \leq \mu_q^*$, which is true by Lemma 3.1.5. □

Note that our SR-PC algorithm uses the specific SR-proximity function $\Phi_\ell(x,s,\mu)$ to define the neighborhood, while the search directions are given by the SR-proximity function $\Phi_q(x,s,\mu)$, where $1 \leq q \leq 1+\log n$. Due to the choice of the proximity function, we can write system (2.16) as:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu^{\frac{q+1}{2}}(x^{\frac{1-q}{2}})s^{\frac{1-q}{2}} - xs. \end{aligned} \tag{5.5}$$

Let us denote the solution of system (5.5) by $(\Delta x(\mu), \Delta y(\mu), \Delta s(\mu))$ and let $\mu = \mu_\ell^*$. The following two lemmas discuss the change of the duality gap along the search direction $(\Delta x(\mu_\ell^*), \Delta y(\mu_\ell^*), \Delta s(\mu_\ell^*))$.

Lemma 5.1.3 *Let $(\Delta x(\mu_\ell^*), \Delta y(\mu_\ell^*), \Delta s(\mu_\ell^*))$ be the solution of system (5.5) with $\mu = \mu_\ell^*$ and let $1 \leq q \leq 1+\log n$. Then the relation*

$$x^T\Delta s(\mu_\ell^*) + s^T\Delta x(\mu_\ell^*) \leq 0$$

holds. If $q = 1+\log n$, then equality holds in this relation.

Proof: Using the third equation in (5.5) one has

$$x^T\Delta s(\mu_\ell^*) + s^T\Delta x(\mu_\ell^*) = (\mu_\ell^*)^{\frac{q+1}{2}}(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - x^T s \leq 0,$$

where the last inequality follows from Lemma 5.1.2. This completes the proof of the lemma. □

Corollary 5.1.1 *If $\mu = \mu_\ell^*$, then the duality gap decreases, i.e.,*

$$(x + \alpha\Delta x(\mu_\ell^*))^T(s + \alpha\Delta s(\mu_\ell^*)) = x^T s + \alpha\left(x^T\Delta s(\mu_\ell^*) + s^T\Delta x(\mu_\ell^*)\right) \leq x^T s.$$

When μ_ℓ^t is the target value, analogous to Lemma 5.1.3, we can get the following result.

Lemma 5.1.4 *Let $(\Delta x(\mu_\ell^t), \Delta y(\mu_\ell^t), \Delta s(\mu_\ell^t))$ be the solution of system (5.5), where $\mu = \mu_\ell^t$ is defined by equation (4.8) and let $1 \leq q \leq 1 + \log n$. Then*

$$x^T\Delta s(\mu_\ell^t) + s^T\Delta x(\mu_\ell^t) = (\mu_\ell^t)^{\frac{q+1}{2}}(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}} - x^T s,$$

and

$$(x + \alpha\Delta x(\mu_\ell^t))^T(s + \alpha\Delta s(\mu_\ell^t)) = x^T s\left(1 - \alpha + \frac{(\mu_\ell^t)^{\frac{q+1}{2}}\alpha}{\mu_g(\mu_q^h)^{\frac{q-1}{2}}}\right), \tag{5.6}$$

where

$$\mu_q^h = \left(\frac{n}{(x^{\frac{1-q}{2}})^T s^{\frac{1-q}{2}}}\right)^{\frac{2}{q-1}}$$

as it is defined in Chapter 3 on page 37.

For notational convenience, we define

$$\sigma_{\ell q} = \left(\sum_{i=1}^{n}(v_i - {v_i}^{-1-\log n})(v_i - {v_i}^{-q})\right)^{\frac{1}{2}} \tag{5.7}$$

and

$$\sigma_q = \left\| v - v^{-q} \right\| = \|d_x + d_s\| ,$$

where $1 \leq q \leq 1+\log n$. The following two lemmas specify the relation between $\sigma_{\ell q}$, σ_q and $\|(d_x, d_s)\|$.

Lemma 5.1.5 *For $1 \leq q \leq 1 + \log n$ one has $\sigma_q \leq \sigma_{\ell q}$.*

Proof: If $v_i \geq 1$ then $v_i - {v_i}^{-q} \leq v_i - {v_i}^{-\log n - 1}$, otherwise ${v_i}^{-q} - v_i \leq {v_i}^{-1-\log n} - v_i$, thus $\sigma_q \leq \sigma_{\ell q}$. □

Lemma 5.1.6 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of (5.5), where $\mu = \mu_\ell^*$ as it is defined by (4.3) and $1 \leq q \leq 1 + \log n$. Then*

$$\sigma_{\ell q} \geq \|(d_x, d_s)\| .$$

Proof: From the definition of the search directions we have

$$\sigma_{\ell q}^2 = \sum_{i=1}^{n} (v_i - {v_i}^{-1-\log n})(v_i - {v_i}^{-q}) \geq \sum_{i=1}^{n} (v_i - {v_i}^{-q})(v_i - {v_i}^{-q}) = \|(d_x, d_s)\|^2 ,$$

where the last inequality follows from the orthogonality of d_x and d_s. □

Now in what follows we derive bounds for the smallest component of v.

Lemma 5.1.7 *Suppose that the present iterate (x, s) is on the boundary of the neighborhood $\mathcal{N}_\ell(\tau, n)$, where $\mu = \mu_\ell^*$ as it is defined by (4.3). Then for $\tau \geq 8$ and $n \geq 4$ one has*

$$v_{\min} \geq \exp(-1)\tau^{\frac{-1}{2+\log n}}, \tag{5.8}$$

$$v_{\min}^{1+\log n} \sigma_{\ell q} \geq \theta_2(n, \tau) \exp\left(\frac{q - 1 - \log n}{2}\right), \tag{5.9}$$

where $\theta_2(n, \tau) = \tau^{\frac{-\log n}{2(2+\log n)}} \left(1 - \exp\left(-\frac{1}{2}\right)\right)$.

Proof: From the definition of μ_ℓ^* we have

$$v_{\min}^{-\log n} \leq \left\| v^{\frac{-\log n}{2}} \right\|^2 = \|v\|^2 = \frac{\mu_g}{\mu_\ell^*} n = \tau^{\frac{\log n}{2+\log n}} n$$

that implies (5.8). To prove (5.9), first assume that $v_{\min} \leq \exp(-\frac{1}{4})$, then by using (5.7) we have

$$\begin{aligned} v_{\min}^{1+\log n}\sigma_{\ell q} &\geq v_{\min}^{\frac{-q+1+\log n}{2}}(1-v_{\min}^{2+\log n})^{\frac{1}{2}}(1-v_{\min}^{q+1})^{\frac{1}{2}} \geq \left(1-\exp(-\frac{1}{2})\right) v_{\min}^{\frac{-q+1+\log n}{2}} \\ &\geq \tau^{\frac{q-1-\log n}{2(2+\log n)}} \left(1-\exp\left(-\frac{1}{2}\right)\right) \exp\left(\frac{q-1-\log n}{2}\right) \\ &\geq \tau^{\frac{-\log n}{2(2+\log n)}} \left(1-\exp\left(-\frac{1}{2}\right)\right) \exp\left(\frac{q-1-\log n}{2}\right). \end{aligned}$$

If $v_{\min} \geq \exp(-\frac{1}{4})$, then

$$v_{\min}^{1+\log n}\sigma_{\ell q} \geq \exp\left(-\frac{1}{4}\right)\left(\tau^{\frac{\log n}{2+\log n}} - 2\right)^{\frac{1}{2}} n^{\frac{1}{4}},$$

where the inequality follows from the inequality[1]

$$\sigma_{\ell q} \geq \left\| v - v^{-1} \right\| \geq \left(\frac{\mu_g}{\mu_\ell^*} - 2\right)^{\frac{1}{2}} n^{\frac{1}{2}} = \left(\tau^{\frac{\log n}{2+\log n}} - 2\right)^{\frac{1}{2}} n^{\frac{1}{2}} \tag{5.10}$$

and the equality follows from Lemma 5.1.1. This completes the proof. □

Remark 5.1.1 *If $q = 1 + \log n$ and $\tau \geq 8$, then*

$$v_{\min}^{1+\log n}\sigma_{\ell q} \geq \theta_2(n,\tau).$$

Remark 5.1.2 *If $q = 1$ and $\tau \geq 8$, then*

$$v_{\min}^{1+\log n}\sigma_{\ell q} \geq \frac{\theta_2(n,\tau)}{\sqrt{n}}.$$

The following lemma gives a lower bound for the maximal feasible step size.

[1]This inequality holds when $\tau \geq 8$ and $n \geq 4$.

Lemma 5.1.8 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (5.5) where $1 \leq q \leq 1 + \log n$ and $\mu = \mu_\ell^t$ is defined by (4.8). Then the maximal feasible step size, $\alpha_{\ell q}^{\max}$, satisfies*

$$\alpha_{\ell q}^{\max} \geq \bar{\alpha}_{\ell q} = \frac{1}{\tau^{\frac{1}{2+\log n}} \exp(1)\sigma_{\ell q}}.$$

Proof: Since

$$v(\alpha_{\ell q}^{\max}) = (v + \alpha_{\ell q}^{\max} d_x)^{\frac{1}{2}}(v + \alpha_{\ell q}^{\max} d_s)^{\frac{1}{2}} = v(e + \alpha_{\ell q}^{\max} v^{-1} d_x)^{\frac{1}{2}}(e + \alpha_{\ell q}^{\max} v^{-1} d_s)^{\frac{1}{2}},$$

it is nonnegative whenever

$$e + \alpha_{\ell q}^{\max} v^{-1} d_x \geq 0, \text{ and } \quad e + \alpha_{\ell q}^{\max} v^{-1} d_s \geq 0.$$

These inequalities imply

$$\alpha_{\ell q}^{\max} \geq \frac{1}{\|(v^{-1} d_x, v^{-1} d_s)\|}.$$

We also know that

$$\left\|(v^{-1} d_x, v^{-1} d_s)\right\| \leq \frac{\|(d_x, d_s)\|}{v_{\min}} \leq \tau^{\frac{1}{2+\log n}} \exp(1)\sigma_\ell,$$

which completes the proof of the lemma. □

5.2 Algorithmic Scheme

In this section we present our new adaptive predictor-corrector algorithm. At each iteration we have a predictor step and a corrector step. In the corrector step we recenter the iterate to a smaller neighborhood. In the predictor step we make either an adaptive SR step or an affine scaling step in order to reduce the duality gap as much as possible in the given large neighborhood. In the

predictor step we compare the decrease of the duality gap for the SR and the affine scaling steps. If the reduction of the duality gap for the affine scaling step is larger than the one theory guarantees for the SR step, then we make an affine scaling step. Otherwise we do an adaptive SR step. With this adaptive choice of the predictor step we preserve the best known polynomial complexity of large-update SR-IPMs for the case $q = 1 + \log n$ and quadratic convergence is achieved as well. Our adaptive large neighborhood SR-PC algorithm is as follows.

Algorithm 5.2: Adaptive Large Neighborhood SR-PC-IPM

Input:

A proximity parameter $\tau \geq 8$ and $\eta_\ell(n,\tau) = \frac{(\tau^{\frac{\log n}{2}}-1)n}{\log n}$;

an accuracy parameter $\varepsilon > 0$; $1 < q \leq 1 + \log n$;

$(x,s) = (x^0, s^0)$ such that $\Phi_\ell(x,s,\mu_g) \leq \eta_\ell(n,\tau)$.

begin

while $x^T s \geq \epsilon$ **do**

begin

Corrector step

Solve (5.5) with $\mu = \mu_\ell^*$ and find the maximum step size $\bar{\alpha}$ such that $(x + \bar{\alpha}\Delta x, s + \bar{\alpha}\Delta s) \geq 0$;

Let[a] $\alpha_1 = \max\{\alpha \in (0,\ \bar{\alpha}] \big|\ \Phi_\ell^+ \leq \Phi_\ell - K^*, (x(\alpha), s(\alpha)) \in \mathcal{N}_\ell(\tau,n)\}$;

Set $(x,y,s) = (x(\alpha_1), y(\alpha_1), s(\alpha_1))$.

end

begin

Predictor step

(**Affine scaling step**) Solve (5.2) and choose the maximum step size α such that $(x(\alpha), y(\alpha), s(\alpha)) \in \mathcal{N}_\ell(\tau,n)$.

If[b] $1 - \alpha \leq 1 - \hat{\alpha}_{\ell q} + \hat{\alpha}_{\ell q} \frac{(\mu_\ell^t)^{\frac{q+1}{2}}}{\mu_g (\mu_q^h)^{\frac{q-1}{2}}}$, then accept the affine scaling step.

else

(**SR-step**) Solve (5.5) with $\mu = \mu_\ell^t$ derived from (4.8) and determine the maximum step size α such that $(x(\alpha), y(\alpha), s(\alpha)) \in \mathcal{N}_\ell(\tau,n)$;

end

Set $(x,y,s) = (x(\alpha), y(\alpha), s(\alpha))$.

end

[a]The values of K^* is given in Theorem 5.3.1.

[b]The value of $\hat{\alpha}_{\ell q}$ is given in Theorem 5.3.2 as the warranted step size in the SR step. The duality gap reduction formula is given in Lemma 5.1.4 for the SR step.

5.3 Complexity Analysis

5.3.1 The Corrector Step

In this subsection we estimate the decrease of the proximity function in the corrector step when μ_ℓ^* is the target value, and we compute a lower bound for the step size that guarantees sufficient reduction of the proximity function.

Theorem 5.3.1 *Let us assume that the current iterate is in the neighborhood $\mathcal{N}_\ell(\tau, n)$, and let $(\Delta x, \Delta y, \Delta s)$ be the solution of (5.5) with $\mu = \mu_\ell^*$ and $1 \leq q \leq 1 + \log n$. Then, for*

$$\tilde{\alpha}_{\ell q} = \frac{\theta_2(n,\tau) v_{\min}}{4\sigma_{\ell q} \log n \exp(\frac{-q+1+\log n}{2})} \quad \text{and} \quad \check{\alpha}_{\ell q} = \frac{\theta_2^2(n,\tau)}{16 \log n \exp(-q+1+\log n)},$$

where $\theta_2(n,\tau)$ is defined on page 97 in Lemma 5.1.7, we have

$$\Phi_\ell(x(\tilde{\alpha}_{\ell q}), s(\tilde{\alpha}_{\ell q}), \mu_\ell^*) \leq \Phi_\ell(x, s, \mu_\ell^*) - K^*, \tag{5.11}$$

where

$$K^* = \max\{\tilde{\alpha}_{\ell q}\sigma_{\ell q}^2,\ \check{\alpha}_{\ell q} v(\tilde{\alpha}_{\ell q})_{\min}^{-\log n}\}.$$

Proof: Let us examine the reduction of the proximity value as a function of the step length. For this purpose let us define

$$\begin{aligned} g(\alpha) :&= \Phi_\ell\left(x(\alpha), s(\alpha), \mu_\ell^*\right) - \Phi_\ell(x, s, \mu_\ell^*) \\ &= \frac{\|v(\alpha)\|^2 - n}{2} + \frac{\left\|v(\alpha)^{\frac{-\log n}{2}}\right\|^2 - n}{\log n} - \frac{\|v\|^2 - n}{2} - \frac{\left\|v^{\frac{-\log n}{2}}\right\|^2 - n}{\log n}, \end{aligned}$$

where $v(\alpha) = \sqrt{\dfrac{x(\alpha)s(\alpha)}{\mu_\ell^*}} = (v + \alpha d_x)^{\frac{1}{2}}(v + \alpha d_s)^{\frac{1}{2}}$. After simplifying the expression, and the fact that $(v_i + \alpha(d_x)_i)^{\frac{-\log n}{2}}(v_i + \alpha(d_s)_i)^{\frac{-\log n}{2}}$ satisfies condition SR.2 of Definition 2.6.1, we have

$$g(\alpha) \leq \frac{1}{2} v^T(d_x + d_s)\alpha + \frac{1}{2\log n}\sum_{i=1}^{n}[(v_i + \alpha(d_x)_i)^{-\log n} + (v_i + \alpha(d_s)_i)^{-\log n}]$$

$$-\frac{\left\|v^{\frac{-\log n}{2}}\right\|^2}{\log n} := g_1(\alpha).$$

Using (5.7), the definition of $\sigma_{\ell q}$, the derivative of $g_1(\alpha)$ at zero can be given as

$$g_1{}'(0) = \frac{1}{2}v^T(d_x + d_s) - \frac{1}{2}(v^{-1-\log n})^T(d_x + d_s) = -\frac{\sigma_{\ell q}^2}{2}.$$

For the second derivative, using Lemma 5.1.5, one has the following upper bound

$$g_1^{''}(\alpha) \leq \frac{(1+\log n)\sigma_{\ell q}^2(v_{\min} - \alpha\sigma_{\ell q})^{-2-\log n}}{2}.$$

Now, using the fact that

$$g_1(\alpha) = g_1(0) + g_1^{'}(0)\alpha + \int_0^\alpha \int_0^\xi g_1^{''}(\eta)d\eta d\xi,$$

one has

$$g_1(\alpha) \leq -\frac{\sigma_{\ell q}^2}{2}\alpha + \frac{(1+\log n)\sigma_{\ell q}^2}{2}\int_0^\alpha \int_0^\xi (v_{\min} - \eta\sigma_{\ell q})^{-2-\log n}d\eta d\xi := g_2(\alpha).$$

It is easy to see, via making use of simple calculus, that $g_2(\alpha)$ is convex and twice continuously differentiable for all α. Let $\breve{\alpha}_{\ell q}$ denote the global minimizer of $g_2(\alpha)$ in the interval $(0, \bar{\alpha}_{\ell k}]$. Now analogous to what we did in the proof of Theorem 3.3.1 we have $\breve{\alpha}_{\ell q}$ as the solution of the equation

$$-\sigma_{\ell q}^2 + \sigma_{\ell q}\left((v_{\min} - \alpha\sigma_{\ell q})^{-1-\log n} - v_{\min}^{-1-\log n}\right) = 0. \tag{5.12}$$

Using Lemma 10.1.1 one can derive that

$$\begin{aligned}\breve{\alpha}_{\ell k} &\geq \frac{v_{\min}\sigma_\ell v_{\min}^{1+\log n}}{\sigma_\ell(1+\log n)(1+\sigma_\ell v_{\min}^{1+\log n})} \geq \frac{v_{\min}\sigma_\ell v_{\min}^{1+\log n}}{\sigma_\ell(1+\log n)(1+\sigma_\ell v_{\min}^{1+\log n})} \\ &\geq \frac{\theta_2(n,\tau)v_{\min}}{4\sigma_\ell \log n \exp(\frac{-q+1+\log n}{2})} = \tilde{\alpha}_{\ell q},\end{aligned}$$

where the last inequality follows from Lemma 5.1.7. By using Lemma 10.1.2, for $\alpha = \tilde{\alpha}_{\ell q}$ we have

$$g(\alpha) \leq g_1(\alpha) \leq \frac{g^{'}(0)}{2}\alpha = -\frac{\sigma_{\ell q}^2}{2}\alpha,$$

which further implies

$$\Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*) \leq \Phi_\ell(x, s, \mu_\ell^*) - \frac{1}{2}\tilde{\alpha}_{\ell q}\sigma_{\ell q}^2.$$

From (5.12) and (5.9) we have

$$(v_{\min} - \alpha\sigma_{\ell q})^{-1-\log n} = v_{\min}^{-1-\log n} + \sigma_{\ell q} \leq \left(1 + \frac{\exp\left(\frac{-q+1+\log n}{2}\right)}{\theta_2(n,\tau)}\right)\sigma_{\ell q}.$$

Then we can write

$$\begin{aligned}\Phi_\ell(x(\alpha), s(\alpha), \mu_\ell^*) \ &\leq \Phi_\ell(x, s, \mu_\ell^*) - \frac{\theta_2^2(n,\tau)(v_{\min} - \alpha\sigma_{\ell q})^{-\log n}}{16\log n \exp(-q+1+\log n)} \\ &\leq \Phi_\ell(x, s, \mu_\ell^*) - \check{\alpha}_{\ell q} v(\alpha)_{\min}^{-\log n}, \end{aligned} \tag{5.13}$$

where $\check{\alpha}_{\ell q} = \dfrac{\theta_2^2(n,\tau)}{16\log n \exp(-q+1+\log n)}$ and the last inequality follows from the fact that

$$v_{\min} - \alpha\sigma_{\ell q} \leq v(\alpha)_{\min}.$$

This completes the proof of the theorem. □

5.3.2 The Predictor Step

In this subsection we discuss the behavior of the search direction with the different μ values that we choose in the predictor step. We use $\mu = 0$ (affine scaling) whenever the reduction of the duality gap is at least as good as theory guarantees for the $q-$degree SR step with $\mu = \mu_\ell^t$. If the reduction is not

satisfactory, we make an SR step with $\mu = \mu_\ell^t$. Thus the worst case that can happen is the $q-$degree SR step. This implies that for the worst case iteration complexity analysis it suffices to discus the behavior of the $q-$degree SR step. In what follows we compute the step size for the affine scaling step and the $q-$degree SR step with $\mu = \mu_\ell^t$.

Theorem 5.3.2 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (5.5), where $\mu = \mu_\ell^t$ is defined by equation (4.8) and $1 \leq q \leq 1 + \log n$. Then, for the step size $\hat{\alpha}_{\ell q} = \frac{\theta_2(n,\tau)}{4\exp(1)\tau^{\frac{1}{2+\log n}}\sigma_{\ell q}\log n \exp(\frac{-q+1+\log n}{2})}$, the relation*

$$\Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^t) \leq \Phi_\ell(x, s, \mu_\ell^t) - \frac{1}{2}\hat{\alpha}_{\ell q}\sigma_{\ell q}^2$$

holds.

Proof: The proof is analogous to the proof of Theorem 5.3.1 except that at the end of the proof we use the bound for $v_{\min}$ from Lemma 5.1.7. □

Theorem 5.3.3 *Let $(\Delta x, \Delta y, \Delta s)$ be the solution of system (5.5), where $\mu = \mu_\ell^t$ is defined by equation (4.8), $1 \leq q \leq 1 + \log n$ and $\frac{\mu_g}{\mu_\ell^h} \leq 0.98\tau$. Then for step size $\hat{\alpha}_{\ell q}$ the relation*

$$\Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_g(\hat{\alpha}_{\ell q})) \leq \eta_\ell(n, \tau) \tag{5.14}$$

holds.

Proof: We start by considering the function

$$g_1(\hat{\alpha}_{\ell q}) = \Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^*) = \frac{\mu_\ell^t}{\mu_\ell^*}\frac{\|v(\hat{\alpha}_{\ell q})\|^2}{2} - \frac{n}{2} + \left(\frac{\mu_\ell^*}{\mu_\ell^t}\right)^{\frac{\log n}{2}} \frac{\left\|v(\hat{\alpha}_{\ell q})^{\frac{-\log n}{2}}\right\|^2}{\log n} - \frac{n}{\log n}.$$

Note that, from the definition of μ_ℓ^* and μ_ℓ^t, we know that

$$\frac{\mu_\ell^t}{\mu_\ell^*}\|v(0)\|^2 = \frac{\mu_\ell^*}{\mu_\ell^t}\left\|v(0)^{\frac{-\log n}{2}}\right\|^2 = n(\tau_0)^{\frac{\log n}{2+\log n}},$$

where $\tau_0 := \dfrac{\mu_g}{\mu_\ell^h}$. On the other hand, we have

$$\|v(\hat{\alpha}_{\ell q})\|^2 = \|v\|^2 + \hat{\alpha}_{\ell q} v^T(d_x + d_s) \leq \|v\|^2,$$

because $v^T(d_x + d_s) = \left\|v^{\frac{1-q}{2}}\right\|^2 - \|v\|^2 \leq 0$. It is easy to verify that

$$\left\|v(\hat{\alpha}_{\ell q})^{\frac{-\log n}{2}}\right\|^2 \leq (1 - \hat{\alpha}_{\ell q} v_{\min}^{-1}\sigma_{\ell q})^{-\log n}\left\|v^{\frac{-\log n}{2}}\right\|^2.$$

By the definition of $\hat{\alpha}_{\ell q}$, we have $(1 - \hat{\alpha}_{\ell q} v_{\min}^{-1}\sigma_{\ell q})^{-\log n} \leq 1.008$ that gives

$$g_1(\hat{\alpha}_{\ell q}) \leq \frac{n(\tau_0)^{\frac{\log n}{2+\log n}}}{2} - \frac{n}{2} + \frac{1.008n(\tau_0)^{\frac{\log n}{2+\log n}}}{\log n} - \frac{n}{\log n}.$$

Since $\Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^*(\hat{\alpha}_{\ell q})) \leq \Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^*)$, then the next iterate is in the neighborhood $\mathcal{N}_\ell(\tau, n)$ if

$$\frac{(\tau_0)^{\frac{\log n}{2+\log n}}}{2} + \frac{1.008(\tau_0)^{\frac{\log n}{2+\log n}}}{\log n} \leq \frac{\tau^{\frac{\log n}{2+\log n}}}{2} + \frac{\tau^{\frac{\log n}{2+\log n}}}{\log n}.$$

This inequality definitely holds if $\tau_0 \leq 0.98\tau$ that completes the proof. □

Remark 5.3.1 *If $0.98\tau \leq \dfrac{\mu_g}{\mu_\ell^h} < \tau$, theoretically, the step size in the SR step may not be as large as $\hat{\alpha}_{\ell q}$. Therefore, in this case either it might do SR step or affine scaling step. Moreover, for polynomial worst case iteration complexity, as we already mentioned, it suffices to consider the SR step in the predictor step because affine scaling step is used only when the affine scaling step size is larger than the warranted SR step size.*

5.3.3 Iteration Complexity

By applying the results of Lemma 4.3.11 to Theorem 5.3.2, we can prove the following theorem.

Theorem 5.3.4 *Let $\tau \geq 8$ and $(\Delta x, \Delta y, \Delta s)$ be the solution of system (5.5) as defined in Algorithm 5.2, and let $\hat{\alpha}_{\ell q}$ be the default step size as defined in Theorem 5.3.2. Then*

$$\Phi_\ell \left(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), (1-\bar{\theta})\mu_\ell^t\right) \leq \Phi_\ell(x, s, \mu_\ell^t),$$

where

$$\bar{\theta} = \frac{\hat{\alpha}_{\ell q}\sigma_{\ell q}}{(\tau+3)}\sqrt{\frac{\tau^{\frac{\log n}{2+\log n}} - 2}{n}}.$$

Proof: From Lemma 4.3.11, it can be seen that to prove the theorem it suffices to choose θ satisfying the inequality

$$\Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^t) + 2n\theta \leq (1-\theta)\Phi_\ell(x, s, \mu_\ell^t).$$

Using Theorem 5.3.2 we conclude that the above inequality will certainly be satisfied if

$$\theta\Phi_\ell(x, s, \mu_\ell^t) + 2n\theta \leq \frac{1}{2}\hat{\alpha}_{\ell q}\sigma_{\ell q}^2.$$

By (5.10) it suffices to have

$$\theta\Phi_\ell(x, s, \mu_\ell^t) + 2n\theta \leq \frac{\hat{\alpha}_{\ell q}\sigma_{\ell q}}{2}\sqrt{\left(\tau^{\frac{\log n}{2+\log n}} - 2\right)n}$$

that implies

$$\theta\left(\frac{(\tau-1)n}{2} + 2n\right) \leq \frac{\hat{\alpha}_{\ell q}\sigma_{\ell q}}{2}\sqrt{\left(\tau^{\frac{\log n}{2+\log n}} - 2\right)n}.$$

This relation implies that if we choose

$$\theta = \frac{\hat{\alpha}_{\ell q}\sigma_{\ell q}}{\tau+3}\sqrt{\frac{\tau^{\frac{\log n}{2+\log n}}-2}{n}},$$

then

$$\Phi_\ell\left(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), (1-\theta)\mu_\ell^t\right) \leq \Phi_\ell(x, s, \mu_\ell^t)$$

that completes the proof. □

Now we can proceed to discuss the complexity of Algorithm 5.2 that follows from the reduction of the duality gap in the predictor step by the $q-$degree SR directions. By the choice of μ_ℓ^t we know that the proximity function $\Phi_\ell(x, s, \mu_\ell^t)$ remains invariant for all the iterates. Let us denote by $\mu_\ell^t(\hat{\alpha}_{\ell q})$ the target parameter value after one step. Then we have

$$\Phi_\ell(x, s, \mu_\ell^t) = \Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^t(\hat{\alpha}_{\ell q})).$$

On the other hand, form Theorem 5.3.4 we have

$$\Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), (1-\theta)\mu_\ell^t) \leq \Phi_\ell(x(\hat{\alpha}_{\ell q}), s(\hat{\alpha}_{\ell q}), \mu_\ell^t(\hat{\alpha}_{\ell q})).$$

Since the proximity function is a strictly convex function w.r.t. μ, we have

$$\mu_\ell^t(\hat{\alpha}_{\ell q}) \leq (1-\theta)\,\mu_\ell^t. \tag{5.15}$$

Now we are ready to give the complexity of Algorithm 5.2.

Remark 5.3.2 *When* $0.98\tau \leq \frac{\mu_g}{\mu_\ell^h} < \tau$, *algorithm might do a slightly shorter step and it implies a similar reduction of* μ_ℓ^t. *For the complexity analysis we do not use this reduction explicitly, however it is important to know that* μ*take this to account.*

Theorem 5.3.5 *After at most*

$$\left\lceil \frac{2}{\theta} \log \frac{\left(\tau + \frac{2}{\log n}\right) n}{\epsilon} \right\rceil .$$

iterations Algorithm 5.2 terminates with a solution for which $x^T s \leq \epsilon$.

Proof: Since we ensured that the worst case duality gap reduction is the one that the SR direction gives, we know that the duality gap reduction of each step is always at least as big as the SR step gives. Using inequality (5.15) for each iteration (predictor-corrector step), after at most

$$\left\lceil \frac{2}{\theta} \log \frac{\left(\tau + \frac{2}{\log n}\right) n}{\epsilon} \right\rceil$$

iterations we have $\mu_\ell^t \leq \dfrac{\epsilon}{\left(\tau + \frac{2}{\log n}\right) n}$, and then Lemma 4.1.1 implies $x^T s \leq \epsilon$. This completes the proof of the theorem. □

Corollary 5.3.1 *If* $q = 1 + \log n$, *then the number of iteration of Algorithm 5.2 is at most*

$$O\left(\sqrt{n} \log n \log \frac{\left(\tau + \frac{2}{\log n}\right) n}{\epsilon} \right).$$

Corollary 5.3.2 *If* $q = 1$, *then the number of iterations of Algorithm 5.2 is at most*

$$O\left(n \log n \log \frac{\left(\tau + \frac{2}{\log n}\right) n}{\epsilon} \right).$$

5.4 Superlinear Convergence

In this section we prove the quadratic convergence of Algorithm 5.2. In case of monotone LCPs, Ye and Anstreicher [68] have proved for predictor-corrector algorithms that for sufficiently small μ_g the relation

$$|(\Delta x)_i(\Delta s)_i| = \mathcal{O}({\mu_g}^2),\ i = 1, \cdots, n \tag{5.16}$$

holds. Since LO is a special case of monotone LCP, relation (5.16) is valid for LO as well. Since in the predictor step we have two choices for the target value, it suffices to discuss the case when the affine scaling direction is used as the predictor direction. The reason is that close to optimality the affine scaling direction is always dominant to the SR direction. Then we have

$$|(d_x)_i(d_s)_i| = \frac{(\Delta x)_i(\Delta s)_i}{\mu_\ell^*} = \mathcal{O}(\mu_g).$$

In the following lemma we give a lower bound for the step size of the affine scaling predictor step.

Lemma 5.4.1 *Let (x, s) be an iterate in Algorithm 5.2. If the present duality gap is sufficiently small, such that (5.16) holds, then the step size α in the predictor step satisfies $\alpha \geq 1 - O\left(\mu_g^r\right)$ for some $r \in (0, 1)$.*

Proof: First we give an estimate for the maximal feasible step size in a predictor step. Let us consider (x, s) be an iterate after a corrector step. The new iterate is strictly feasible if both $v + \alpha d_x$ and $v + \alpha d_s$ are strictly feasible. Then, the maximal feasible step size $\alpha_{\max}$ satisfies the following inequality

$$(v + \alpha d_x)(v + \alpha d_s) = v^2 - \alpha v^2 + \alpha^2 d_x d_s \geq 0, \quad \alpha \in [0, \alpha_{\max}],$$

or, equivalently

$$e - \alpha e + \alpha^2 v^{-2} d_x d_s \geq 0, \quad \forall \alpha \in [0, \alpha_{\max}].$$

Since $|d_x d_s| = \mathcal{O}(\mu_g)$, we can conclude that $\alpha_{\max} \geq 1 - \mathcal{O}(\mu_g)$.

Now we prove that the step size in the affine scaling step also satisfies $\alpha \geq 1 - \mathcal{O}\left(\mu_g^r\right)$ for some $r \in (0,1)$. Let us define $v_+(\alpha) = \sqrt{\dfrac{x(\alpha)s(\alpha)}{(1-\alpha)\mu_\ell^*}}$, then we have

$$\begin{aligned}
\Phi(x(\alpha), s(\alpha),\ (1-\alpha)\mu_\ell^*) - \Phi(x,s,\mu_\ell^*) &= \frac{\left\| v_+(\alpha)^{\frac{-\log n}{2}} \right\|^2}{\log n} - \frac{\left\| v^{\frac{-\log n}{2}} \right\|^2}{\log n} \\
&= \frac{1}{\log n} \sum_{i=1}^{n} v_i^{-\log n} \left(\left(1 + \frac{\alpha^2}{1-\alpha} v_i^{-2} (d_x)_i (d_s)_i \right)^{\frac{-\log n}{2}} - 1 \right) \\
&\leq \frac{v_{\min}^{-\log n}}{\log n} \left(\left(1 + \frac{\alpha^2}{1-\alpha} \sum_{(d_x)_i (d_s)_i \leq 0} v_i^{-2} (d_x)_i (d_s)_i \right)^{\frac{-\log n}{2}} - 1 \right) \\
&\leq \frac{v_{\min}^{-\log n}}{\log n} \left(\left(1 - \frac{\alpha^2}{1-\alpha} \mathcal{O}(\mu_g) \right)^{\frac{-\log n}{2}} - 1 \right).
\end{aligned}$$

This inequality shows that for sufficiently small μ_g we have

$$\Phi(x(\alpha), s(\alpha), (1-\alpha)\mu_\ell^*) - \Phi(x,s,\mu_\ell^*) \leq \check{\alpha}_{\ell q} v_{\min}^{-\log n},$$

where α has to satisfy the following inequality

$$\left(1 - \frac{\alpha^2}{1-\alpha} \mathcal{O}(\mu_g) \right)^{\frac{-\log n}{2}} - 1 \leq \frac{\theta_2^2(n,\tau)}{16 \exp(-q+1+\log n)}.$$

That is equivalent to find an α such that

$$\begin{aligned}
O(\mu_g)\alpha^2 \ &+ \left(1 - \left(1 + \frac{\theta_2^2(n,\tau)}{16\exp(-q+1+\log n)} \right)^{\frac{-2}{\log n}} \right) \alpha \\
&-1 + \left(1 + \frac{\theta_2^2(n,\tau)}{16\exp(-q+1+\log n)} \right)^{\frac{-2}{\log n}} \leq 0.
\end{aligned}$$

This inequality definitely holds for $\alpha \geq 1 - O(\mu_g^r)$ for some $r \in (0,1)$, which completes the proof of Lemma. □

Our superlinear convergence result is the following.

Theorem 5.4.1 *Let (x^k, y^k, s^k) be the iterate generated by Algorithm 5.2. When μ_g is sufficiently small, the algorithm is superlinearly convergent in the sense that $\mu_g^{k+1} = \mathcal{O}\left((\mu_g^k)^(1+r)\right)$ for some $r \in (0,1)$ and any accumulation point of the sequence (x^k, s^k) is a strictly complementary optimal solution of the problem.*

Proof: Using Lemma 5.4.1 we have

$$\mu_g(\alpha) = (1-\alpha)\mu_g \leq \mathcal{O}\left((\mu_g^k)^(1+r)\right),$$

where $\mu_g(\alpha) = \dfrac{x(\alpha)^T s(\alpha)}{n}$. The proof of the convergence to an accumulation point and the proof of their properties is analogous to the proof of Theorem 5.14 in [63] and it is omitted here. □

In summary, in this chapter a new variant of the predictor-corrector algorithm using the idea of self-regularity is proposed. Having multiple choices in the predictor step and using various SR search directions in the corrector step are the significant differences of this variant compared to the classical predictor-corrector algorithms. The polynomial complexity of the algorithm with its superlinear convergence is proved. In certain cases we improved the best so far iteration complexity for large-update large neighborhood predictor-corrector algorithms. In Chapter 8 we will report some limited computational results based on this algorithm.

Chapter 6

On Mehrotra-type Predictor-Corrector IPMs

In this chapter we analyze a feasible version of a more practical algorithm originally proposed by Mehrtora[1] [34] that has been widely used by most IPMs based software packages [21, 70, 73] in their implementations. He also extended this algorithm to a second order variant of it by using the power series extension of Monteiro et al. [35, 37]. Zhang and Zhang [71] proved the polynomiality of the second order variant of Mehrotra's algorithm without considering the adaptivity of the barrier (centering) parameter, which is a significant element of the algorithm. Here, first we describe the feasible version of the original algorithm of Mehrotra, then by a numerical example we show that this algorithm might take very small steps in order to keep the iterates in a certain neighborhood of the central path. This makes it hard to argue about

[1]Mehrotra's original algorithm is an infeasible algorithm. However, the self-dual embedding model (see Section 10.2 in the Appendix) can be used to construct slightly bigger LO problem which has an obvious starting point on the central path

possible polynomial iteration complexity for a practically efficient algorithm. This observation motivated us to combine this algorithm with a safeguard that gives us a lower bound for the maximum step step size at each corrector iteration and consequently polynomial complexity. It is worth mentioning that our approach is different from more recent variations of this algorithm that are using the multiple centrality and multiple corrections to avoid possible bad behavior of the algorithms [19, 24, 36]. The results we present in this chapter mostly appeared in [53].

6.1 Mehrotra-type Algorithm

Let us first describe a feasible variant of the original algorithm of Mehrotra. This algorithm has two steps, a predictor step and a corrector step. In the predictor step, like all the other predictor-corrector algorithms, it solves the so called affine scaling system of equations:

$$\begin{aligned} A\Delta x^a &= 0, \\ A^T\Delta y^a + \Delta s^a &= 0, \\ s\Delta x^a + x\Delta s^a &= -xs, \end{aligned} \tag{6.1}$$

then it computes the maximum feasible step size that ensures

$$(x + \alpha_{\mathrm{a}}\Delta x^a, s + \alpha_{\mathrm{a}}\Delta s^a) \geq 0.$$

However, the algorithm does not take such a step right away. It uses the information from the predictor step to compute the corrector direction as follows (the predictor step is included here).

$$A\Delta x = 0,$$

$$\begin{aligned} A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \Delta x^a \Delta s^a, \end{aligned} \tag{6.2}$$

where μ, the centering (barrier) parameter, is defined adaptively by

$$\mu = \left(\frac{g_a}{g}\right)^2 \frac{g_a}{n},$$

where $g_a = (x + \alpha_{\rm a}\Delta x^a)^T(s + \alpha_{\rm a}\Delta s^a)$ and $g = x^T s$. Since $(\Delta x^a)^T \Delta s^a = 0$, the previous relation implies

$$\mu = (1 - \alpha_{\rm a})^3 \mu_g. \tag{6.3}$$

Finally, Mehrotra's algorithm makes a step in the $(\Delta x, \Delta y, \Delta s)$ direction by an appropriate step size[2], see Figure 6.1. As we see, in both the predictor and corrector steps, this algorithm uses one coefficient matrix, while this was not the case in the algorithms of the previous chapters. This is significant numerically, because one does not need to do an extra factorization in the corrector step, which is costly and might create numerical instability. The excellent practical performance of this algorithm motivated us to analyze its worst case iteration complexity.

6.2 Motivation

In this section we present a numerical example that shows using the strategy described in the introduction might force the algorithm to make very small steps to keep the iterate in a certain neighborhood of the central path, which

[2]Mehrotra's original algorithm uses different step sizes in both the primal and the dual spaces. However, here for simplicity of the analysis we assume identical step size for primal and dual problems.

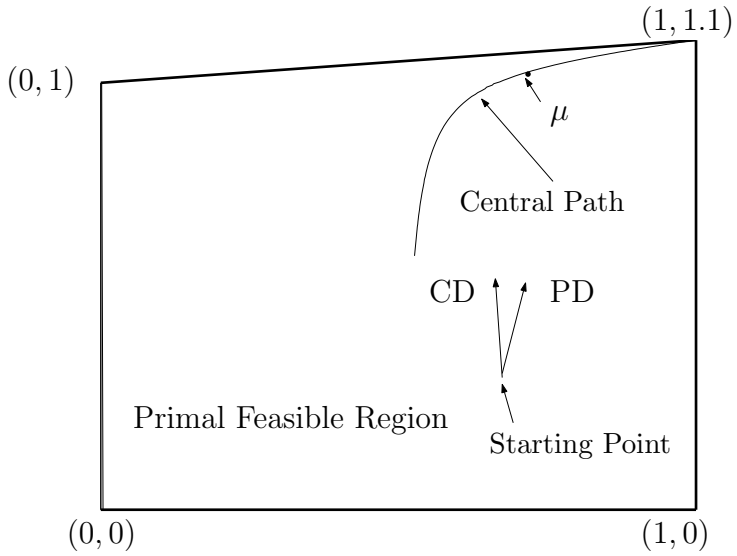

Figure 6.1: One Iteration of Mehrotra's algorithm (PD: Predictor Direction and CD: Corrector Direction)

further implies the algorithm needs to take many iterations to converge. The example indicates that Mehrotra's adaptive updating scheme of the centering parameter and adding correction terms to the corrector system of equations has to be combined with certain safeguards to get a warranted step size at each iteration. Before going to the details of the example let us introduce the neighborhood that we use throughout this chapter.

Most efficient IPM solvers work in the negative infinity norm neighborhood defined by

$$\mathcal{N}_\infty^-(\gamma) := \{(x, y, s) \in \mathcal{F}^0 : \ x_i s_i \geq \gamma \mu_g \ \forall i = 1, \cdots, n\}, \tag{6.4}$$

where $\gamma \in (0, 1)$ is a constant independent of n. In this chapter, we consider algorithms that work in $\mathcal{N}_\infty^-(\gamma)$ (called large neighborhood).

Let us consider the following simple LO:

$$\begin{aligned} \min \quad & -x_2 \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1 + \delta x_1. \end{aligned} \tag{6.5}$$

Let the algorithm start with from the following feasible points in the neighborhood $\mathcal{N}_\infty^-(0.1)$:

$$x^0 = (0.03; 0.9; 0.97; 0.103),\ s^0 = (6.8; 1; 7; 2),\ y^0 = (-7, -2),$$

where $\delta = 0.1$. For the given starting point, by using identical step sizes for both primal and dual problems, in the third iteration the maximum step size in the predictor is $\alpha_{\text{a}} = 0.96$ while the maximum step size in the corrector step is $\mathcal{O}(10^{-4})$ and this value is getting worse for later iterations. To explain what we observed, we examine the constraints

$$x_i(\alpha)s_i(\alpha) \geq \gamma\mu_g(\alpha)\ \forall i \in \mathcal{I} \tag{6.6}$$

for $\gamma = 0.1$ that keep the next iterate in $\mathcal{N}_\infty^-(0.1)$ neighborhood.

By expanding inequality (6.6) and reordering one has

$$(1-\alpha)x_i s_i + \alpha(1-0.1)\mu - \alpha\Delta x_i^a \Delta s_i^a + \alpha^2 \Delta x_i \Delta s_i \geq 0.1(1-\alpha)\mu_g\ \forall i \in \mathcal{I}.$$

For the given staring point, $x_1 s_1 - 0.1\mu_g$ is a very small nonnegative number, while $\Delta x_1^a \Delta s_1^a$ and $\Delta x_1 \Delta s_1$ are both negative numbers whose absolute values are dominated by $x_1 s_1$, and finally $\mu = \mathcal{O}(10^{-5})$ due to a big affine scaling step size. Incorporating all this information into (6.6) implies that the algorithm requires a very small step in order to satisfy (6.6). This phenomenon might be the result of following:

- An aggressive update of centering parameter μ using (6.3).

- The usage of the correction terms in the corrector system of equations.

To alleviate this phenomenon, we propose the following remedies:

- Using a fixed fraction of μ_g, for example $\mu = \frac{\mu_g}{10}$ rather than an adaptive update.

- Cutting the maximum step size in the predictor step, if it is greater than a certain threshold. This might prevent an aggressive update.

- Removing the correction terms in the corrector system of equations.

For this specific example, these ideas help us to solve the difficulty that might arise. However, in general, removing the correction terms is not as effective as using a simple large update of the centering parameter. It is worthwhile to mention that by using different step sizes in the predictor and corrector steps while using a slightly different starting point one still may observe this phenomenon even when removing the correction terms in case of small steps.

These observations motivate us to introduce a safeguard strategy that will help us have control over the minimal warranted step size both theoretically and practically. In our safeguard we simply use a fixed fraction of μ_g as the μ value. In the case that the affine scaling step size is very small, which implies marginal reduction of the duality gap, we employ the same large update safeguard technique.

6.3 A Safeguard Based Algorithm

In this section, following the discussion in the previous section, first we outline our new safeguard-based algorithm. Then we establish its worth case iteration complexity.

Our new safeguard based algorithm can be outlined as follows:

Algorithm 6.3: Safeguard Based Algorithm

Input:

A proximity parameters $\gamma \in (0, \frac{1}{3})$; a safeguard parameter

$\beta \in [\gamma, \frac{1}{3})$;

an accuracy parameter $\epsilon > 0$; $(x^0, y^0, s^0) \in \mathcal{N}_\infty^-(\gamma)$.

begin

while $x^T s \geq \epsilon$ **do**

begin

Predictor Step

Solve (6.1) and compute the maximum step size $\alpha_{\rm a}$ such that $(x(\alpha_{\rm a}), y(\alpha_{\rm a}), s(\alpha_{\rm a})) \in \mathcal{F}$;

end

begin

Corrector step

If $\alpha_{\rm a} \geq 0.1$, **then** solve (6.2) with $\mu = (1 - \alpha_{\rm a})^3 \mu_g$ and compute the maximum step size $\alpha_{\rm c}$ such that $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) \in \mathcal{N}_\infty^-(\gamma)$;

If $\alpha_{\rm c} < \frac{3\gamma^2}{2n^2}$, then solve (6.2) with $\mu = \frac{\beta}{1-\beta}\mu_g$ and compute the maximum step size $\alpha_{\rm c}$ such that $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) \in \mathcal{N}_\infty^-(\gamma)$;

end

else

Solve (6.2) with $\mu = \frac{\beta}{1-\beta}\mu_g$

and compute the maximum step size $\alpha_{\rm c}$ such that $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) \in \mathcal{N}_\infty^-(\gamma)$;

end

Set $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) = (x + \alpha_{\rm c}\Delta x, y + \alpha_{\rm c}\Delta y, s + \alpha_{\rm c}\Delta s)$.

end

end

Remark 6.3.1 *In the worst case using the identical step sizes for both the primal and dual problems, comparing with the Mehrotra's algorithm, our new algorithm requires an extra backsolve to make a better step.*

6.3.1 Estimation of Step Size

Before going further let us present the following two technical results that will be recalled frequently in this and the next chapters.

Lemma 6.3.1 *Let* $(\Delta x^a, \Delta y^a, \Delta s^a)$ *be the solution of (6.1), then*

$$\Delta x_i^a \Delta s_i^a \leq \frac{x_i s_i}{4}, \ \forall i \in \mathcal{I}_+,$$

where

$$\mathcal{I}_+ = \{i \mid \Delta x_i^a \Delta s_i^a > 0\}.$$

Proof: By the third equation of (6.1) for $i \in \mathcal{I}_+$ we have

$$s_i \Delta x_i^a + x_i \Delta s_i^a = -x_i s_i.$$

If we divide this equation by $x_i s_i$ we get

$$\frac{\Delta x_i^a}{x_i} + \frac{\Delta s_i^a}{s_i} = -1.$$

Since $\Delta x_i^a \Delta s_i^a > 0$, this equality implies that both $\Delta x_i^a < 0$ and $\Delta s_i^a < 0$. Then, from

$$0 \leq \left(\frac{\Delta x_i^a}{x_i} - \frac{\Delta s_i^a}{s_i}\right)^2 = \left(\frac{\Delta x_i^a}{x_i}\right)^2 + \left(\frac{\Delta s_i^a}{s_i}\right)^2 - 2\frac{\Delta x_i^a \Delta s_i^a}{x_i s_i} = 1 - 4\frac{\Delta x_i^a \Delta s_i^a}{x_i s_i}$$

we get

$$\Delta x_i^a \Delta s_i^a \leq \frac{x_i s_i}{4}.$$

□

Lemma 6.3.2 *Let $(\Delta x^a, \Delta y^a, \Delta s^a)$ be the solution of (6.1), then we have*

$$\sum_{i\in\mathcal{I}_+} \Delta x_i^a \Delta s_i^a = \sum_{i\in\mathcal{I}_-} |\Delta x_i^a \Delta s_i^a| \leq \frac{1}{4}\sum_{i\in\mathcal{I}_+} x_i s_i \leq \frac{x^T s}{4}.$$

Proof: Since $(\Delta x^a)^T \Delta s^a = 0$, the proof is a direct consequence of the previous lemma. □

The following theorem gives a lower bound for the maximum feasible step size in the worst case for the predictor step. This shows that there always exists a positive step size for the predictor step.

Theorem 6.3.1 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ and $(\Delta x^a, \Delta y^a, \Delta s^a)$ is the solution of (6.1). Then the maximum feasible step size, $\alpha_{\rm a}$, such that*
$(x(\alpha_{\rm a}), s(\alpha_{\rm a}))$
≥ 0, *satisfies*

$$\alpha_{\rm a} \geq \frac{2\sqrt{\gamma^2 + \gamma n} - 2\gamma}{n}. \tag{6.7}$$

Proof: For $i \in \mathcal{I}_+$ any step size in $(0, 1]$ is feasible. Since $(\Delta x^a)^T \Delta s^a = 0$, there is at least one $i \in \mathcal{I}_-$. Since $(x, s) \in \mathcal{N}_\infty^-(\gamma)$, then by Lemma 6.3.2 we have

$$x_i(\alpha)s_i(\alpha) = (1-\alpha)x_i s_i + \alpha^2 \Delta x^a{}_i \Delta s^a{}_i \geq \gamma\left(1 - \alpha - \frac{n\alpha^2}{4\gamma}\right)\mu_g. \tag{6.8}$$

Our aim is to ensure that $x_i(\alpha)s_i(\alpha) \geq 0$. For this, it suffices to require that

$$\gamma\left(1 - \alpha - \frac{n\alpha^2}{4\gamma}\right)\mu_g \geq 0$$

that is equivalent to

$$n\alpha^2 + 4\gamma\alpha - 4\gamma \leq 0.$$

This inequality holds when $\alpha \in [\frac{-2\sqrt{\gamma^2+\gamma n}-2\gamma}{n}, \frac{2\sqrt{\gamma^2+\gamma n}-2\gamma}{n}]$ that completes the proof of the lemma. □

We use the maximum step size $\alpha_{\rm a}$ computed in the predictor step, to define the target value μ in the corrector step. The following technical lemma will be used in the next theorem, which estimates the maximum step size in the corrector step. This is a special safeguard step in a large update strategy in which the next iterate stays in the $\mathcal{N}_\infty^-(\gamma)$ neighborhood.

Lemma 6.3.3 *Suppose that the current iterate is* $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$ *and let* $(\Delta x, \Delta y,$
$\Delta s)$ *be the solution of (6.2), where* $\mu \geq 0$. *Then we have*

$$\|\Delta x \Delta s\| \leq 2^{\frac{-3}{2}} \left(\frac{1}{\gamma} \left(\frac{\mu}{\mu_g} \right)^2 - \left(2 - \frac{1}{2\gamma} \right) \frac{\mu}{\mu_g} + \frac{17\gamma + n}{16\gamma} \right) n\mu_g.$$

Proof: If we multiply the third equation of (6.2) by $(XS)^{-\frac{1}{2}}$, then by Lemma 5.3 of [63] we have

$$\begin{aligned}
\|\Delta x \Delta s\| &\leq 2^{\frac{-3}{2}} \left\| \mu (XSe)^{-\frac{1}{2}} - (XSe)^{\frac{1}{2}} - (XS)^{-\frac{1}{2}} \Delta x^a \Delta s^a \right\|^2 \\
&= 2^{\frac{-3}{2}} \left(\sum_{i \in \mathcal{I}} \frac{\mu^2}{x_i s_i} + \sum_{i \in \mathcal{I}} x_i s_i + \sum_{i \in \mathcal{I}} \frac{(\Delta x_i^a \Delta s_i^a)^2}{x_i s_i} - 2n\mu - 2\mu \sum_{i \in \mathcal{I}} \frac{\Delta x_i^a \Delta s_i^a}{x_i s_i} \right) \\
&\leq 2^{\frac{-3}{2}} \left(\frac{n\mu^2}{\gamma \mu_g} + n\mu_g + \frac{n\mu_g}{16} + \frac{n^2 \mu_g}{16\gamma} - 2n\mu + \frac{n\mu}{2\gamma} \right),
\end{aligned}$$

where the last inequality follows from Lemma 6.3.2 and the assumption that the previous iterate is in $\mathcal{N}_\infty^-(\gamma)$. By reordering and factorizing we conclude the proof of the lemma. □

Motivated from the computational practice, since the neighborhood parameter γ might be very small, therefore we use $\mu = \frac{\beta}{1-\beta}\mu_g$ as the value of

safeguard, where $\gamma \leq \beta < \frac{1}{3}$. Since taking $\beta = \gamma$ when γ is very small might imply an aggressive update. This choice avoids having such an update. The following corollary gives an explicit upper bound for this specific value of μ.

Corollary 6.3.1 *If* $\mu = \frac{\beta}{1-\beta}\mu_g$, *where* $\gamma \leq \beta < \frac{1}{3}$ *and* $\gamma \in (0, \frac{1}{3})$, *then*

$$\|\Delta x \Delta s\| \leq \frac{1}{2\gamma} n^2 \mu_g.$$

In the following theorem we give a lower bound for the maximum step size in the corrector step for this specific choice of the centering parameter μ.

Theorem 6.3.2 *Suppose that the current iterate* $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$, *where* $\gamma \in (0, \frac{1}{3})$ *and* $(\Delta x, \Delta y, \Delta s)$ *is the solution of (6.2) with*

$$\mu = \frac{\beta}{1-\beta}\mu_g,$$

where $\gamma \leq \beta < \frac{1}{3}$. *Then the maximum step size in the corrector step* α_c *for which* $(x(\alpha_c), y(\alpha_c), s(\alpha_c))$ *in* $\mathcal{N}_\infty^-(\gamma)$, *satisfies*

$$\alpha_c \geq \frac{3\gamma^2}{2n^2}.$$

Proof: The goal is to find the maximum nonnegative α for which (6.6) holds. To do so, first let us define

$$t = \max_{i \in \mathcal{I}_+} \left\{ \frac{\Delta x_i^a \Delta s_i^a}{x_i s_i} \right\}. \tag{6.9}$$

Since $(\Delta x^a)^T \Delta s^a = 0$, then $\mathcal{I}_+ \neq \emptyset$. The worst case for inequalities in (6.6) may happen when $\Delta x_i^a \Delta s_i^a > 0$. Therefore we have

$$\begin{aligned} x_i(\alpha)s_i(\alpha) &= x_i s_i + \alpha(\mu - x_i s_i - \Delta x^a{}_i \Delta s^a{}_i) + \alpha^2 \Delta x_i \Delta s_i \\ &\geq (1-\alpha)x_i s_i + \alpha\mu - \alpha t x_i s_i - \frac{\alpha^2 n^2 \mu_g}{2\gamma} \end{aligned}$$

$$= (1-(1+t)\alpha)x_i s_i + \alpha\mu - \frac{\alpha^2 n^2 \mu_g}{2\gamma},$$
$$\geq (1-(1+t)\alpha)\gamma\mu_g + \alpha\mu - \frac{\alpha^2 n^2 \mu_g}{2\gamma},$$

where the first inequality follows from corollary 6.3.1, $\alpha \geq 0$, the definition of t given by (6.9) and the last inequality holds for $0 \leq \alpha \leq \frac{4}{5}$ since $\frac{1}{1+t} \geq \frac{4}{5}$. We also have

$$\mu_g(\alpha) = \left(1 - \alpha + \alpha\frac{\mu}{\mu_g}\right)\mu_g. \tag{6.10}$$

Therefore, in order to stay in $\mathcal{N}_\infty^-(\gamma)$ one has to have

$$(1-(1+t)\alpha)\gamma\mu_g + \alpha\mu - \frac{\alpha^2 n^2 \mu_g}{2\gamma} \geq \gamma\left(1 - \alpha + \alpha\frac{\mu}{\mu_g}\right)\mu_g,$$

which is equivalent to

$$(1-\gamma)\mu - \gamma t\mu_g \geq \frac{\alpha n^2 \mu_g}{2\gamma}. \tag{6.11}$$

Using Lemma 6.3.1 and the definition of μ one has

$$(1-\gamma)\mu - \gamma t\mu_g \geq \frac{(1-\gamma)\beta}{(1-\beta)}\mu_g - \frac{\gamma\mu_g}{4} \geq \frac{3\gamma\mu_g}{4}.$$

Therefore, inequality (6.11) holds if

$$\frac{3\gamma\mu_g}{4} \geq \frac{\alpha n^2 \mu_g}{2\gamma},$$

which implies

$$\alpha \leq \frac{3\gamma^2}{2n^2}.$$

Finally, we can conclude that

$$\alpha_c \geq \min\left(\frac{4}{5}, \frac{3\gamma^2}{2n^2}\right) = \frac{3\gamma^2}{2n^2},$$

which completes the proof. □

6.3.2 Iteration Complexity

The following theorem gives an upper bound for the maximum number of iterations in which Algorithm 6.3 stops with an $\epsilon-$approximate solution.

Theorem 6.3.3 *Algorithm 6.3 stops after at most*

$$O\left(n^2 \log \frac{(x^0)^T s^0}{\epsilon}\right)$$

iterations with a solution for which $x^T s \leq \epsilon$.

Proof: If $\alpha_{\rm a} < 0.1$ or $\alpha_{\rm c} < \frac{3\gamma^2}{2n^2}$, then the algorithm uses the safeguard strategy, and by (6.10) and Theorem 6.3.2 one has

$$\mu_g(\alpha) \leq \left(1 - \frac{3\gamma^2(1-2\beta)}{2(1-\beta)n^2}\right)\mu_g.$$

If $\alpha_{\rm a} \geq 0.1$ and $\alpha_{\rm c} \geq \frac{3\gamma^2}{2n^2}$, then the algorithm uses the Mehrotra's updating strategy, and thus one has

$$\mu_g(\alpha) \leq \left(1 - \frac{\gamma^2}{3n^2}\right)\mu_g.$$

That completes the proof by Theorem 3.2 of [63]. □

6.4 A Second Modification of the Algorithm

In this section we propose a slightly modified version of Algorithm 6.3 that reduces the iteration complexity significantly. The motivation for this modification is based on the following lemma that enables us to strengthen the bound in Lemma 6.3.3 and consequently results in an improved iteration complexity of the algorithm.

Lemma 6.4.1 *For $i \in \mathcal{I}_-$ and for any $0 < \alpha_{\rm a} \leq 1$ one has*

$$-\Delta x_i^a \Delta s_i^a \leq \frac{1}{\alpha_{\rm a}} \left(\frac{1}{\alpha_{\rm a}} - 1 \right) x_i s_i. \tag{6.12}$$

Proof: For the maximum step size in the predictor step, $\alpha_{\rm a}$, one has

$$x_i(\alpha_{\rm a}) s_i(\alpha_{\rm a}) \geq 0, \ i = 1, \cdots, n.$$

This is equivalent to

$$(1 - \alpha_{\rm a}) x_i s_i + \alpha_{\rm a}^2 \Delta x_i^a \Delta s_i^a \geq 0, \ i = 1, \cdots, n.$$

This completes the proof. □

We modify the corrector step by introducing the factor of $\alpha_{\rm a}$ in the right hand side of the third equation of the corrector step. The new system becomes

$$\begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \alpha_{\rm a} \Delta x^a \Delta s^a, \end{aligned} \tag{6.13}$$

where the centering parameter μ is defined as in the previous section. In other words, in the new algorithm one solves system (6.13) instead of system (6.2) in the corrector step. This slight modification of Algorithm 6.3 is based on the following observation. If the affine scaling search direction is a good choice, which typically implies that $\alpha_{\rm a}$ is large, then the classical corrector direction should also be a good direction. If the affine scaling is not that good, then we should be more careful in the corrector step.

Analogous to Lemma 6.3.3, we have the following bound for $\|\Delta x \Delta s\|$ which is much stronger than the one given in Lemma 6.3.3.

Lemma 6.4.2 *Suppose that the current iterate* $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$, $0 < \alpha_{\rm a} \leq 1$ *and let* $(\Delta x, \Delta y, \Delta s)$ *be the solution of (6.13). Then we have*

$$\|\Delta x \Delta s\| \leq 2^{\frac{-3}{2}} \left(\frac{1}{\gamma} \left(\frac{\mu}{\mu_g} \right)^2 - \left(2 - \frac{\alpha_{\rm a}}{2\gamma} \right) \frac{\mu}{\mu_g} + \frac{20 - 4\alpha_{\rm a} + \alpha_{\rm a}^2}{16} \right) n\mu_g.$$

Proof: Since $(\Delta x^a)^T \Delta s^a = 0$, both $\mathcal{I}_+$ and $\mathcal{I}_-$ are nonempty. If we multiply the third equation of (6.13) by $(XS)^{-\frac{1}{2}}$, then by Lemma 5.3 of [63] we have

$$\begin{aligned}
\|\Delta x \Delta s\| &\leq 2^{\frac{-3}{2}} \left\| \mu (XSe)^{-\frac{1}{2}} - (XSe)^{\frac{1}{2}} - \alpha_{\rm a} (XS)^{-\frac{1}{2}} \Delta x^a \Delta s^a \right\|^2 \\
&= 2^{\frac{-3}{2}} \left(\sum_{i \in \mathcal{I}} \frac{\mu^2}{x_i s_i} + x^T s + \alpha_{\rm a}^2 \sum_{i \in \mathcal{I}} \frac{(\Delta x_i^a \Delta s_i^a)^2}{x_i s_i} - 2n\mu - 2\alpha_{\rm a} \mu \sum_{i \in \mathcal{I}} \frac{\Delta x_i^a \Delta s_i^a}{x_i s_i} \right) \\
&\leq 2^{\frac{-3}{2}} \left(\frac{n\mu^2}{\gamma \mu_g} + n\mu_g + \frac{\alpha_{\rm a}^2 n \mu_g}{16} - (1 - \alpha_{\rm a}) \sum_{i \in \mathcal{I}_-} \Delta x_i^a \Delta s_i^a - 2n\mu + \frac{\alpha_{\rm a} n \mu}{2\gamma} \right) \\
&\leq 2^{\frac{-3}{2}} \left(\frac{1}{\gamma} \left(\frac{\mu}{\mu_g} \right)^2 + 1 + \frac{\alpha_{\rm a}^2}{16} + \frac{(1 - \alpha_{\rm a})}{4} - 2\frac{\mu}{\mu_g} + \frac{\alpha_{\rm a}}{2\gamma} \frac{\mu}{\mu_g} \right) n\mu_g \\
&= 2^{\frac{-3}{2}} \left(\frac{1}{\gamma} \left(\frac{\mu}{\mu_g} \right)^2 - \left(2 - \frac{\alpha_{\rm a}}{2\gamma} \right) \frac{\mu}{\mu_g} + \frac{20 - 4\alpha_{\rm a} + \alpha_{\rm a}^2}{16} \right) n\mu_g,
\end{aligned}$$

where the first inequality follows from (6.12), Lemma 6.3.2, and the assumption that the previous iterate is in $\mathcal{N}_\infty^-(\gamma)$. The second inequality also follows from Lemma 6.3.2. This completes the proof of the lemma. □

The following corollary gives an explicit upper bound for a specific μ.

Corollary 6.4.1 *Let* $\mu = \frac{\beta}{1-\beta} \mu_g$, *where* $\gamma \in (0, \frac{1}{3})$ *and* $\beta \in [\gamma, \frac{1}{3})$, *then*

$$\|\Delta x \Delta s\| \leq \frac{\beta}{\gamma(1 - \beta)} n\mu_g.$$

In the following theorem we estimate the maximum step size in the corrector step of the modified algorithm, where the corrector step is defined by (6.13).

Theorem 6.4.1 *Suppose that the current iterate $(x, y, s) \in \mathcal{N}_\infty^-(\gamma)$, where $\gamma \in (0, \frac{1}{3})$, $\beta \in [\gamma, \frac{1}{3})$, and $(\Delta x, \Delta y, \Delta s)$ is the solution of (6.13) with $\mu = \dfrac{\beta}{1-\beta}\mu_g$. Then the maximum step size α_c, such that $(x(\alpha_c), y(\alpha_c), s(\alpha_c)) \in \mathcal{N}_\infty^-(\gamma)$, satisfies*

$$\alpha_c \geq \frac{3\gamma(1-\gamma)}{4n}.$$

Proof: We need to estimate the maximum nonnegative α for which (6.6) holds. Since $(\Delta x^a)^T \Delta s^a = 0$, then $\mathcal{I}_+ \neq \emptyset$. The worst case for (6.6) may happen when $\Delta x_i^a \Delta s_i^a > 0$. Therefore, one has

$$\begin{aligned} x_i(\alpha)s_i(\alpha) &= x_i s_i + \alpha(\mu - x_i s_i - \alpha_a \Delta x^a{}_i \Delta s^a{}_i) + \alpha^2 \Delta x_i \Delta s_i \\ &\geq (1-\alpha)x_i s_i + \alpha\mu - \alpha\alpha_a t x_i s_i - \frac{\beta}{\gamma(1-\beta)}\alpha^2 n\mu_g \\ &= (1-\alpha(1+\alpha_a t))\, x_i s_i + \alpha\mu - \frac{\beta}{\gamma(1-\beta)}\alpha^2 n\mu_g \\ &\geq (1-\alpha_a(1+t))\,\gamma\mu_g + \alpha\mu - \frac{\beta}{\gamma(1-\beta)}\alpha^2 n\mu_g \qquad (6.14) \end{aligned}$$

where the first inequality follows from Corollary 6.4.2, $\alpha \geq 0$, (6.9) and the second inequality is true when $0 \leq \alpha \leq \frac{4}{5}$ since $\frac{1}{1+\alpha_a t} \geq \frac{4}{5}$. Hence by using (6.10) the new iterate is in $\mathcal{N}_\infty^-(\gamma)$ when

$$\gamma\left(1-\alpha(1+\alpha_a t)\right) + \frac{\mu}{\mu_g}\alpha - \frac{\beta}{\gamma(1-\beta)}\alpha^2 n \geq \gamma\left(1-\alpha+\frac{\mu}{\mu_g}\alpha\right).$$

One can easily see that this inequality holds for

$$\alpha \leq \frac{3\gamma(1-\gamma)}{4n}.$$

Finally we can conclude that

$$\alpha_c \geq \min\left(\frac{3\gamma(1-\gamma)}{4n}, \frac{4}{5}\right) = \frac{3\gamma(1-\gamma)}{4n},$$

which completes the proof. □

6.4.1 Iteration Complexity

Based on the previous results, the following theorem gives the worst case iterations complexity of modified version of Algorithm 6.3, where the corrector step is calculated by (6.13).

Theorem 6.4.2 *The modified version of Algorithm 6.3 stops after at most*

$$\mathcal{O}\left(n \log \frac{n}{\epsilon}\right)$$

iterations with a solution for which $x^T s \leq \epsilon$.

Proof: If $\alpha_{\rm a} < 0.1$ or $\alpha_{\rm c} < \frac{3\gamma(1-\gamma)}{4n}$, then the algorithm uses the safeguard strategy, and by (6.10) and Theorem 6.4.1 one has

$$\mu_g(\alpha) \leq \left(1 - \frac{3\gamma(1-\gamma)(1-2\beta)}{4(1-\beta)n}\right)\mu_g.$$

If $\alpha_{\rm a} \geq 0.1$ and $\alpha_{\rm c} \geq \frac{3\gamma(1-\gamma)}{4n}$, then the algorithm uses the Mehrotra's updating strategy, and thus one has

$$\mu_g(\alpha) \leq \left(1 - \frac{\gamma(1-\gamma)}{5n}\right)\mu_g,$$

which completes the proof by Theorem 3.2 of [63]. □

6.5 Superlinear Convergence

In this section we analyze the asymptotic behavior of the previous algorithms using a slight modification of the centering parameter μ, rather than using (6.3) due to the following observation: We note that [63] (Chapter 7) for sufficiently small μ_g the relation

$$|\Delta x_i^a \Delta s_i^a| \leq \mathcal{O}(\mu_g^2) \ \forall i \in \mathcal{I} \tag{6.15}$$

holds which implies

$$\begin{aligned} x_i(\alpha)s_i(\alpha) &= (1-\alpha)x_i s_i + \alpha^2 \Delta x^a{}_i \Delta s^a{}_i \\ &\geq (1-\alpha)x_i s_i - \mathcal{O}(\mu_g^2)\alpha^2 \\ &\geq \gamma(1-\alpha)\mu_g - \mathcal{O}(\mu_g^2)\alpha^2. \end{aligned}$$

This further implies that $\alpha_{\rm a} \geq 1 - \mathcal{O}(\mu_g)$. Now, for the asymptotic case, one has to estimate α that satisfies the following inequality:

$$(1-\alpha)x_i^k s_i^k + (1-\gamma)\alpha\mu - \alpha\Delta x_i^a \Delta s_i^a + \alpha^2 \Delta x_i \Delta s_i \geq \gamma(1-\alpha)\mu_g^k.$$

By using (6.3) one also has $\mu \leq \mathcal{O}\left((\mu_g^k)^4\right)$. The worst asymptotic value for α will be the result of the case when $x_i s_i = \gamma\mu_g$ and $\Delta x_i^a \Delta s_i^a > 0$. Assuming this, it is not clear wheather $\Delta x_i \Delta s_i$ is nonnegative or negative. In case of nonnegativity the previous inequality holds for a positive value of α, but if $\Delta x_i \Delta s_i < 0$, that inequality might not hold since μ is too small.

Based on the previous discussion, by a slight modification of Mehrotra's heuristic, while preserving the adaptivity of the updating scheme, one can ensure the superlinear convergence. The new adaptive updating strategy is defined by

$$\mu = \frac{\gamma t + \gamma \min(\mu_g^{\frac{1}{2}}, 1)}{1-\gamma}\mu_g, \tag{6.16}$$

where t is given by (6.9) and $0 < \gamma < \frac{1}{3}$. Changing Mehrotra's updating scheme to the new updating strategy in Algorithms 6.3 and its modified version, while preserving the safeguard, does not change their polynomial complexity. One can analogously prove the polynomiality of these algorithms by using the new definition of the centering parameter μ, and for simplicity those complexity proofs are omitted here.

Theorem 6.5.1 *Let the iterate (x^k, y^k, s^k) be generated by Algorithm 6.3 or its modified version, where μ is given by (6.16). When μ_g is sufficiently small, Algorithms 6.3 and its modified version are superlinearly convergent in the sense that $\mu_g^{k+1} = \mathcal{O}\left((\mu_g^k)^{1+r}\right)$, for some $r \in (0,1)$.*

Proof: Since $\left|(\Delta x^a)_i^k (\Delta s^a)_i^k\right| \leq \mathcal{O}((\mu_g^k)^2)\ \forall i \in \mathcal{I}$, as it is proved in Chapter 7 of [63], one analogously show that

$$\left|(\Delta x)_i^k (\Delta s)_i^k\right| \leq \mathcal{O}\left((\mu_g^k)^2\right)$$

for the new definition of μ. Therefore, the next iterate is in the neighborhood $\mathcal{N}_\infty^-(\gamma)$ whenever

$$\begin{aligned}(1-\alpha)x_i^k s_i^k + \alpha\gamma t\mu_g^k + \ \alpha\gamma \min(\mu_g^{\frac{1}{2}}, 1)\mu_g - \alpha\Delta x_i^a \Delta s_i^a + \alpha^2 \Delta x_i \Delta s_i \\ \geq \gamma(1-\alpha)\mu_g^k.\end{aligned}$$

The worst case for this inequality is might be result of $(\Delta x^a)_i^k (\Delta s^a)_i^k > 0$ and $(\Delta x)_i^k (\Delta s)_i^k < 0$. Therefore, using the definition of t, for positive component of $\Delta x^a \Delta s^a$ one also has $\Delta x_i^a \Delta s_i^a \leq t x_i s_i$. Therefore, it suffices to show that the following inequality holds for an $\alpha > 0$:

$$\begin{aligned}(1-\alpha(1+t))x_i^k s_i^k + \alpha\gamma t\mu_g^k + \alpha\gamma \min(\mu_g^{\frac{1}{2}}, 1)\mu_g - \alpha^2\mathcal{O}((\mu_g^k)^2) \\ \geq \gamma(1-\alpha)\mu_g^k. \quad (6.17)\end{aligned}$$

If (6.17) holds for $\alpha \geq \frac{1}{1+t}$, then $\alpha \geq 1 - \mathcal{O}(\mu_g^k)$, since one has $\frac{1}{1+t} \geq \frac{1}{1+\mathcal{O}(\mu_g^k)} \geq 1 - \mathcal{O}(\mu_g^k)$. Now let us assume that $\alpha < \frac{1}{1+t}$. In order to have (6.17), using the fact that $x_i^k s_i^k \geq \gamma\mu_g^k$, it suffices to have

$$(1-\alpha(1+t))\gamma\mu_g^k + \alpha\gamma t\mu_g^k + \alpha\gamma \min(\mu_g^{\frac{1}{2}}, 1)\mu_g - \alpha^2\mathcal{O}((\mu_g^k)^2) \geq \gamma(1-\alpha)\mu_g^k$$

for some positive α, which is equivalent to

$$\gamma\mu_g^{\frac{3}{2}} - \alpha\mathcal{O}((\mu_g^k)^2) \geq 0. \tag{6.18}$$

Inequality (6.18) holds definitely for $\alpha \geq 1 - \mathcal{O}\left((\mu_g^k)^{r_0}\right)$ for some $r_0 \in (0,1)$. Therefore, using (6.10) one has

$$\begin{aligned}
\mu_g^k(\alpha_c^k) &= \left(1 - \alpha_c^k(1 - \mathcal{O}((\mu_g^k)^{\frac{1}{2}})\right)\mu_g^k \\
&\leq \left(1 - (1 - \mathcal{O}((\mu_g^k)^{r_0}))(1 - \mathcal{O}((\mu_g^k)^{\frac{1}{2}}))\right)\mu_g^k \\
&\leq \mathcal{O}\left((\mu_g^k)^{1+r}\right)
\end{aligned}$$

for some $r \in (0,1)$. This gives the superlinear convergence of Algorithm 6.3 with the new choice of the parameter μ. The superlinear convergence of the modified version also can be proved analogously, which completes the proof. □

In summary, in this chapter we analyzed the worst case theoretical behavior of a feasible version of Mehrotra's original predictor-corrector algorithm whose variants are widely used in IPMs based software packages implementations. Based on a numerical example we modified the algorithm in order to ensure a minimal step size that itself implies the polynomial complexity of the algorithm. The second modification of the algorithm enjoys better iteration complexity. Further modification of centering parameter updating scheme preserves the same order of polynomial complexity while leading to asymptotic superlinear convergence of the algorithms. some encouraging computational results are reported in Chapter 8.

Chapter 7

Postponing the choice of the Barrier Parameter in Mehrotra-type IPMs

In this chapter, motivated by the example presented in Chapter 6, we propose a different approach to solve an LO problem rather than classical methods. In this approach, by fixing the step size at the beginning of the corrector step of the algorithm, we aim to compute the barrier parameter μ. We have to mention that this is in contrast with the algorithms in the previous chapters. We will prove that the proposed algorithm enjoys the same order of polynomiality without any safeguard strategy akin the one presented in Chapter 6. Some limited, encouraging, computational results based on this approach will be reported in Chapter 8. The results we present in this chapter are mostly appeared in [56].

7.1 Algorithmic Scheme

Let us first describe the structure of the algorithm throughout this chapter. In the predictor step, similar to Chapter 6, it solves the affine scaling system of equation (6.1) and in the corrector step it deals with system (6.2). Hereafter in system (6.2) μ is not a given parameter. The goal is to compute this parameter by staying in a certain neighborhood of the central path and reducing the duality gap as much as possible at the same time. Analogous to Chapter 6, here also our new algorithms operate in the negative infinity norm neighborhood $\mathcal{N}_\infty^-(\gamma)$ for $\gamma \in (0,1)$ defined in (6.4). Since we use system (6.2) frequently in this chapter, we present it here again:

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \Delta x^a \Delta s^a. \end{aligned} \tag{7.1}$$

The following lemma gives information about the complementarity gap after one step. The proof is a direct consequence of the third equation of (7.1).

Lemma 7.1.1 *Let g^+ be the complementarity gap after one step with a predetermined step size α. Then*

$$g^+ = (1-\alpha)g + \alpha n\mu.$$

7.1.1 Estimation of the Barrier Parameter

In this subsection we aim to compute μ based on certain conditions. To do so, let us first derive the explicit solution of (7.1) as a function of μ. From the third equation of (7.1) one has

$$\Delta s = X^{-1}(\mu e - xs - \Delta x^a \Delta s^a - s\Delta x).$$

Then, from the second equation one has

$$\Delta y = -(ADA^T)^{-1}AS^{-1}(\mu e - xs - \Delta x^a \Delta s^a - s\Delta x),$$

where $D = \text{diag}\,(xs^{-1})$. Let us denote $P = (ADA^T)^{-1}AS^{-1}$ that allows us to write

$$\Delta y = -\mu Pe + PXs + P\text{diag}\,(\Delta x^a)\Delta s^a.$$

Using again the second equation of (7.1) one has

$$\Delta s = \mu A^T Pe - A^T PXs - A^T P\text{diag}\,(\Delta x^a)\Delta s^a.$$

Finally, from the third equation one can get

$$\begin{aligned}\Delta x = \mu S^{-1}(e - XA^T Pe) - x \\ -S^{-1}\left(\text{diag}\,(\Delta x^a)\Delta s^a - XA^T PXs - XA^T P\text{diag}\,(\Delta x^a)\Delta s^a\right).\end{aligned}$$

Therefore, one has the following representation for $(\Delta x, \Delta y, \Delta s)$ as a function of μ:

$$\begin{aligned}\Delta x &= \mu p_x^1 + p_x^2, \\ \Delta y &= \mu p_y^1 + p_y^2, \\ \Delta s &= \mu p_s^1 + p_s^2,\end{aligned} \tag{7.2}$$

where

$$\begin{aligned}p_x^1 &= S^{-1}(e - XA^T Pe), \\ p_x^2 &= -x - S^{-1}\left(\text{diag}\,(\Delta x^a)\Delta s^a - XA^T PXs - XA^T P\text{diag}\,(\Delta x^a)\Delta s^a\right), \\ p_y^1 &= -Pe, \\ p_y^2 &= PXs + P\text{diag}\,(\Delta x^a)\Delta s^a,\end{aligned}$$

$$p_s^1 = A^T Pe,$$
$$p_s^2 = -A^T PXs - A^T P\text{diag}\,(\Delta x^a)\Delta s^a.$$

Now the goal is to solve the following one dimensional optimization problem in order to estimate the parameter μ with an already chosen target step size α_t.[1]

$$\begin{aligned} \min \quad & (1-\alpha_t)g + \alpha_t n\mu_t \\ & (x(\alpha_t), s(\alpha_t)) \in \mathcal{N}_\infty^-(\gamma), \end{aligned} \tag{7.3}$$

where $x(\alpha_t) = x + \alpha_t \Delta x$ and $s(\alpha_t) = s + \alpha_t \Delta s$. We will show in the sequel that problem (7.3) is solvable for properly chosen α_t values and we give an upper bound for the optimal value of (7.3). To do so, after each corrector step with a given step size α_t one has

$$\begin{aligned} x(\alpha_t)s(\alpha_t) &= (x + \alpha_t\Delta x)\,(s + \alpha_t \Delta s) \\ &= (1-\alpha_t)xs + \alpha_t\mu_t e - \alpha_t \Delta x^a \Delta s^a + \alpha_t^2 \Delta x \Delta s \\ &= (1-\alpha_t)xs + \alpha_t\mu_t e - \alpha_t \Delta x^a \Delta s^a + \alpha_t^2\mu_t^2 p_x^1 p_s^1 + \alpha_t^2 \mu_t (p_x^1 p_s^2 + p_x^2 p_s^1) \\ &\quad + \alpha_t^2 p_x^2 p_s^2 \\ &= \alpha_t^2 p_x^1 p_s^1 \mu_t^2 + (\alpha_t e + \alpha_t^2 p_x^1 p_s^2 + \alpha_t^2 p_x^2 p_s^1)\mu_t + (1-\alpha_t)xs + \alpha_t^2 p_x^2 p_s^2 \end{aligned}$$

and

$$\mu_g(\alpha_t) = (1-\alpha_t)\mu_g + \alpha_t \mu_t. \tag{7.4}$$

Therefore, the constraint of (7.3) is equivalent to

[1]The subscript 't' for α_t and μ_t denotes the target step size and penalty parameter.

$$\alpha_t^2 p_x^1 p_s^1 \mu_t^2 + (\alpha_t e + \alpha_t^2 p_x^1 p_s^2 + \alpha_t^2 p_x^2 p_s^1 - \gamma\alpha_t e)\mu_t + (1-\alpha_t)xs + \alpha_t^2 p_x^2 p_s^2 - \gamma(1-\alpha_t)\mu_g e \geq 0, \quad (7.5)$$

i.e., n quadratic inequalities with μ_t as a variable. It is worth mentioning that in advance we do not have any information about the convexity or concavity of these polynomials.

In what follows we give an upper bound for the optimal solution of problem (7.3) μ_t by fixing the target step size α_t. This estimate enables us to prove the polynomial iteration complexity of the new algorithm that we will outline in sequel.

Let us fix $\alpha_t = \frac{\gamma^2}{n^2}$; then the ith constraint of (7.5) holds if

$$x_i(\alpha_t)s_i(\alpha_t) \geq \gamma\mu_g(\alpha_t).$$

After expanding this inequality using (7.1) and (7.4) one has

$$(1-\alpha_t)x_i s_i + \alpha_t(1-\gamma)\mu - \alpha_t \Delta x_i^a \Delta s_i^a + \alpha_t^2 \Delta x_i \Delta s_i \geq \gamma(1-\alpha_t)\mu_g. \quad (7.6)$$

We get the worst bound if $\Delta x_i^a \Delta s_i^a > 0$ and $\Delta x_i \Delta s_i < 0$. By Lemma 6.3.1 we also know that $\Delta x_i^a \Delta s_i^a \leq \frac{x_i s_i}{4}$. Therefore, using this inequality and the fact that the previous iterate is in $\mathcal{N}_\infty^-(\gamma)$, inequality (7.6) certainly holds if

$$-\frac{\gamma\mu_g}{4} + (1-\gamma)\mu - \frac{\gamma^2 |\Delta x_i \Delta s_i|}{n^2} \geq 0.$$

Using Lemma 6.3.3, the previous inequality is true if

$$-\frac{\gamma\mu_g}{4} + (1-\gamma)\mu - \frac{2^{\frac{-3}{2}}\gamma\mu^2}{\mu_g n} - \frac{2^{\frac{-3}{2}}17\gamma^2\mu_g}{16n} - \frac{2^{\frac{-3}{2}}\gamma\mu_g}{16} + \frac{2^{\frac{-1}{2}}\gamma^2\mu}{n} - \frac{2^{\frac{-3}{2}}\gamma\mu}{2n} \geq 0.$$

After reordering and simplifying one has

$$-\frac{\gamma}{2\sqrt{2}\mu_g n}\mu^2 + \left(\frac{\gamma^2}{\sqrt{2}n} - \frac{\gamma}{4\sqrt{2}n} + 1 - \gamma\right)\mu - \frac{(8\sqrt{2}+1)\gamma\mu_g}{32\sqrt{2}} - \frac{17\gamma^2\mu_g}{32\sqrt{2}n} \geq 0. \tag{7.7}$$

The following lemma gives the interval for which inequality (7.7) holds.

Lemma 7.1.2 *For $\gamma \in (0, 0.65)$ and $n \geq 2$ the step size $\alpha_t = \frac{\gamma^2}{n^2}$ is feasible for any*

$$\mu_t \in [\beta_1\mu_g \quad \beta_2\mu_g],$$

where

$$\beta_1 = \frac{-4\sqrt{2}\gamma n + 4\sqrt{2}n + 4\gamma^2 - \gamma - \sqrt{\Delta}}{4\gamma}$$

and

$$\beta_2 = \frac{-4\sqrt{2}\gamma n + 4\sqrt{2}n + 4\gamma^2 - \gamma + \sqrt{\Delta}}{4\gamma}$$

and

$$\Delta = 32\gamma^2 n^2 - 64\gamma n^2 - 32\sqrt{2}\gamma^3 n + (32\sqrt{2}-1)\gamma^2 n + 32n^2 - 8\sqrt{2}\gamma n + 16\gamma^4 - 25\gamma^3 + \gamma^2.$$

Moreover $0 < \beta_1 < \frac{7}{8}$.

Proof: For the worst case analysis it suffices to find the interval for which inequality (7.7) holds. By simple calculus one may verify that $\beta_1\mu_g$ and $\beta_2\mu_g$ are two real roots of the second order polynomial in (7.7) if and only if $\Delta \geq 0$, which holds for all $\gamma \in (0, 0.73)$ and $n \geq 2$. Finally we have to show that $0 < \beta_1 < \frac{7}{8}$. To this end, using the definition of β_1, $\beta_1 > 0$ is equivalent to

$(8\sqrt{2}+1)\gamma^2 n + 17\gamma^3 > 0$, which is definitely true. Analogously, $\beta_1 < \frac{7}{8}$ is equivalent to $(144\sqrt{2}+4)\gamma^2 n - 112\sqrt{2}\gamma n - 44\gamma^3 + 77\gamma^2 < 0$. This inequality holds for all $\gamma \in (0, 0.68)$ and $n \geq 2$ that completes the proof of the lemma. □

Corollary 7.1.1 *The optimal value μ_t of problem (7.3) satisfies $\mu_t \leq \beta_1 \mu_g$.*

Remark 7.1.1 *Because $\mu_t \leq \beta_1 \mu_g \leq \frac{7}{8}\mu_g$, our new algorithm features a large update of parameter μ at each step.*

Now, let us outline the scheme of our new algorithm.

Algorithm 7.1.1

Input:

A neighborhood parameter $\gamma \in (0, 0.68)$;

$\theta \in (0, 1)$ a target value's update parameter;

an accuracy parameter $\epsilon > 0$; $(x^0, y^0, s^0) \in \mathcal{N}_\infty^-(\gamma)$.

begin

while $x^T s > \epsilon$ **do**

begin

Predictor Step

Solve (6.1) for $(\Delta x^a, \Delta y^a, \Delta s^a)$.

end

begin

Corrector step

Let $\alpha_t \in (0, 1]$;

Solve (7.3) with α_t to determine $\mu_t \geq 0$;

While $\mu_t < 0$

Let $\alpha_t = \theta\alpha_t$ and solve (7.3) with α_t to determine $\mu_t \geq 0$;

end

Compute $(\Delta x, \Delta y, \Delta s)$ by (7.2);

Compute the maximum step size α_c such that

$(x(\alpha_c), y(\alpha_c), s(\alpha_c)) \in \mathcal{N}_\infty^-(\gamma)$;

Set $(x(\alpha_c), y(\alpha_c), s(\alpha_c)) = (x + \alpha_c\Delta x, y + \alpha_c\Delta y, s + \alpha_c\Delta s)$.

end

end

end

7.1.2 Iteration Complexity

The following theorem gives the maximum number of iterations of Algorithm 7.1.1.

Theorem 7.1.1 *If we fix $\alpha_t = \frac{\gamma^2}{n^2}$ in the corrector step, then Algorithm 7.1.1 stops after at most*

$$O\left(n^2 \log \frac{n}{\epsilon}\right)$$

iterations.

Proof: After each iteration one has

$$x(\alpha)^T s(\alpha) = \left(1 - \frac{\gamma^2}{n^2} + \frac{\gamma^2 \mu_t}{\mu_g n^2}\right) x^T s.$$

By Corollary 7.1.1 $\mu_t \leq \beta_1 \mu_g$ and by Lemma 7.1.2 we have $0 < \beta_1 < \frac{7}{8}$, therefore

$$x(\alpha)^T s(\alpha) \leq \left(1 - \frac{\gamma^2}{8n^2}\right) x^T s,$$

which completes the proof by Theorem 3.2 of [63]. □

Remark 7.1.2 *We see that the iteration complexity of Algorithm 7.1.1 in the worst case is as the same order as the one presented in [53], but Algorithm 7.1.1 is not using any safeguard strategy like the one in [53].*

7.2 A Modified Version of Algorithm 7.1.1

In this section we slightly modify the corrector step of Algorithm 7.1.1 which reduces the iteration complexity significantly. The motivation for this modification is exactly the same as the one in Chapter 6 so we omitted it here.

This modification requires the maximum step size in the predictor step, which was not the case in Algorithm 7.1.1. Therefore, the modified algorithm differs from Algorithm 7.1.1 both in the predictor and in the corrector steps.

Analogous to Chapter 6, in this case we moved a factor of the maximum step size in the predictor step to the right hand side of the third equation of the corrector step. The new system is given by

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - xs - \alpha_{\mathrm{a}}\Delta x^a \Delta s^a. \end{aligned} \tag{7.8}$$

Analogous to the previous section one has a representation of $(\Delta x, \Delta y, \Delta s)$ as a function of the barrier parameter μ. Therefore, problem (7.3) has to be solved in order to estimate the parameter μ for this modified version.

7.2.1 Estimation of the Barrier Parameter

Now analogous to the analysis of Algorithm 7.1.1, we have to solve the one dimensional optimization problem (7.3) in order to estimate the parameter μ_t. For the worst case analysis the ith constraint of (7.3) for $\alpha_t = \frac{\gamma^2}{n}$ holds if

$$-\frac{\gamma\mu_g}{4} + (1-\gamma)\mu_t - \frac{\gamma^2\left|\Delta x_i \Delta s_i\right|}{n} \geq 0.$$

By Lemma 6.4.2 the previous inequality holds if

$$-\frac{\gamma\mu_g}{4} + (1-\gamma)\mu_t - \frac{2^{\frac{-3}{2}}\gamma\mu_t^2}{\mu_g} - 2^{\frac{-3}{2}}\gamma^2\mu_g - \frac{2^{\frac{-3}{2}}\gamma^2\alpha_{\mathrm{a}}^2\mu_g}{16} - \frac{2^{\frac{-3}{2}}\gamma^2(1-\alpha_{\mathrm{a}})\mu_g}{4} + 2^{\frac{-1}{2}}\gamma^2\mu_t - \frac{2^{\frac{-3}{2}}\alpha_{\mathrm{a}}\gamma\mu_t}{2} \geq 0.$$

After reordering we have

$$-\frac{\gamma}{2\sqrt{2}\mu_g}\mu_t^2+\left(\frac{\gamma^2}{\sqrt{2}}-\frac{\gamma\alpha_{\rm a}}{4\sqrt{2}}+1-\gamma\right)\mu_t-\frac{\gamma^2\mu_g}{2\sqrt{2}}-\frac{\gamma\mu_g}{4}-\frac{\gamma^2(1-\alpha_{\rm a})\mu_g}{8\sqrt{2}}$$
$$-\frac{\gamma^2\alpha_{\rm a}^2\mu_g}{32\sqrt{2}}\geq 0. \quad (7.9)$$

In the following lemma we determine the interval for which the previous inequality holds.

Lemma 7.2.1 *For $\gamma\in(0,0.62)$ and $n\geq 2$ the step size $\alpha_t=\frac{\gamma^2}{n}$ is feasible for any*

$$\mu\in[\eta_1\mu_g \quad \eta_2\mu_g],$$

where

$$\eta_1=\frac{-4\sqrt{2}\gamma+4\gamma^2-\gamma\alpha_{\rm a}+4\sqrt{2}-\sqrt{\Delta}}{4\gamma}$$

$$\eta_2=\frac{-4\sqrt{2}\gamma+4\gamma^2-\gamma\alpha_{\rm a}+4\sqrt{2}+\sqrt{\Delta}}{4\gamma}$$

with

$$\Delta=(32+24\sqrt{2})\gamma^2-(32\sqrt{2}+20)\gamma^3+8\sqrt{2}\gamma^2\alpha_{\rm a}-64\gamma+16\gamma^4-4\gamma^3\alpha_{\rm a}$$
$$+\gamma^2\alpha_{\rm a}^2-8\sqrt{2}\gamma\alpha_{\rm a}+32-\gamma^3\alpha_{\rm a}^2.$$

Moreover, $0<\eta_1<0.9$.

Proof: For the worst case analysis it suffices to find the interval for which inequality (7.9) holds. By simple calculus one can show that $\eta_1\mu_g$ and $\eta_2\mu_g$ are two real roots of inequality (7.9) if and only if $\Delta\geq 0$, which holds for all $\gamma\in(0,0.64)$. By using the definition of $\eta_1>0$ is equivalent to $8\sqrt{2}\gamma^2+20\gamma^3-4\gamma^3\alpha_{\rm a}+\gamma^3\alpha_{\rm a}^2>0$, which is definitely true for all $\gamma\in(0,1)$. Finally, $\eta_1<0.9$ is

equivalent to $(920\sqrt{2}+324)\gamma-220\gamma^2+180\gamma\alpha_{\rm a}-720\sqrt{2}-100\gamma^2\alpha_{\rm a}+25\gamma^2\alpha_{\rm a}^2<0$, which holds for all $\gamma\in(0,0.62)$. This completes the proof of the lemma. □

Corollary 7.2.1 *The optimal value μ_t of problem (7.3) satisfies $\mu_t\leq\eta_1\mu_g$.*

Remark 7.2.1 *Because $\mu_t\leq\eta_1\mu_g\leq 0.9\mu_g$, Algorithm 7.2.1 features a large update of parameter μ at each step.*

The scheme of the modified version of Algorithm 7.1.1 can be outlined as Algorithm 7.2.1:

Algorithm 7.2.1

Input:

A neighborhood parameter $\gamma \in (0, 0.62)$; $\theta \in (0, 1)$;

an accuracy parameter $\epsilon > 0$; $(x^0, y^0, s^0) \in \mathcal{N}_\infty^-(\gamma)$.

begin

while $x^T s > \epsilon$ **do**

begin

Predictor Step

Solve (6.1) and compute the maximum step size $\alpha_{\rm a}$ such that $(x(\alpha_{\rm a}), y(\alpha_{\rm a}), s(\alpha_{\rm a})) \in \mathcal{F}$ (algorithm does not make this step now).

end

begin

Corrector step

Let $\alpha_t \in (0, 1]$;

Solve (7.3) with given α_t to determine the target $\mu_t \geq 0$;

While $\mu_t < 0$

Let $\alpha_t = \theta \alpha_t$ and solve (7.3) to determine the target $\mu_t \geq 0$;

end

Compute $(\Delta x, \Delta y, \Delta s)$ by (7.2);

Compute the maximum step size $\alpha_{\rm c}$ such that

$(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) \in \mathcal{N}_\infty^-(\gamma)$;

Set $(x(\alpha_{\rm c}), y(\alpha_{\rm c}), s(\alpha_{\rm c})) = (x + \alpha_{\rm c}\Delta x, y + \alpha_{\rm c}\Delta y, s + \alpha_{\rm c}\Delta s)$.

end

end

end

7.2.2 Iteration Complexity

The following theorem gives the maximum number of iterations that Algorithm 7.2.1 needs to perform to find an $\epsilon-$approximate solution.

Theorem 7.2.1 *If we fix $\alpha_t = \frac{\gamma^2}{n}$ in the corrector step, then Algorithm 7.2.1 stops after at most*

$$O\left(n \log \frac{n}{\epsilon}\right).$$

Proof: After each iteration one has

$$x(\alpha)^T s(\alpha) = \left(1 - \frac{\gamma^2}{n} + \frac{\gamma^2 \mu_t}{\mu_g n}\right) x^T s.$$

By Corollary 7.2.1 $\mu_t \leq \eta_1 \mu_g$ and by Lemma 7.2.1 $0 < \beta_1 < 0.9$, therefore

$$x(\alpha)^T s(\alpha) \leq \left(1 - \frac{\gamma^2}{10n}\right) x^T s,$$

which completes the proof by Theorem 3.2 of [63]. □

We see that the iteration complexity of Algorithm 7.2.1 is as the same order of the modified algorithm in Chapter 6, but it is not using any safeguard strategy like the one in [53].

7.3 Superlinear Convergence

In this section we prove the superlinear convergence of both algorithms presented in this chapter. As we know, for sufficiently small μ_g it has been proved in [63] (Chapter 7) that $|\Delta x_i^a \Delta s_i^a| \leq \mathcal{O}(\mu_g^2) = C\mu_g^2$, which further implies

$$\begin{aligned} x_i(\alpha)s_i(\alpha) &= (1-\alpha)x_i s_i + \alpha^2 \Delta x_i^a \Delta s_i^a \geq (1-\alpha)x_i s_i - \mathcal{O}(\mu_g^2)\alpha^2 \\ &\geq \gamma(1-\alpha)\mu_g - \mathcal{O}(\mu_g^2)\alpha^2 \end{aligned} \tag{7.10}$$

This last ineauality further implies $\alpha_{\rm a} \geq 1 - \mathcal{O}(\mu_g)$. In the following theorem we prove the superlinear convergence of Algorithms 7.1.1 and 7.2.1.

Theorem 7.3.1 *Let (x^k, y^k, s^k) be the iterate generated by Algorithm 7.1.1 or Algorithm 7.2.1. When μ_g is sufficiently small, by postponing the choice of the barrier parameter μ, both algorithms are superlinearly convergent in the sense that $\mu_g^{k+1} = \mathcal{O}\left((\mu_g^k)^{1+r}\right)$, where $r \in (0,1)$.*

Proof: Let $\alpha = 1 - \mu_g^{r_0}$ for some $r_0 \in (0,1)$. Then, for $\mu = \frac{2C}{1-\gamma}\mu_g^{1+r}$, where $r \in (0, r_0)$, analogous to the proof of Theorem 7.4 in [63] one can prove that $|\Delta x_i \Delta s_i| \leq \mathcal{O}(\mu_g^2)\ \forall i \in \mathcal{I}$. Therefore, the next iterate belongs to the neighborhood whenever

$$2C\mu_g^{1+r} - C\mu_g^2 - (1 - \mu_g^{r_0})\mathcal{O}(\mu_g^2) \geq 0,$$

which is definitely true for sufficiently small μ_g. This implies the superlinear convergence of Algorithm 7.1.1. The superlinear convergence of Algorithm 7.2.1 can be proved analogously. This completes the proof. □

In summary, in this chapter we presented a new way of analyzing the Mehrotra-type predictor-corrector algorithm that is different than the existing approach in IPMs literature. We proved the algorithms in this chapter enjoy the same order of the polynomial iteration complexity as the algorithms of Chapter 6. However, they are superlinearly convergent while the algorithms presented in chapter 6 achieve this property by changing the updating scheme of the barrier (centering) parameter.

Chapter 8

Computational Results

In this chapter we report some limited computational results for the algorithms of Chapters 5 to 7. The aim of these computational results is to demonstrate the practical performance of the proposed algorithms compared to some state of the art software packages [70, 73]. For implementation of our algorithms we use the McIPM and LIPSOL software packages. The implementation is done in MATLAB 6.5.1 by modifying the source code of McIPM and LIPSOL on a pentium 4, 2.53 GHZ machine with 512 MB RAM. In Tables 8.1 and 8.2 we give some information about the test problems that are chosen from the standard NETLIB [40] test repository for LO problems.

8.1 The McIPM and LIPSOL Software Packages

Let us first briefly describe the McIPM software package. It has been developed by X. Zhu at the Advanced Optimization Laboratory (AdvOL) at

McMaster University [72]. It uses the self-dual embedding model approach discussed in Section 10.2. The implementation is based on the Mehrotra-type predictor-corrector algorithm discussed in Chapter 6, and uses a dynamic approach to take advantages of SR search directions. First it solves the affine scaling system (6.1). Then it computes the parameter μ using formula (6.3) and finally makes the step in the direction computed from the system (6.2). If that direction is not giving a satisfactory result, then it uses an SR direction with a higher q value to bring the iterate closer to the central path, so that the algorithm will be able to make better progress in the next iteration. This is done by choosing μ_q^* as the target value (see Chapter 5 for more details). Then, the new iterate is made like Mehrotra-type algorithm in Chapter 6. Numerical results show the efficiency of the dynamic strategy compared to the pure SR approach. To compute the Newton direction it uses the normal equation approach rather than the augmented system approach [72] (see Section 10.5 for more details). On the other hand, the LIPSOL software package, developed by Y. Zhang at Rice University, implements an infeasible variant of the Mehrotra-type algorithm [70] using the classical approach. It uses the same strategy to find an starting point as discussed in Section 10.3 of the Appendix with some minor modifications. Analogous to the McIPM package, it also uses the normal equation approach to calculate the search direction. Both of these software packages are able to deal with dense columns in the coefficient matrix efficiently.

8.2 Implementations

Now let us describe the details of the implementations of the developed algorithms in Chapters 5, 6 and 7.

8.2.1 Algorithm 5.2

We implemented a Mehrotra-type variant of Algorithm 5.2 which uses the dynamic strategy of [72] as we explained it briefly in the previous section. Due to numerical difficulty[1], we report the results for the case $q = 3$ in the last columns of Tables 8.3 and 8.4 (SRMcIPM). In these tables we compare our new algorithm with some other algorithms that we will discuss in the sequel. These limited computational results shows that using the SR approach in the predictor step whenever it performs better than the affine scaling direction is at least as good as the pure affine scaling direction. Therefore, it can be cast as an alternative approach when the affine scaling performs very poorly.

8.2.2 Algorithm 6.3

The implementation of Algorithm 6.3 is done by using both the McIPM and the LIPSOL software packages. If the affine scaling performs very poorly i.e., $\alpha_a < 0.05$, then we use the Newton system which is considered in the modified version of Algorithm 6.3. In other words, a combined version of the algorithms in Chapter 6 is used for implementation. For simplicity the following abbreviations are used for different variants of the algorithm using

[1]Since some of the components of primal and dual variables during the iterates might get smaller, then using higher q values might make them even much smaller and subsequently the algorithm might run to numerical difficulties.

the McIPM software package.

- **PMMcIPM:** Mehrotra's original algorithm without any heuristics in the definition of the centering parameter determination.
- **HMcIPM:** Mehrotra's original algorithm combined with heuristics used in the definition of the centering parameter.
- **NMcIPMI:** Mehrotra's original algorithm combined with the large update safeguard proposed in Chapter 6.
- **NMcIPMII:** Mehrotra's original algorithm with the new definition of the centering parameter (6.16) instead of using (6.3) combined with the proposed large update safeguard.

The following abbreviations also are used for the LIPSOL software package based implementation.

- **PLIPSOL:** Mehrotra's algorithm without any heuristics in the definition of the centering parameter.
- **HLIPSOL:** Mehrotra's original algorithm with heuristics used in the definition of the centering parameter.
- **SLIPSOL:** Mehrotra's algorithm combined with a simple large update safeguard.

We have to mention that the parameter γ and β are fixed to be 0.0001 and 0.1, respectively, in both implementations. Our limited computational results in Tables 8.3, 8.4, 8.5, and 8.6 show that employing the safeguard in the algorithms not only theoretically guarantees polynomial iteration complexity, but

also preserves the efficiency of the algorithms and sometimes even outperforms the heuristics. We have not included the results using our new definition of the barrier parameter for LIPSOL, since we have not done a detailed analysis for the infeasible case.

By close looking at the step size in the predictor step, the duality gap of the current iterate, and the position of the point in the feasible region, we observed that aggressive update of the barrier (centering) parameter and being closer to the boundary of the feasible region might imply a very small step size in the corrector step and require many iterations to find the desired solution. However, in our new safeguard based algorithms, in case of a bad step using the simple safeguard we proposed leads to competitive number of iterations compared to the algorithms that are using heuristics. It is also worth mentioning that significant differences in times occur when number of iterations are significantly different, specially for high dimensional problems, for example 'pds20', 'osa-60.'

8.2.3 Algorithm 7.1.1

The implementation of the Algorithm 7.1.1 is done by using the LIPSOL package. The predictor step of the implemented algorithm is the same as in other Mehrotra-type predictor-corrector algorithms, namely it solves the affine scaling system of equations (6.1), without actually making this step right away (one does not compute the maximum step size in this direction in Algorithm 7.1.1, while in Algorithm 7.2.1 one does). Then, by using the search direction computed in the predictor step, one computes the target parameter μ_t by solving problem (7.3) (for the infeasible case this problem is slightly different than (7.3)). Since there is no information regarding the convexity or

concavity of the quadratic polynomials in (7.3), we use a simple line search technique to solve problem (7.3) (it can be solved directly in $\mathcal{O}(n)$ arithmetic bits operations). If the affine scaling performs very poor i.e., $\alpha_{\rm a} \leq 0.05$, then we consider the Newton system in the modified version of the algorithm. This way both algorithms of Chapter 7 are used for the implementation. Postponing the choice of the barrier parameter is the major difference between our new algorithm and classical IPMs. We report the computational results on some NETLIB and the Kennington test problems. This version of LIPSOL is denoted by POLIPSOL in the last columns of Tables 8.5 and 8.6.

For all test problems we choose $\alpha_t = 1 - \min(\mu_g^{\frac{1}{2}}, 0.1)$ and $\gamma = 10^{-4}$. If there is no feasible solution of problem (7.3) for this fix step size, then we reduce α_t by a constant factor of 0.7. These limited computational results show that postponing the barrier parameter as an implicit safeguard not only has the same theoretical efficiency as the algorithms of Chapter 6, and enjoys better asymptotic convergence, but also is practically competitive with those algorithms.

Problem	Rows	Columns	Problem	Rows	Columns
25fv47	798	1854	pilotja	924	2044
80bau3b	2235	11516	pilotwe	722	2930
adlittle	55	137	pilot4	402	1137
afiro	27	51	pilot87	2030	6460
bandm	269	436	pilotnov	951	2242
brandy	149	259	scagr25	471	671
czprob	930	14173	scagr7	129	185
degen3	1503	2604	stocfor1	109	157
forplan	135	463	stocfor2	2157	3045
ganges	1137	1534	stocfor3	16675	23541
perold	625	1530	truss	1000	8806
pilot	1441	4657	tuff	292	617

Table 8.1: Test Problems from NETLIB

Problem	Rows	Columns	Problem	Rows	Columns
cre-a	3517	4067	osa-07	1119	23949
cre-b	9649	72477	osa-14	2338	52460
cre-c	3069	3678	osa-30	4351	100024
cre-d	8927	69980	osa-60	10281	232966
ken-07	2427	3602	pds-02	2954	7535
ken-11	14695	21349	pds-06	9882	28655
ken-13	28633	42659	pds-10	16559	48763
ken-18	105128	154699	pds-20	33875	105728

Table 8.2: The Kennington Test Problems

Problem	HMcIPM	PMMcIPM	NMcIPMI	NMcIPMII	SRMcIPM
80bau3b	42	43	43	42	41
25fv47	27	28	28	28	27
afiro	10	10	10	10	10
adlittle	14	14	14	14	14
bandm	18	17	17	18	17
brandy	18	18	18	17	18
cycle	39	40	38	39	36
degen3	14	43	22	14	14
forplan	31	30	29	29	30
ganges	20	21	21	20	20
perold	42	48	48	42	41
pilot	48	53	58	53	49
pilotja	43	51	44	42	42
pilotwe	42	45	44	41	42
pilot4	37	37	37	37	37
pilot87	79	76	71	69	72
pilotnov	26	28	28	26	26
scagr7	13	14	14	13	14
scagr25	15	17	17	16	16
stocfor1	14	14	14	14	14
stocfor2	33	31	31	30	31
stocfor3	50	50	50	48	49
truss	20	21	21	20	20
tuff	19	20	20	19	19

Table 8.3: Comparison of Iterations Number for Variants of McIPM on NETLIB Test Problems

Problem	HMcIPM	PMMcIPM	NMcIPMI	NMcIPMII	SRMcIPM
cre-a	29	28	29	27	27
cre-b	34	37	36	34	34
cre-c	32	31	31	32	33
cre-d	32	35	34	33	33
ken-07	17	17	17	17	17
ken-11	20	21	21	20	20
ken-13	26	29	28	27	26
ken-18	35	37	36	33	33
osa-07	40	33	31	34	35
osa-14	52	38	39	41	36
osa-30	44	44	41	44	44
osa-60	57	51	48	43	47
pds-02	32	34	33	32	32
pds-06	45	51	50	43	45
pds-10	58	70	69	57	56
pds-20	81	96	85	79	77

Table 8.4: Comparison of Iterations Number for Variants of McIPM on the Kennington Test Problems

Problem	HLIPSOL	PLIPSOL	SLIPSOL	POLIPSOL
80bau3b	42	46	41	41
25fv47	24	27	23	24
afiro	8	8	8	9
adlittle	13	13	12	13
bandm	18	18	16	16
brandy	17	18	16	15
czprob	34	36	36	37
degen3	23	26	20	20
forplan	23	25	23	21
ganges	20	19	19	20
perold	31	38	35	32
pilot	31	39	27	31
pilotja	31	34	32	30
pilotwe	37	40	33	35
pilot4	29	31	37	31
pilot87	42	43	43	41
pilotnov	20	20	19	19
scagr7	14	14	13	14
scagr25	19	17	17	18
stocfor1	19	15	20	18
stocfor2	22	24	24	23
stocfor3	33	36	36	34
truss	19	20	18	19
tuff	20	23	18	18

Table 8.5: Comparison of Iteration Numbers for Variants of LIPSOL on NETLIB Test Problems

Problem	HLIPSOL	PLIPSOL	SLIPSOL	POLIPSOL
cre-a	30	33	29	29
cre-b	42	45	37	41
cre-c	30	32	31	30
cre-d	38	47	38	38
ken-07	16	16	15	16
ken-11	22	23	21	22
ken-13	27	30	28	28
ken-18	38	42	42	39
osa-07	27	29	27	21
osa-14	37	30	34	28
osa-30	36	37	39	31
osa-60	34	39	38	31
pds-02	29	29	28	29
pds-06	42	44	45	45
pds-10	52	58	55	54
pds-20	67	69	63	67

Table 8.6: Comparison of Iterations Number for Variants of LIPSOL on the Kennington Test Problems

Chapter 9

Conclusions and Future Works

In this chapter we present a summary of the main results of the thesis and outline some future research directions.

First we gave a brief overview of LO problems. Then we outlined some of their fundamental properties, such as duality results. In Chapter 2 we described the basics of IPMs for LO and then we introduced the family of self-regular functions and self-regular IPMs. In chapter 3, using a specific family of self-regular functions we proposed an adaptive algorithm for LO that enjoys better iteration complexity than the classical analogues. The adaptive updating scheme was one of the significant feature of our new algorithm. Using the same adaptive technique a new variant of infeasible IPMs based on a specific self-regular function is proposed in Chapter 4. The worst case analysis of this algorithm reveals that it enjoys better iteration complexity than the classical analogues. By further use of the adaptive technique developed in Chapter 3, we designed a new predictor-corrector algorithm that differs from the classical predictor-corrector algorithms both in the predictor and the corrector steps in Chapter 5. It employs SR search directions both in the predictor and corrector

steps, namely it uses the adaptive updating scheme in the predictor step and a SR search direction with μ_q^* in the corrector step. We proved that the new algorithm in some special cases enjoys better iteration complexity than the classical analogues. Moreover it converges superlinearly. However, the algorithms proposed in Chapters 3 to 5–in the worst case–might do a bad step after each good step. Therefore, it is interesting to consider the issue:

- How to avoid bad steps in those algorithms?

One might consider the following issue for further investigation as well.

- Design a hybrid algorithm that uses always the classical Newton search directions unless its necessary to take advantages of self-regularity to bring the iterate closer to the central path.

A more practical algorithm [34] is analyzed in Chapter 6 whose variants have been used by most IPMs based software packages [70, 73]. We showed by an example that this algorithm might suffer from some drawbacks that might make the algorithm inefficient in practice. The example motivated us to combine the algorithm with a safeguard that gives a lower bound proportional to the problem size for the maximum step size at each iteration. We proved that the algorithm has an $\mathcal{O}\left(n^2 \log \frac{n}{\epsilon}\right)$ iteration complexity. The second modification of the algorithm enabled us to reduce the iteration complexity to $\mathcal{O}\left(n \log \frac{n}{\epsilon}\right)$. A slight modification of the centering parameter updating strategy allows us to prove the superlinear convergence of the algorithms, while keeping the same order of iterations complexity. We made computational experiments with our new algorithms by using the McIPM and LIPSOL software packages [70, 73]. Computational results show that our new algorithms are competitive with the ones implemented in those software packages while our

algorithms enjoy stronger theoretical properties. Further, the following ideas might be considered for future developments:

- Analyze the theoretical and practical performance of our Mehrotra-type algorithms combined with the idea of self-regularity.

  As we know preprocessing is one the most important element of a successful implementation. However, finding all the redundant constraints of an LO problem is more difficult than solving the LO problem itself. Therefore, it is interesting to consider:

- The effect of redundancy on the performance of Mehrotra-type algorithms.

  The first order Mehrotra-type algorithms as they presented in Chapter 6 are the backbone of most IPMs based software packages. However, it is worthwhile to:

- Analyze the behavior of the second order Mehrotra-type algorithm [35] based on the observation of Chapter 6.

  Since several IPMs based software packages are using infeasible interior point algorithms, therefore it is natural to:

- Analyze the infeasible variant of the Mehrotra-type algorithms that might give some useful insights about the performance of the infeasible IPMs implementations.

Motivated by the observations of Chapter 6, we analyzed a Mehrotra-type algorithm from a different point of view. As is done in the IPMs literature, one first determines the barrier parameter μ, then calculates the Newton

direction, finally the algorithm makes a step with a certain step size in the computed direction. In this new approach, first we fix the minimum step size in the corrector step, then by solving a one dimensional optimization problem we calculate the parameter μ. Finally, by computing the search direction and appropriate step size the algorithm can move to the next iterate. We proved that the algorithm has an $\mathcal{O}\left(n^2 \log \frac{n}{\epsilon}\right)$ worst case iteration complexity with superlinear convergence. A slightly modified version of the algorithm is presented that enjoys better iteration complexity i.e., $\mathcal{O}\left(n \log \frac{n}{\epsilon}\right)$. It is worth mentioning that unlike the algorithms in Chapter 6 these algorithms do not require any change of the barrier parameter to guarantee the superlinear convergence. Our limited computational results using the LIPSOL software package show the efficiency and competitiveness of this new approach in addition its appealing theoretical properties.

Due to its strong theoretical properties and the encouraging computational results, one might consider the following questions for further investigation:

- Use this idea in the framework of self regular IPMs.
- Design a hybrid algorithm that always uses the classical Newton search directions, unless it is necessary to use the self regular search directions to bring the iterate closer to central path.
- Analyze the infeasible variant of the algorithm.
- Analyze the theoretical behavior of the second order Mehrotra-type algorithm.

Finally, due to the rapid developments of IPMs to other classes of op-

timization problems, such as *Semidefinite and Second Order Conic Optimization*, it is interesting to extend our results to these classes of problems which is quite a challenging task due to the technical difficulties that may arise in the analysis of the algorithms.

Chapter 10

Appendix

In this appendix first we present some technical results that are referred to in the thesis. After that we describe the *self-dual embedding model* that is one of the two fundamental approaches in IPMs based software packages. We also review the stopping criteria used by IPMs based software packages. A popular heuristic is discussed that most infeasible IPM solvers are using to find a reasonably good starting point. Finally, we briefly describe how one computes the Newton direction in computational practice [50].

10.1 Some Technical Results

In this section we present some technical results that are used in the proof of some results in the thesis.

Theorem 10.1.1 *Let S be a nonempty, closed convex set in $\mathcal{R}^n$ and $y \notin S$. Then, there exists a unique vector $\bar{z} \in S$ with minimum distance from y. Furthermore, $\bar{z}$ is the minimizing point if and only if $(y - \bar{z})^T(z - \bar{z}) \leq 0$ for all $z \in S$.*

Proof: See [9]. □

Theorem 10.1.2 *Let S be a nonempty closed convex set in $\mathcal{R}^n$ and $y \notin S$. Then, there exists a nonzero vector u and a scalar η such that $u^T y > \eta$ and $u^T z \leq \eta$ for each $z \in S$.*

Proof: By Theorem 10.1.1 there exist a unique minimizing point $\bar{z}$ such that $(y - \bar{z})^T (z - \bar{z}) \leq 0$ for each $z \in S$. Letting $u = y - \bar{z} \neq 0$ and $\eta = \bar{z}^T (y - \bar{z}) = u^T \bar{z}$, we get $u^T z \leq \eta$ for each $z \in S$, while $u^T y - \eta = \|y - \bar{z}\|^2 > 0$. This completes the proof of the theorem. □

Lemma 10.1.1 *Suppose that $\alpha \in [0, 1]$. Then*

$$(1 + t)^\alpha \leq 1 + \alpha t, \quad \forall t \geq -1.$$

Proof: See [42]. □

Lemma 10.1.2 *Suppose that $h(t) : \mathcal{R} \to \mathcal{R}$ is a twice differentiable convex function with*

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

Moreover, suppose $h(t)$ attains its global minimum at $t^* > 0$ and $h^{''}(t)$ is an increasing function of t. Then for any $t \in [0, t^*]$

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

Proof: See [42]. □

10.2 The Self-Dual Embedding Model

In order to be able to apply feasible primal-dual IPMs one needs a strictly feasible starting point. However, there are LO problems that are infeasible or do not have strictly feasible (interior) solution. Even if a strictly feasible solution exists, it is not an easy task to find one. Thus, it is an important and interesting topic to develop techniques that alleviate these difficulties. The self-dual embedding model [2, 50, 65] is well studied and theoretically the most appealing approach. The idea is to build a slightly larger problem that has nonempty interior and a feasible starting point is available with no extra cost. Moreover, a strictly complementary solution of the self-dual model provides a strictly complementary optimal solution for the LO problem.

10.2.1 Self-Duality

A self-dual LO problem has the property that the primal and the dual problems are the same. Obviously, this property can hold only if the problems have some very special structure. A self-dual LO problem has the following general form:

$$\begin{aligned} \min \quad & d^T u + c^T w \\ & M_{11}u + M_{12}w \geq -d \\ & -M_{12}^T u + M_{22}w = -c \\ & u \geq 0, \ w \text{ free} \end{aligned} \tag{10.1}$$

where $d \in R^p$, $c \in R^q, M_{12} \in R^{p\times q}$ and finally $M_{11} \in R^{p\times p}$ and $M_{22} \in R^{q\times q}$ are skew-symmetric matrices. One can easily check that the dual of (10.1) is itself. The following proposition follows from self-duality.

Proposition 10.2.1 *If the self-dual LO problem (10.1) is feasible, then its optimal solution is zero.*

Proof: See [50, 63]. □

By the Goldman-Tucker Theorem 2.2.1 one has the following fundamental result for the self-dual model as well.

Proposition 10.2.2 *If the self-dual LO problem (10.1) is feasible, then it has a strictly complementary solution.*

Proof: See [50, 63]. □

10.2.2 Linear Feasibility Model

Now let us formulate the problems (P) and (D) as a single self-dual LO problem. To do so, let us first introduce the so called *Homogenous Linear Feasibility* (HLF) model which originally was introduced by Goldman and Tucker [17]:

$$\begin{array}{rrrl} Ax & -b\zeta & & = 0 \\ -A^T y & & +c\zeta & \geq 0 \\ b^T y & -c^T x & & \geq 0 \\ & & (\zeta, x) \geq 0, \ y \text{ free.} & \end{array} \tag{10.2}$$

Now, using the HFL model, let us consider the following LO problem:

$$\begin{array}{rrrl} \min & 0 & & \\ Ax & -b\zeta & & = 0 \\ -A^T y & & +c\zeta & \geq 0 \\ b^T y & -c^T x & & \geq 0 \end{array} \tag{10.3}$$

$$(\zeta, x) \geq 0, \ y \text{ free.}$$

The following proposition discuss some properties of (10.3) LO.

Proposition 10.2.3 *The LO problem (10.3) has the following properties:*

- *It is a self-dual problem.*
- *It has a trivial feasible solution.*
- *It does not satisfy the IPC.*

Proof: The first and second statements are obvious, then we focus on the last one. If a strictly feasible point $(\bar{\zeta}, \bar{x}, \bar{y})$ would exist, it would satisfy

$$b^T\bar{y} > c^T\bar{x}, \ A^T\bar{y} < \bar{\zeta}c, \quad A\bar{x} = b\bar{\zeta}, \ (\bar{\zeta}, \bar{x}) > 0. \tag{10.4}$$

Hence, the point (x, y, s) defined by

$$(x, y, s) = \left(\frac{\bar{x}}{\bar{\zeta}}, \frac{\bar{y}}{\bar{\zeta}}, c - A^T\frac{\bar{y}}{\bar{\zeta}}\right)$$

would be a strictly feasible primal-dual point for problems (P) and (D), for which

$$0 < x^Ts = c^Tx - b^Ty = \frac{c^T\bar{x} - b^T\bar{y}}{\bar{\zeta}} < 0$$

would hold, where the last inequality follows from (10.4). This is a contradiction, thus (10.3) does not admit a strictly feasible solution. □

The previous discussion shows that although problem (10.3) is self-dual, it does not fit in the class of problems that can be handled by feasible IPMs. Therefore, one has to further modify problem (10.3) in order to obtain

a problem with a nonempty interior. The new problem which can be obtained from (10.3) by adding an extra column and an extra row is

$$\begin{array}{rrrrcl} & & \min & \beta\nu & & \\ & Ax & -b\zeta & -r_P\nu & = & 0 \\ -A^Ty & & +c\zeta & -r_D\nu & = & 0 \\ b^Ty & -c^Tx & & -r_G\nu & \geq & 0 \\ {r_P}^Ty & +{r_D}^Tx & +{r_G}^T\zeta & & \geq & -\beta \\ & x \geq 0,\ \zeta \geq 0,\ \nu \geq 0,\ y \text{ free}, & & & & \end{array}$$

where

$$\begin{aligned} r_P &= \frac{Ax^0 - b\zeta^0}{\nu^0}, & r_D &= \frac{-A^Ty^0 + c\zeta^0 - e}{\nu^0} \\ r_G &= \frac{b^Ty^0 - c^Tx^0 - 1}{\nu^0}, & \beta &= -r_P^Ty^0 - r_D^Tx^0 - r_G\zeta^0, \end{aligned} \tag{10.5}$$

and (x^0, y^0, s^0) are arbitrary vectors for which $(x^0, s^0) > 0$. One can easily check that (10.5) is self dual and as we mentioned it has strictly feasible solution. Therefore, it fits in the class of problems that can be handled by feasible IPMs for LO.

The following proposition gives the information about the original LO problems that can be obtained by solving the self-dual model.

Proposition 10.2.4 *Let $(\zeta^*, \nu^*, x^*, y^*)$ be a strictly complementary optimal solution of problem (10.5). Then we have:*

- *If $\zeta^* > 0$, the $\frac{x^*}{\zeta^*}$, $\frac{y^*}{\zeta^*}$ are strictly complementary optimal solutions for problems (P) and (D), respectively.*
- *If $\zeta^* = 0$, the following cases may occur:*

- *If $c^T x^* < 0$, then (D) is infeasible.*
- *If $b^T y^* > 0$, then (P) is infeasible.*
- *If $c^T x < 0$ and $b^T y > 0$, then both (P) and (D) are infeasible.*

Proof: See [50, 63]. □

Now let us list some of the advantages of using the self-dual model [50, 72].

- It solves the LO problem (P) without any assumption on existence of a solution, or nonempty interior.
- It detects primal and/or dual infeasibility of the original problems.
- The user is free to choose any initial point (x^0, y^0, s^0) with $(x^0, s^0) > 0$, being feasible or infeasible.
- Any feasible IPM can be applied to solve the self-dual model and solution produced by the chosen IPM provide either a strictly complementary solution or an infeasibility certificate for the original problems (P) and (D).
- It preserves the $\mathcal{O}\left(\sqrt{n}\log\frac{n}{\epsilon}\right)$ worst case iteration complexity for the small-update methods and $\mathcal{O}\left(n\log\frac{n}{\epsilon}\right)$ iteration complexity for large-update methods.

10.3 Starting Point in Infeasible IPMs

In Theorem 6.1 of [63] the global convergence of infeasible IPMs from any starting point (x^0, y^0, s^0) with $(x^0, s^0) > 0$ is proved. However, the analysis of

infeasible algorithm suggests that the starting point, should be well centered, so that the componentwise product $x_i s_i,\ i = 1 \cdots, n$ are similar. Secondly, the starting point should not be "too infeasible;" that is, the ratio $\frac{(r_b^0, r_c^0)}{\mu_g^0}$ of infeasibility to the duality gap should not be too large. Third, the starting point has to be sufficiently far away from the origin. Computational experience also supports these two conditions. The point (4.17) satisfies these criteria. It is perfectly centered, and the ratio $\frac{(r_b^0, r_c^0)}{\mu_g^0}$ is bounded from above.

In practice however, usually heuristics are used. Now we present a popular heuristic that has been used in many state of the art software packages.

A popular heuristic [35] for finding a proper starting point include first to calculate solutions of the following two least square problems that we denote by $(\bar{x}, \bar{y}, \bar{s})$.

$$\min_x \|x\|^2 \quad \text{s.t.} \quad Ax = b,$$
$$\min_{y,s} \|s\|^2 \quad \text{s.t.} \quad A^T y + s = c.$$

It is obvious that for the solutions of these two least square problems, the residuals are zero for both the primal and the dual problems. However, the solutions might not be necessarily positive, what is required by IPMs. Therefore, in order to reach positivity of the starting point, it is defined as

$$(x^0, y^0, s^0) = (\bar{x}, \bar{y}, \bar{s}) + (\bar{\delta_x}, 0, \bar{\delta_s})e,$$

where

$$\bar{\delta_x} = \delta x + \frac{(x + \delta_x e)^T (s + \delta_s e)}{2\sum_{i=1}^n (s_i + \delta_s)},$$
$$\bar{\delta_s} = \delta_s + \frac{(x + \delta_x e)^T (s + \delta_s e)}{2\sum_{i=1}^n (x_i + \delta_x)},$$

and $\delta_x = \max(-1.5\bar{x}_{\min}, 0)$ and $\delta_s = \max(-1.5\bar{s}_{\min}, 0)$. One can easily show that $(x^0, s^0) > 0$. It is worth mentioning that the cost of solving the least

squares problems is of the same order as the cost of one iteration of the interior point algorithms, namely $\mathcal{O}(n^3)$.

Prior information is often available about the solution of an LO problem, in the form of the solution of a slightly different problem, for example a perturbation in data, change of price/cost or change of supply or demand. Primal-dual methods can use this information to construct "hot" starting points, which often yield faster convergence than "cold" starting points. However, up to now there is no unified way of handling this situation and consequently most packages do not include this feature, except for some specially structured problems that are included in some packages [20]. Unfortunately, at the time this thesis is written "hot" start in IPMs is not as efficient as it for simplex methods.

10.4 Stopping Criteria

Unlike the simplex method, primal-dual algorithms never find an exact solution of LO problem. Therefore, termination criteria must be applied to decide when the current iterate is close enough to the solution set. However, in certain applications like integer programming, one requires an optimal basis of the given problem. To do so, first an optimal partition of the variables has to be determined that can be done in finite number of iterations [50]. Then, using the strongly polynomial algorithm of Megiddo [32] one can construct optimal bases for both primal and dual problems.

Most primal-dual solvers report an approximate optimal solution for which the residuals and duality gap are sufficiently small. A typical test for accepting the point (x, y, s) as an approximate optimal solution consists of

checking the following three conditions:

$$\frac{\|r_b\|}{1+\|b\|} = \frac{\|Ax-b\|}{1+\|b\|} \leq \epsilon,$$

$$\frac{\|r_c\|}{1+\|c\|} = \frac{\left\|A^T y + s - c\right\|}{1+\|c\|} \leq \epsilon,$$

$$\frac{\left|c^T x - b^T y\right|}{1+|c^T x|} \leq \epsilon,$$

where ϵ is a small positive number, usually 10^{-8}.

10.5 Newton System

In this section we describe how the Newton system in most IPMs based software packages is solved. For more details see [4, 50, 63, 72]. Our discussion is for the infeasible case which contains the feasible case by assuming residuals equal to zero. One can do analogous discussion for the self-dual model which can be found in [50, 72]. As we saw in Chapter 2 the following system of equation has to be solved in IPMs based algorithms:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_b \\ -r_c \\ -r_{xs} \end{bmatrix}, \tag{10.6}$$

where $r_b = Ax - b,\ r_c = A^T y + s - c$ and r_{xs} can be different for different proximity measures. By eliminating Δs from (10.6) and using the notation $D = S^{-\frac{1}{2}} X^{\frac{1}{2}}$, we obtain the following form, often known as the *augmented system*:

$$\begin{bmatrix} A & 0 \\ -D^{-2} & A^T \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -r_b \\ -r_c + X^{-1} r_{xs} \end{bmatrix}. \tag{10.7a}$$

Thus Δs can be calculated as

$$\Delta s = -X^{-1}(r_{xs} + S\Delta x). \tag{10.7b}$$

Finally, by eliminating Δx from (10.7a) we have:

$$AD^2A^T\Delta y = -r_b + A(-S^{-1}Xr_c + S^{-1}r_{xs}). \tag{10.8a}$$

After solving (10.8a), Δs and Δx can be calculated by

$$\Delta s = -r_c - A^T\Delta y, \tag{10.8b}$$

$$\Delta x = -S^{-1}(r_{xs} + X\Delta s). \tag{10.8c}$$

Equation system (10.8a) is called the *normal equation* form because, in case of $r_b = 0$, the equation (10.8a) is the normal equation for the least square problem with coefficient matrix DA^T.

The normal equation approach is used by several primal-dual codes because the coefficient matrix in (10.8a) is symmetric and positive definite, which can be factorized by sparse Cholesky factorization. On the other hand, the coefficient matrix in (10.7a) is symmetric but indefinite, and thus algorithms for factorizing it are more complicated (for example one can use the Bunch-Parlett factorization) [18]. Various techniques have been developed for how to deal with dense columns in the coefficient matrix and how to handle numerical difficulties that naturally arise during factorization [21, 31, 70, 73].

Bibliography

[1] F. Alizadeh. *Combinatorial Optimization with interior point methods and semidefinite matrices.* Ph.D. thesis, University of Minnesota, 1991.

[2] E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, eds., High Performance Optimization, pp. 197–232. Kluwer Academic Publishers, 2000.

[3] E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, Ser. B, 95(2), pp. 249–277, 2003.

[4] E.D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programming. In: T. Terlaky, ed., *Interior Point Methods of Mathematical Programming*, pp. 189–252. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[5] K.M. Anstreicher and R.A. Bosch. A new infinity-norm path following algorithm for linear programming. *SIAM Journal on Optimization*, 5(2), pp. 236–246, 1995.

[6] Y.Q. Bai, M. El Ghami, and C. Roos. A new effiecient large-update primal-dual interior-point method based on a finite barrier. *SIAM Journal on Optimization,* 13(3), pp. 766–782, 2003.

[7] Y.Q. Bai, M. EL Ghami, and C. Roos. A comparative study of new barrier functions for primal-dual interior-point algorithms in linear optimization. *SIAM Journal on Optimization,* 15(1), pp. 101–128, 2004.

[8] D.A. Bayer and J.C. Lagarias. Karmarkar's linear programming algorithm and Newton's method. *Mathematical Programming*, 50, pp. 291–330, 1991.

[9] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons, Chichester, UK, 1993.

[10] L. Behjat, A. Vannelli, and W. Rosehart. Integer Linear programming models for global routing. *Informs Journal on computing,* 18(2),2006.

[11] A. Ben-Tal, A. Nemirovski. *Lectures on Modern Convex Optimization.* AMS-SIAM Series on Optimization, SIAM, Philadelphia, 2002.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[13] CPLEX: ILOG Optimization. `http:// www.ilog.com.`

[14] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, NJ, 1963.

[15] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Wiley, New York, 1968, reprinted by SIAM Publication, 1990.

[16] K.R. Firsch. The logarithmic potential method for solving linear programming problems. Memorandum, University institue of Economics, Oslo, 1955.

[17] A.J. Goldman and A.W. Tucker. Thoery of linear programming. In: H.W. Kuhn and A.W. Tucker, eds., Linear Inequalities and Related Systems, Annals of Mathematical Studies, Princeton, New Jersey, No. 38, 53–97, 1956.

[18] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore and London, second ed., 1989.

[19] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications,* 6, pp. 137–156, 1996.

[20] J. Gondzio. Warm start of the primal-dual method applied in the cutting plane scheme. *Mathematical Programming*, 83(1), pp. 125–143, 1998.

[21] J. Gondzio. HOPDM (version 2.12) - A fast LP solver based on a primal-dual interior point method. *European Journal of Operational Research,* 85, pp. 221–225, 1995.

[22] C.C. Gonzaga. Complexity of predictor-corrector algorithms for LCP based on a large neighborhood of the central path. *SIAM Journal on Optimization,* 10, pp. 183–194, 1999.

[23] P. Hung and Y. Ye. An asymptotically $O(\sqrt{n}L)$-iteration path-following linear programming algorithm that uses long steps. *SIAM Journal on Optimization*, 6, pp. 570–586, 1996.

[24] F. Jarre and M. Wechs. Extending Mehrotra's corrector for linear programs.*Advanced Modeling and Optimization*, 1(2), pp. 38–60, 1999.

[25] L.G. Khachian. Polynomial-time algorithm for linear programming (in Russian). *Doklady Academiia Nauk SSR*, 224, pp. 1039–1096. English Translation: Soviet Mathematics Doklady, 20, pp. 191–194, 1979.

[26] N.K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4, pp. 373–395, 1984.

[27] V. Klee and G.J. Minty. How good is the simplex algorithm? In: O. Shisha, ed., *Inequalities III*, Academic Press, pp. 159–175, 1972.

[28] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior-point algorithm for linear programming. In: N. Megiddo ed.,*Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods*, Springer Verlage, Berling, pp. 29–47, 1989.

[29] K. Krishnan. *Linear Programming Approach to Semidefinite Programming Problems.* Ph.D. Thesis, Mathematical Sciences, Rensselaer Polytechnique, Troy, NY 12180, July 2002.

[30] I.J. Lustig. Feasiblity issues in a primal-dual interior-point methods for linear programming. *Mathematical Programming*, 49, pp. 145–162, 1990/91.

[31] I.J. Lustig, R.E. Marsten, and D.F. Shanno. Interior point methods for linear programming problems: Computational state of the art. *ORSA Journal on Computing,* 6, pp. 1–14, 1994.

[32] N. Megiddo. On finding primal and dual optimal bases. *ORSA Journal on Computing,* 3, pp. 63–65, 1991.

[33] N. Megiddo. Pathways to the optimal set in linear programming. In: N. Megiddo ed., *Progress in Mathematical Programming: Interior Point and Related Methods*, pp. 131–158. Springer Verlag, New York, 1989. Identical version in: *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, pp. 1–35, 1986.

[34] S. Mehrotra. On finding a vertex solution using inteior-point methods. *Linear Algebra and its Applications*, 152, pp. 233–253, 1991.

[35] S. Mehrotra. On the implementation of a (primal-dual) interior point method. *SIAM Journal on Optimization*, 2, pp. 575–601, 1992.

[36] S. Mehrotra and Z. Li. Convergence conditions and krylov subspace-based corrections for primal-dual interior point method. *SIAM Journal on Optimization*, 15, pp. 635–653, 2005.

[37] R.D.C. Monteiro, I. Adler, and M.G.C. Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extensions. *Mathematics of Operations Research*, 15, pp. 191–214, 1990.

[38] S. Mizuno, M.J. Todd, and Y. Ye. On adaptive step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18, pp. 964–981, 1993.

[39] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* newblock John Wiley & Sons, Chichester, UK, 1999.

[40] NETLIB:A Repository of Linear Programming Test Problems. `http://www-fp.mcs.anl.gov/otc/Guide/TestProblems/LPtest`.

[41] J. Peng, C. Roos, and T. Terlaky. *New Design and Analysis of Interior Point Methods.* Ph.D. Thesis, Delft University of Technology, The Netherland, 2001.

[42] J. Peng, C. Roos, and T. Terlaky. *Self-Regularity: A New Paradigm for Primal-Dual Interior-Point Algorithms.* Princeton University Press, Princeton, NJ, 2002.

[43] J. Peng, C. Roos, and T. Terlaky. Self-regular functions and new search directions for linear and semidefinite optimization. *Mathematical Programming*, 93, pp. 129–171, 2002.

[44] J. Peng and T. Terlaky. A dynamic large-update primal-dual interior-point method for linear optimization. *Optimization Methods and Software*, 17, pp. 1077–1104, 2002.

[45] J. Peng, T. Terlaky and Y.B. Zhao. A predictor-corrector algorithm for linear optimization based on a specific self-regular proximity function. *SIAM Journal on Optimization,* 15(4), pp. 1105–1127, 2005.

[46] J. Peng and Y. Xia. A cutting plane algorithm for the minimum-sum-of-squared error clustering. Proceeding of SIAM Conference on Data Mining, eds., H. Kargupta, J. Srivastava, C. Kamath, and A. Goodman, 2005.

[47] F.A. Potra. The Mizuno-Todd-Ye algorithm in a large neighborhood of central path. *European Journal of Operational Research,* 143, pp. 257–267, 2002.

[48] F.A. Potra. A superlinearly convergent predictor-corrector method for degenrate LCP in a wide neighborhood of the central path. *Mathematical Programming,* 100(2), pp. 317–337, 2004.

[49] F.A. Potra and X. Liu. Predictor-corrector methods for sufficient linear complementarity problems in a wide neighborhood of the central path. *Optimization Methods and Software,* 20(1), pp. 145–168, 2005.

[50] C. Roos, T. Terlaky, and J.-Ph.Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach,* John Wiley & Sons, Chichester, UK, 1997. Second ed., Springer Science, 2005.

[51] M. Salahi, T. Terlaky, and G. Zhang. The complexity of self-regular proximity based infeasible IPMs. *Technical Report* 2003/3*, Advanced Optimization Lab. Department of Computing and Software, McMaster University. Computational Optimization and Applications,* DOI: 10.1007/s10589-005-3064-1, 2005.

[52] M. Salahi and T. Terlaky. An adaptive self-regular proximity based large-update interior point method for linear optimization. *Optimization Methods and Software,* 20(1), pp. 169–185, 2005.

[53] M. Salahi, J. Peng, and T. Terlaky. On Mehrotra-Type Predictor-Corrector Algorithms. *Technical Report* 2005/4*, Advanced Optimization Lab. Department of Computing and Software, McMaster University.* `http://www.cas.mcmaster.ca/~oplab/publication.` Submitted to *SIAM Journal on Optimization,* May 2005.

[54] M. Salahi and T. Terlaky. Adaptive large neighborhood self-regular predictor-corrector IPMs for LO. *Technical Report* 2004/7*, Advanced Op-*

timization Lab. Department of Computing and Software, McMaster University, http://www.cas.mcmaster.ca/∼oplab/publication. To appear in *Journal of Optimization Theory and Applications,* 130(1), 2006.

[55] M. Salahi, M.R. Peyghami, and T. Terlaky. New complexity analysis of IIPMs for linear optimization based on a specific self-regular function. *Technical Report* 2005/10*, Advanced Optimization Lab. Department of Computing and Software, McMaster University.* http://www.cas.mcmaster.ca/∼oplab/publication. Submitted to *European journal of Operational Research,* June 2005.

[56] M. Salahi and T. Terlaky. Postponing the choice of the barrier parameter in Mehrotra-tpye predictor-corrector algorithms. *Technical Report* 2005/16*, Advanced Optimization Lab. Department of Computing and Software, McMaster University.* http://www.cas.mcmaster.ca/∼oplab/publication. Submitted to *European journal of Operational Research,* September 2005.

[57] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In: A. Prékopa, J. Szelezsán, and B. Strazicky, eds., *System Modeling and Optimization: Proceeding of the 12th IFIP Conference*, Budapest, Hungary, September 1985, Volume 84, Lecture Notes in Control and Information Sciences, pp. 866–876. Springer Verlag, Berlin 1986.

[58] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11/12, pp.

625–653. Special issue on Interior Point Methods (CD supplement with software), 1999.

[59] T. Terlaky, A. Vannelli, and H. Zhang. On routing in VLSI design and comunication networks. To appear in Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005), LNCS, Sanya, China.

[60] T. Terlaky and S. Zhang. Pivot rules for linear programming, a survey. *Analls of Operations Research*, 46, pp. 203–233, 1993.

[61] R.J. Vanderbei. LOQO: An interior-point code for qudaratic programming. *Optimization Methods and Software,* 12, pp. 451–488, 1999.

[62] H. Wolkowicz, R. Saigal, and L. Vandenberghe *Handbook of Semidefinite Programming: Theory, Algorithm and Applications.* Kluwer academic publisher, 2000.

[63] S.J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, USA, 1997.

[64] XpressMP: Dash Optimization. `http://www.dashoptimization.com.`

[65] X. Xu, P. Hung, and Y. Ye. A simplified homogenous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research,* 50, pp. 151–172, 1996.

[66] Y. Ye. *Interior Point Algorithms, Theory and Analysis.* John Wiley & Sons, Chichester, UK, 1997.

[67] Y. Ye. On the finite convergence of inteior-point algorithms for linear programming. *Mathematical Programming*, 57, pp. 325–336, 1992.

[68] Y. Ye and K.M. Anstreicher. On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for LCP. *Mathematical Programming,* 62, pp. 537–551, 1993.

[69] Y. Yuan (ed.). *Advances in Nonlinear Programming.* Kluwer Academic Publisher, 1998.

[70] Y. Zhang. Solving large-scale linear programms by inteior point methods under the MATLAB enviroment. *Optimization Methods and Software,* 10, pp. 1–31, 1999.

[71] Y. Zhang and D. Zhang. On the polynomiality of the Mehrotra-type predictor-corrector interior point algorithms. *Mathematical Programming*, 68, pp. 303–317, 1995.

[72] X. Zhu. *Implementing the New Self-Regular Proximity Based IPMs.* M.Sc. Thesis, Advanced Optimization Laboratory, Department of Computing and Software, McMaster University, Hamilton, Canada, 2002. `http://www.cas.mcmaster.ca/~oplab/publication.`

[73] X. Zhu, J. Peng, T. Terlaky, and G. Zhang. On implementing self-regular proximity based feasible IPMs. *Technical Report* 2003/2*, Advanced Optimization Lab. Department of Computing and Software, McMaster University,* `http://www.cas.mcmaster.ca/~oplab/publication.`