Large-Update Infeasible Interior-Point Methods for Linear Optimization

Large-Update Infeasible Interior-Point Methods for Linear Optimization

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 29 augustus 2011 om
12.30 uur

door

Alireza ASADI

Master of Science in Applied Mathematics
Tabriz University, Tabriz, Iran
geboren te Marand, East Azerbijan, Iran.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. C. Roos

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. C. Roos	Technische Universiteit Delft, promotor
Prof. dr. K. I. Aardal	Technische Universiteit Delft
Prof. dr. ir. D. den Hertog	Universiteit van Tilburg
Prof. dr. Y. Bai	Shanghai University
Prof. dr. ir. C. Vuik	Technische Universiteit Delft
Prof. dr. ir. F. van Keulen	Technische Universiteit Delft
Dr. M. Dür	Rijksuniversiteit Groningen

NWO
Netherlands Organisation for Scientific Research

This research was financially supported by the Netherlands Organization for Scientific Research (NWO) under the project number 613.000.441.

Copyright © 2011 by Alireza Asadi

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without the prior permission of the author.

ISBN 978-90-79982-12-7

Typeset by the author with the LaTeX Documentation System.
Author email: A.Asadi@tudelft.nl or a.r.asadi@gmail.com

To my mother and
the memory of my father
and my love Parisa Zahedi

Acknowledgements

This thesis is the outcome of my research during my PhD in TU Delft for which I am in debt of many people for making it possible. First of all, I am much obliged to my advisor, professor Kees Roos for his abundant help and invaluable assistance. Many thanks go to the members of my committee, professors Karen Aardal, Dick den Hertog, Yanquin Bai, Kees Vuik, Fred van Keulen and dr. Mirjam Dür, who braved the storm of the century to attend my defense. Their advice and patience is highly appreciated.

I would like to dedicate my special thank to dr. H. Mansouri and dr. M. Zangiabadi for their precious discussions and all of those things that I can not put them in words but they both know what I mean!. I am also thankful dr. I. Ivanov for providing his kernel function-based interior-point code at the beginning of my PhD.

Many thanks go to my master thesis supervisor, prof. K. Mirnia who opened my eyes to the world of interior-point methods, and without whom my PhD in Delft would not be started. I am also thankful dr. A. Ghaffari Hadigheh, dr. J. Mehri, dr. M. Mehdizadeh Khalsarayee and ir. M. Ghaffari for their direct and/or indirect influences on my way to Delft.

Many acknowledgments also go out to the technical support staff of our department, Paulo Anita, Stephen van der Laan and Munire van der Kruyk, the secretary of our department, Conny van Driel-Vogelesang, Rina Abbriata and Esther van Seters, and the CICAT organization of TU Delft, especially Veronique van der Varst, for their earnest support during these years. I also thank my lovely friends who made my life cheerful in Delft.

My wholehearted gratitude goes to my family for their love and support during my remoteness. My mother who suffered a lot for my not being with her, and my father who passed away while I was away and I could not even give him a right thank for all he gave me in my life. This thesis is dedicated to his memory.

Last but not least, I would like to thank my wife Parisa Zahedi for her love and compassion.

Alireza Asadi
Delft, August 2011

Contents

List of Tables

List of Figures

List of Algorithms

1

Introduction

1.1 Linear optimization

This thesis deals with linear optimization (LO), which amounts to the problem of minimizing or maximizing a linear function subject to some linear constraints. LO is a branch of *Mathematical Programming* which in turn is a part of *Operations Research*. Economically speaking, an optimization problem is a formalized version of the economic principle, i.e., depending on the viewpoint of the decision maker, it either maximizes the output for some given input (e.g., profit maximization), or minimizes the input for some required output (e.g., cost minimization). Nowadays, LO has a wide range of practical applications. In [28], a list of a variety of those applications is provided among which are The Diet Problem, Allocation Problem, Cutting Stock Problem, Crew Scheduling and Data Envelopment Analysis.

Neglecting the primitive works on the solution of a system of linear inequalities, done by such people as J. Fourier [49] and J. Farkas [48], the modern concept of LO problem traces back to L. V. Kantorovich in 1939[1]. As a consultant for the Laboratory of the Plywood Trust, Kantorovich dealt with the problem of distributing some initial raw materials in order to maximize equipment productivity under certain restrictions. His studies were interrupted by World War II (WWII) during which his results remained unknown[2]. Postwar, in 1947, this problem was studied also by some other people like G. B. Dantzig and T. C. Koopmans.

[1] Here we mention only a few of the highlights which directly influenced the field of LO. For a comprehensive history of operations research and LO, we refer to [70] and [34]. The personal reminiscences of Dantzig [19] are also interesting. A collection of personal reminiscences of contributors in the field of mathematical programming can be found in [59].

[2] Nobel Lectures, Economics 1969-1980, Editor Assar Lindbeck, World Scientific Publishing Co., Singapore, 1992. One may also find an autobiography of Kantorovich at `http://nobelprize.org/nobel_prizes/economics/laureates/1975/kantorovich-autobio.html`.

Dantzig introduced his well-known *simplex method* for solution of LO problem. We refer to, e.g., [20] for an extensive description of the simplex method.

Associated with any LO problem there is another LO problem, called its *dual* problem. In [19, pages 45-46] it is mentioned that existence of the dual problem was conjectured first by John von Neumann, also in 1947 (during a conversation with Dantzig)[3]. A rigorous proof of the *duality theorem* was published later by Gale, Kuhn and Tucker in 1948.

In an LO problem, the constraints may be equalities and/or inequalities. There may be some variables constrained as nonnegative and some unconstrained. However, any LO problem can be transformed into the so-called *standard form* which is the LO problem with only equality constraints and nonnegative variables. Therefore, most literature on LO, e.g., [60, 98, 103, 115, 118], deals with the standard form which is defined as follows:

$$\text{(P)} \qquad \min\left\{c^T x : Ax = b, \quad x \geq 0\right\},$$

where $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. Without loss of generality, it can be assumed that A has full row rank. The dual problem, associated with (P), is given by:

$$\text{(D)} \qquad \max\left\{b^T y : A^T y + s = c, \quad s \geq 0\right\},$$

with $y \in \mathbb{R}^m$. After introduction of the dual problem, Dantzig named the problem (P) *primal* problem.

In this thesis, we deal with the standard form of the LO problem and, unless otherwise stated, 'LO problem' stands for the 'standard LO problem'.

The *feasible regions* of (P) and (D) are denoted by $\mathcal{P}$ and $\mathcal{D}$, respectively:

$$\begin{aligned}\mathcal{P} &:= \{x : Ax = b,\ \ x \geq 0\}\\ \mathcal{D} &:= \left\{(y,s) : A^T y + s = c,\ \ s \geq 0\right\}.\end{aligned}$$

The problem (P) is called *feasible* if $\mathcal{P}$ is nonempty and otherwise *infeasible*. If $c^T x$ is unbounded below over $\mathcal{P}$, we call (P) unbounded, otherwise bounded. We use similar terminology for the dual problem (D).

The relative interiors of $\mathcal{P}$ and $\mathcal{D}$ are denoted by $\mathcal{P}^\circ$ and $\mathcal{D}^\circ$, respectively:

$$\begin{aligned}\mathcal{P}^\circ &:= \{x : Ax = b,\ \ x > 0\}\\ \mathcal{D}^\circ &:= \left\{(y,s) : A^T y + s = c,\ \ s > 0\right\}.\end{aligned}$$

We say that (P) and (D) satisfy the *interior-point condition* (IPC) if both $\mathcal{P}^\circ$ and $\mathcal{D}^\circ$ are nonempty.

[3] The theory which Neumann suggested for the dual problem was analogue to his theory for *Game Theory*.

1.2 Duality results

We recall the well-known weak duality result for LO problem.

Proposition 1.2.1. (Weak Duality)(cf. [98, Proposition II.1]) *Let x and (y, s) be feasible for, respectively, (P) and (D). Then $c^T x - b^T y = x^T s \geq 0$. Consequently, $c^T x$ is an upper bound for the optimal value of (D), if it exists, and $b^T y$ is a lower bound for the optimal value of (P), if it exists. Moreover, if the* duality gap *$x^T s$ is zero then x is an optimal solution of (P) and (y, s) is an optimal solution of (D).*

It can be concluded from Proposition 1.2.1 that if one of the problems (P) and (D) is unbounded then the other is infeasible. As we mentioned in Section 1.1, the duality theory was introduced by van Neumann [83], and later explicitly formulated and proven by Gale, Kuhn and Tucker [33]. The classical duality theorem for LO can be stated as follows.

Theorem 1.2.2. (Strong Duality)(cf. [98, Theorem II.2]) *If (P) and (D) are feasible then both problems have optimal solutions. Then, if $x \in \mathcal{P}$ and $(y, s) \in \mathcal{D}$, these are optimal solutions if and only if $x^T s = 0$. Otherwise neither of the two problems has optimal solutions: either both (P) and (D) are infeasible or one of the two problems is infeasible and the other one is unbounded.*

Below we state another duality theory result for LO which is due to Goldman and Tucker [41].

Theorem 1.2.3. (Goldman-Tucker Theorem)(cf. [98, Theorem II.3]) *If (P) and (D) are feasible then there exists a* strictly complementary pair *of optimal solutions, that is an optimal triple (x, y, s), with $x \in \mathcal{P}$ and $(y, s) \in \mathcal{D}$, satisfying $x + s > 0$.*

Recall that by Theorem 1.2.2, a primal-dual feasible triple (x, y, s) is optimal if and only if $x^T s = 0$. This is called the *complementarity condition* for (P) and (D) (see e.g., [98]). Because the vectors x and s are nonnegative, the complementarity condition is equivalent to $xs = 0$[4]. In short, any primal-dual optimal solution (x^*, y^*, s^*) of (P) and (D) satisfies the following conditions:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ x \geq 0,\ s &\geq 0, \\ xs &= 0, \end{aligned} \tag{1.1}$$

[4] Throughout this thesis, we denote by 0 and $\mathbf{e}$ (used later) the zero and the all-one vectors, respectively, of appropriate size. Moreover, if $x, s \in \mathbb{R}^n$, then xs denotes the componentwise (or Hadamard) product of the vectors x and s.

where the first three lines require that $x \in \mathcal{P}$ and $(y, s) \in \mathcal{D}$ and the last line is the complementarity condition. The system (1.1) is known as the *Karush-Kuhn-Tucker (KKT)* optimality conditions for LO.

1.3 Algorithms

In order to solve LO problem, various methods have been introduced. In this section we briefly discuss the most important methods, thereby focusing on interior-point methods because these are the methods studied in this thesis.

1.3.1 Simplex method

As mentioned in Section 1.1, the simplex method was introduced by Dantzig in 1947. It starts from a vertex of the feasible region, which is actually a polyhedron, and moves along an edge to a vertex with non-increasing (for a minimization problem) values of the objective function; this is repeated until an optimal vertex is reached. In unbounded problems, some feasible ray is detected during the simplex procedure along which the objective function is decreasing. Despite its nice practical performance, there is an example, given by Klee and Minty [53], having $2n$ inequality constraints and n variables for which the simplex method needs 2^n iterations. This means that the simplex method may not have a polynomial worst-case iteration bound.

Indeed, for many variants of the simplex method, according to their different pivoting rules, exponential running time examples have been found.

1.3.2 Ellipsoid method

A polynomial-time algorithm for LO remained unknown until Khachiyan [52] introduced his *ellipsoid method* in 1979. The ellipsoid method generates a sequence of ellipsoids enclosing an optimal solution, if any exists, whose volumes uniformly decrease at every step. If there is no optimal solution, the method stops when the ellipsoid is so small that it can be established that no optimal solution exists. The iteration bound of the ellipsoid method is $O(n^2L)$ with L denotes the length of input data bits. However, the ellipsoid method turned out to be too slow for practical purposes (see e.g., [15, 40]) and the simplex method remained the favorite method in practice.

1.3.3 Interior-point methods

Another polynomial algorithm for LO was presented by Karmarkar [51] in 1984. The iteration bound of Karmarkar's algorithm is better than that of Khachiyan's algorithm by a factor $O(n)$. Although some initiatory implementations of Karmarkar's algorithm were disappointing (see e.g., [61, 109]), some authors, like

Adler et al. [1], Monma and Morton [79] and McShane et al. [71], implemented variants of Karmarkar's algorithm which favorably competed with the simplex method in practice. Karmarkar's algorithm is within the class of *interior-point methods* (IPMs). In contrast with the simplex method, IPMs move through the interior of the feasible region to find the optimal solution[5].

Although, IPMs have been known since 1960 in the form of *barrier methods* [30], they received renewed attention after Karmarkar's result. This has led to the following categories of algorithms:

- *projective methods*, as proposed by Karmarkar [51] and studied by others in [4, 5, 22, 29, 31, 35, 43, 84, 94, 107, 109, 116],
- *affine-scaling methods*, as proposed by Dikin [26], and investigated further in [1, 13, 27, 73, 79, 81, 110, 113, 114],
- *path-following methods (PFMs)*, which can be divided in *small-update* algorithms, as studied in [10, 39, 42, 56, 80, 92, 95, 96, 100, 108, 111], *large-update algorithms*, as studied in [10, 13, 24, 25, 99], and *predictor-corrector (PC)* methods, as studied in [74, 77],
- *potential-reduction methods (PRMs)*, as described in [7, 32, 44, 57]. See also [6, 106] for a survey on potential-reduction methods.

Karmarkar's iteration bound, namely $O(nL)$, was improved a couple of years later by Renegar [92] by a factor of $\sqrt{n}$.

IPMs have shown their efficiency in solving LO problem in both practice and theory. For a survey of IPMs, we refer to e.g. [23, 45, 98, 115, 118].

IPMs are divided into *feasible* IPMs (FIPMs) and *infeasible* IPMs (IIPMs). FIPMs start from a primal-dual strictly feasible triple (x^0, y^0, s^0), i.e., $x^0 \in \mathcal{P}^\circ$ and $(y^0, s^0) \in \mathcal{D}^\circ$, and generate a sequence of strictly feasible triples (x, y, s) converging to an optimal solution of (P) and (D). In contrast, in IIPMs the iterates are not feasible, and apart from reaching optimality one needs to strive for feasibility. Precisely speaking, IIPMs start from a triple (x^0, y^0, s^0), where $x^0 > 0$ and $s^0 > 0$, and generate triples (x, y, s) satisfying $x > 0$ and $s > 0$ but not necessarily (1.2a) and (1.2b). IIPMs attempt to obtain feasibility and optimality simultaneously.

The current chapter deals with FIPMs. IIPMs are extensively considered in Chapter 2. Because projective methods, PRMs and affine-scaling methods are beyond the scope of the thesis we do not explain them in detail. We concentrate

[5] It is worth mentioning that, nowadays, both simplex-based algorithms and primal-dual IPMs are used in commercial packages as SeDuMi, COIN, CPLEX, MOSEK and LINPROG (in MATLAB). In LINPROG, the default algorithm is based on the simplex method. If the problem is labeled as "Large-Scale“, the well-known LIPSOL package is used which is the best known IPM-based software for solving LO problems.

on PFMs. To be more specific, we deal in this chapter with the feasible *full-Newton step* PFM which was introduced and discussed in [98]. We are interested in this algorithm because the full-Newton step IIPM of Roos [97], to be discussed in Chapter 3, is inspired by this algorithm. The main result of this thesis, i.e., the algorithm, explained in Chapters 5 and 6, is actually a large-update variant of the above mentioned full-Newton step IIPM.

PFMs use a virtual path inside the feasible region of (P) and (D) as a guideline to an optimal solution of (P) and (D). The next section is devoted to the definition of this so-called *central path*.

1.4 Central path

Most IPMs consider the *parameterized* KKT system, defined as follows:

$$\begin{aligned} Ax &= b, && (1.2\text{a})\\ A^T y + s &= c, && (1.2\text{b})\\ x \geq 0,\ s &\geq 0, && (1.2\text{c})\\ xs &= \mu \mathbf{e}, && (1.2\text{d}) \end{aligned}$$

where $\mu > 0$ is a positive parameter which is called the *barrier parameter* (see e.g., [55, 56, 80]). The system (1.2) is called the KKT system with respect to μ. Because of (1.2d), any solution of the system (1.2) will satisfy $x > 0$ and $s > 0$. Therefore, a solution exists only if (P) and (D) satisfy the IPC. Surprisingly, it has been shown (see [98, Theorem II.4 or Remark II.5]) that if the IPC holds then, for any $\mu > 0$, the system (1.2) has a (unique) solution. It follows that (P) and (D) satisfy the IPC if and only if the system (1.2) has a unique solution for any $\mu > 0$. This unique solution is denoted by $(x(\mu), y(\mu), s(\mu))$. The vector $x(\mu)$ is called the *μ-center* of (P) and $(y(\mu), s(\mu))$, the μ-center of (D). The set of the primal-dual μ-centers $(x(\mu), y(\mu), s(\mu))$, as μ runs through $\mathbb{R}_{++}$, is called the central path of (P) and (D). Megiddo [72] established that as μ tends to zero, the central path converges to a primal-dual optimal solution[6].

Unfortunately, the system (1.2) is nonlinear because of the equation (1.2d), which makes obtaining the μ-center rather difficult. IPMs overcome this issue by using a numerical iterative procedure based on the well-known Newton-Raphson's method. This is the subject of the next section.

[6] The notion of *analytic center* of a bounded convex set was introduced by Sonnevend [102]. If the feasible region of an LO problem is bounded, then its analytic center is the limit of the central path if μ tends to infinity. On the other hand, if the optimal set of an LO problem is bounded, then its analytic center is the limiting point of the central path as μ tends to zero [98].

1.5 Search directions

Given a triple (x, y, s) and some $\mu > 0$, to obtain the μ-center, we need displacements $(\Delta x, \Delta y, \Delta s)$ such that $x + \Delta x$, $y + \Delta y$ and $s + \Delta s$ coincide with the μ-center of (P) and (D):

$$\begin{aligned} A(x+\Delta x) &= b, \\ A^T(y+\Delta y) + (s+\Delta s) &= c, \\ x+\Delta x > 0,\ s+\Delta s &> 0, \\ (x+\Delta x)(s+\Delta s) &= \mu \mathbf{e}. \end{aligned}$$

Defining the primal and dual *residual vectors* r_b and r_c as

$$r_b := b - Ax \quad \text{and} \quad r_c := c - A^T y - s, \tag{1.3}$$

and ignoring the inequalities for the moment, the last system can be rewritten as follows:

$$\begin{aligned} A\Delta x &= r_b, \\ A^T\Delta y + \Delta s &= r_c, \\ s\Delta x + x\Delta s + \Delta x\Delta s &= \mu\mathbf{e} - xs. \end{aligned} \tag{1.4}$$

By neglecting the quadratic term $\Delta x \Delta s$ from the third equation, according to Newton's iterative method for solving nonlinear systems, one obtains the following system in Δx, Δy and Δs:

$$\begin{aligned} A\Delta x &= r_b, \\ A^T\Delta y + \Delta s &= r_c, \\ s\Delta x + x\Delta s &= \mu\mathbf{e} - xs. \end{aligned} \tag{1.5}$$

The directions $(\Delta x, \Delta y, \Delta s)$, given by (1.5), are called the primal-dual Newton directions at the triple (x, y, s). Because A has full row rank, it can be verified (see [98, Theorem II.42]) that the primal-dual Newton step $(\Delta x, \Delta y, \Delta s)$ is uniquely defined by the system (1.5).

In FIPMs, i.e., if $x \in \mathcal{P}^\circ$ and $(y, s) \in \mathcal{D}^\circ$, then one has $r_b = r_c = 0$. As a result, the system (1.5) is reduced as follows:

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

By the first two equations of the system (1.6), it can be easily verified that the vectors x^+, y^+ and s^+, given by

$$\begin{aligned} x^+ &:= x + \Delta x, \\ y^+ &:= y + \Delta y, \\ s^+ &:= s + \Delta s, \end{aligned} \tag{1.7}$$

satisfy the equations (1.2a) and (1.2b). However, they may not satisfy the inequality conditions (1.2c). Hence, some step size $\alpha \in (0, 1]$ may be required such that

$$\begin{aligned} x^+ &:= x + \alpha\Delta x \in \mathcal{P}^\circ, \\ y^+ &:= y + \alpha\Delta y \in \mathcal{D}^\circ, \\ s^+ &:= s + \alpha\Delta s \in \mathcal{D}^\circ. \end{aligned}$$

As we mentioned above, the quadratic term $\Delta x \Delta s$ is neglected from the third equation of the system (1.4). This causes the equation (1.2d) to be not satisfied, i.e., the pairwise products $x_i^+ s_i^+$, $i = 1, \cdots, n$ may not be equal to μ, except in the ideal case that $\Delta x \Delta s = 0$. This means that PFMs follow the central path approximately and do not stay exactly on the central path. The next section is devoted to definition of the so-called *proximity measure* which is used to measure the deviation of the iterates from the μ-centers.

1.6 Proximity measure

The proximity measure which we introduce in this section was first used by Jansen et al. [50] and later, with some minor modification, by Roos, Terlaky and Vial [98].

Given the iterates x and s and some $\mu > 0$, the *variance vector* v of the iterates x and s with respect to μ is defined as follows:

$$v := \sqrt{\frac{xs}{\mu}}. \tag{1.8}$$

Note that

$$v = \mathbf{e} \Leftrightarrow xs = \mu\mathbf{e},$$

which means that the variance vector is the all-one vector if and only if the iterates x and (y, s) are the μ-centers.

Using v, the proximity measure $\delta(v)$ is defined as follows:

$$\delta(v) := \frac{1}{2}\|v^{-1} - v\|. \tag{1.9}$$

It can be easily verified that $\delta(v) = 0$ if and only if $v = \mathbf{e}$, which means that $\delta(v)$ vanishes only at the μ-center. In other words, one has

$$\delta(v) = 0 \Leftrightarrow v = \mathbf{e} \Leftrightarrow xs = \mu\mathbf{e}. \tag{1.10}$$

In [98], the proximity measure $\delta(v)$ was motivated as follows. Defining *the scaled Newton directions* d_x and d_s as

$$d_x := \frac{v\Delta x}{x} \quad \text{and} \quad d_s := \frac{v\Delta s}{s},$$

the system (1.6) can be rewritten as below:

$$\begin{aligned} \bar{A}d_x &= 0, \\ \bar{A}^T d_y + d_s &= 0, \\ d_x + d_s &= v^{-1} - v, \end{aligned} \tag{1.11}$$

with $d_y := \frac{\Delta y}{\sqrt{\mu}}$ and $\bar{A} := AV^{-1}X$ where $V = \operatorname{diag}(v)$ and $X = \operatorname{diag}(x)$.

Because the vector d_x belongs to the null space of $\bar{A}$ and d_s to the row space of $\bar{A}$, the vectors d_x and d_s are orthogonal, i.e., $(d_x)^T d_s = 0$. This implies that

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

Due to this, the vectors d_x and d_s are zero if and only if

$$\|v^{-1} - v\| = 0,$$

which is the case only at the μ-center, according to (1.10).

The quantity $\|v^{-1} - v\|$ seems to be a natural tool for measuring 'distance' between the iterates and the μ-center. The idea of using this quantity is due to Jansen et al. [50]. However, Roos et al. [98] used this quantity divided by 2, i.e., $\delta(v)$, given by (1.9)[7].

[7] In many papers on IPMs, e.g., [55, 80], one does not measure proximity to the μ-center, but to the central path. A popular way is to use the following expression:

$$\left\| \frac{xs}{\mu_g} - \mathbf{e} \right\|, \tag{1.12}$$

where $\|.\|$ is some norm and μ_g is the average value of the iterates, i.e.,

$$\mu_g := \frac{x^T s}{n}. \tag{1.13}$$

Recall that on the central path the pairwise products $x_i s_i$ for $i = 1, \cdots, n$ are identical and their common value is their average value μ_g:

$$xs = \mu\mathbf{e} \Rightarrow \mu = \frac{x^T s}{n} = \mu_g.$$

In the literature, e.g., [115] and the references therein, both the 2-norm and the ∞-norm were used. The 2-norm variant was introduced by Kojima, Mizuno and Yoshise [55] and also used by Monteiro and Adler [80] to derive primal-dual algorithms for LCP and LO problem, respectively.

In order to express the proximity measure in terms of the iterates and the barrier parameter μ, the following notation will also be used:

$$\delta(x, s; \mu) := \delta(v).$$

We now proceed with explaining the feasible full-Newton step PFM which was introduced and discussed in [98].

1.7 A feasible full-Newton step PFM

It is convenient to start with a formal description of the algorithm, as below.

Algorithm 1.1 A full-Newton step PFM

```
Input:
    an accuracy parameter $\varepsilon > 0$;
    a barrier update parameter $\theta \in (0, 1)$.
begin
    $x = s = \mathbf{e}$, $y := 0$, and $\mu := 1$;
    while $n\mu \geq \varepsilon$,
        Newton step:
        $(x, y, s) := (x, y, s) + (\Delta x, \Delta y, \Delta s)$;
        $\mu$-update:
        $\mu := (1 - \theta)\mu$;
    endwhile
end
```

Assume that a primal $x^0 \in \mathcal{P}^\circ$ and a dual pair $(y^0, s^0) \in \mathcal{D}^\circ$ and a $\mu^0 > 0$ are given such that $x^0 s^0 = \mu^0 \mathbf{e}$. Without loss of generality, it can be assumed that[8]

$$x^0 = s^0 = \mathbf{e}, \; y^0 = 0 \quad \text{and} \quad \mu^0 = \frac{\left(x^0\right)^T s^0}{n} = 1. \tag{1.14}$$

Moreover, let a barrier-update parameter $\theta \in (0, 1)$ be given. In Subsection 1.7.2 we discuss how to obtain θ. At the beginning of an iteration of the algorithm the system (1.6) is employed to compute Newton steps $(\Delta x, \Delta y, \Delta s)$ and then the new iterates (x^+, y^+, s^+) are calculated using (1.7). After that the barrier parameter μ is reduced to $\mu^+ := (1 - \theta)\mu$. This process is repeated until the duality gap is less than or equal to a prescribed tolerance ε.

[8] This can be realized by embedding the given (P) and (D) into a homogenous self-dual problem.

Because Δx and Δs are orthogonal[9], i.e., $\Delta^T x \Delta s = 0$, then, using the third equation of (1.6), after a full Newton step the duality gap is computed as follows:

$$(x+\Delta x)^T(s+\Delta s) = x^T s + (x^T\Delta s + s^T \Delta x) + \Delta^T x \Delta s = \\ x^T s + \mu \mathbf{e}^T \mathbf{e} - x^T s = n\mu. \quad (1.15)$$

This means that after each iteration of the algorithm the duality gap is identical to $n\mu$, and hence that each iteration reduces the duality gap by a factor $1-\theta$.

The difficult part of the analysis of the algorithm is the analysis of the Newton step. In the next subsection, we recall a condition under which the Newton step is (strictly) feasible and quadratically convergent.

1.7.1 Properties of the Newton step

We start this subsection with the following lemma.

Lemma 1.7.1. (cf. [98, Theorem II.50]) *Denote $\delta := \delta(x,s;\mu)$. If $\delta \leq 1$ then after each full Newton step the new iterates (x^+, y^+, s^+) are feasible, i.e., $x^+ \in \mathcal{P}$ and $(y^+, s^+) \in \mathcal{D}$. Moreover, if $\delta < 1$ then $x^+ \in \mathcal{P}^\circ$ and $(y^+, s^+) \in \mathcal{D}^\circ$ and* [10]

$$\delta(x^+, s^+, \mu) \leq \frac{\delta^2}{\sqrt{2(1-\delta^2)}}. \quad (1.17)$$

Lemma 1.7.1 defines a certain neighborhood of the μ-center where the full Newton step $(\Delta x, \Delta y, \Delta s)$, obtained from (1.6), certainly yields a primal-dual strictly feasible point.

By slightly narrowing the neighborhood of the μ-center, one gets a region where the Newton's method is quadratically convergent: if $\delta \leq 1/\sqrt{2}$ then it can be verified that

$$\delta(x^+, s^+, \mu) \leq \delta^2. \quad (1.18)$$

This means that, given a $\mu > 0$, if the current triple (x, y, s) satisfies $\delta(x, s; \mu) \leq 1/\sqrt{2}$, then Newton's method quadratically converges to the μ-center. The set of primal-dual (feasible) pairs (x, s) satisfying $\delta(x, s; \mu) \leq 1/\sqrt{2}$ is called the *quadratically convergent region* of the μ-center.

The algorithm generates a sequence of strictly feasible pairs (x, s) in the quadratically convergent region of the μ-centers. This means that the parameter θ should be such that after reducing μ to $\mu^+ := (1-\theta)\mu$, the current pair (x, s) lies in the quadratically convergent region of the μ^+-center. In the next subsection we explain how to derive such a θ.

[9] Δx and Δs belong to the null space and the row space of the matrix A, respectively.

[10] We would like to mention that a slightly sharper than (1.17) was proven in [98, Theorem II.52] which says:

$$\delta(x^+, s^+, \mu) \leq \frac{\delta^2}{\sqrt{2(1-\delta^4)}}. \quad (1.16)$$

Since this result has no impact on the order of the convergence rate of the algorithm, for the sake of simplicity, we present an analysis which is based on (1.17).

1.7.2 Value of the barrier-updating parameter

Given a primal-dual feasible pair (x, s) and $\mu > 0$, satisfying $\delta(x, s; \mu) \leq 1/\sqrt{2}$, (1.18) implies that after a (full) Newton step, the new iterates (x^+, y^+, s^+) satisfy

$$\delta(x^+, s^+; \mu) \leq \frac{1}{2}. \tag{1.19}$$

The aim is to derive the barrier-updating parameter θ such that $\delta(x^+, s^+; \mu^+) \leq 1/\sqrt{2}$. We first recall the following lemma which investigates the influence of a μ-update on δ.

Lemma 1.7.2. (cf. [98, Lemma II.54]) *Given a primal-dual pair (x, s) and $\mu > 0$ such that $x^T s = n\mu$, one has*

$$\delta(x, s; \mu^+)^2 = (1 - \theta)\delta(x, s; \mu)^2 + \frac{\theta^2 n}{4(1-\theta)}, \quad \textit{with} \quad \mu^+ := (1-\theta)\mu.$$

By Lemma 1.7.2 and using (1.19), one gets

$$\delta(x^+, s^+; \mu^+)^2 \leq \frac{1-\theta}{4} + \frac{\theta^2 n}{4(1-\theta)}.$$

Using this, $\delta(x^+, s^+; \mu^+) \leq 1/\sqrt{2}$ certainly holds only if

$$\frac{1-\theta}{4} + \frac{\theta^2 n}{4(1-\theta)} \leq \frac{1}{2}.$$

It can be verified that this inequality holds for all θ satisfying

$$0 < \theta \leq \frac{1}{\sqrt{n+1}}.$$

In the sequel, the barrier-updating parameter θ is taken as follows:

$$\theta = \frac{1}{\sqrt{n+1}}.$$

In the next subsection we explain how to estimate the number of iterations of the algorithm.

1.7.3 Iteration bound

An iteration of Algorithm 1.1 consists of one Newton step plus a μ-update. After a Newton step, the duality gap is identical to $n\mu$, according to (1.15). On the other hand, the parameter μ is reduced by a factor $1 - \theta$ per iteration. The algorithm stops if the duality gap is less than or equal to a prescribed tolerance $\varepsilon > 0$. A natural way to estimate the number of iterations is to count the number of the μ-updates before $n\mu < \varepsilon$ is satisfied. We recall the following lemma.

Lemma 1.7.3. (cf. [98, Lemma II.17]) *If the barrier parameter μ has the initial value μ^0 and is repeatedly multiplied by $1-\theta$, with $0<\theta<1$, then after at most*

$$\left\lceil \frac{1}{\theta}\log\frac{n\mu^0}{\varepsilon}\right\rceil$$

iterations we obtain $n\mu \le \varepsilon$.

Using $\theta = \frac{1}{\sqrt{n+1}}$, Lemma 1.7.3 implies that the total number of iterations of the algorithm is bounded above by

$$\sqrt{n+1}\log\frac{n}{\varepsilon}.$$

1.8 Motivation

Recently, an infeasible PFM for LO has been introduced by Roos [97]. This algorithm uses a virtual path outside the feasible region of (P) and (D), namely the *homotopy path* (See Chapter 4), as a guideline to an optimal solution. The algorithm can be considered as a generalization of the full-Newton step PFM presented in Section 1.7 to the case where the starting point is infeasible. Therefore, in addition to the optimality, Roos' algorithm strives for the feasibility of the iterates. In this algorithm, as usual, the duality gap[11] is measured by x^Ts and the primal and the dual infeasibility by the size of the residual vectors r_b and r_c, respectively. The algorithm is designed in such a way that a full-Newton step reduces the sizes of the residual vectors with the same speed as the duality gap. Precisely speaking, after a full-Newton step, the quantity $\epsilon(x,y,s)$, defined as

$$\epsilon(x,y,s) := \max\left\{\|r_b\|, \|r_c\|, x^Ts\right\}, \tag{1.20}$$

is reduced by a factor $1-\theta$ where $\theta\in(0,1)$ is the barrier-updating parameter. Analogue to its feasible counterpart, the algorithm is a so-called *small-update* approach, in the sense that the barrier-updating parameter θ is inversely proportional to the problem dimension n. Precisely speaking, one has[12] $\theta = O(1/n)$. Using this θ, Roos establishes that in order to obtain an ε-solution for a prescribed $\varepsilon>0$, i.e., a triple (x,y,s) satisfying

$$\epsilon(x,y,s) \le \varepsilon, \tag{1.21}$$

at most

$$16\sqrt{2}n\log\frac{\epsilon(x^0,y^0,s^0)}{\varepsilon}$$

[11] It should be mentioned that, in an IIPM, in general the iterates x and s are not feasible for the original primal-dual pair of problems, and hence the quantity x^Ts can not be called the duality gap at x and s (with respect to the original primal-dual pair of problems). However, as it will be mentioned in Chapter 2, x and s are always feasible for a *perturbed pair* of problems and with respect to this pair x^Ts is the duality gap at x and s. Therefore, when dealing with IIPMs, the quantity x^Ts can still be called the duality gap at x and s.

[12] Note that in the full-Newton FIPM, presented in Section 1.7, we have $\theta = O(1/\sqrt{n})$.

iterations of the algorithm are required. This iteration bound coincides with the best known iteration bound ever obtained for IIPMs, which is due to Mizuno [75][13].

Despite their nice theoretical iteration bound and the feature of using full Newton steps, small-update methods have the disadvantage that they are too slow in practice. PFMs that use a fixed barrier-updating parameter $\theta \in (0, 1)$, independent of n, e.g., $\theta = \frac{1}{2}$, turned out to be more efficient in practice. These methods are known as so-called large-update methods. In large-update methods, full Newton steps may not be feasible[14] ($x + \Delta x$ and $s + \Delta s$ might have negative components). Therefore, one has to use *damped* Newton steps. The step size is calculated using a line search with respect to some *barrier function* (see e.g., [98]). Precisely speaking, starting from a point $z = (x, y, s)$, after updating the barrier parameter μ to $\mu^+ := (1 - \theta)\mu$ with e.g., $\theta = \frac{1}{2}$, in searching for the μ^+-center $z(\mu^+) := (x(\mu^+), y(\mu^+), s(\mu^+))$, a finite number of (feasible) points $\{z^k\}_{k=1}^{K}$ are generated where K is such that z^K is a *good* approximation of $z(\mu^+)$ (with respect to a barrier function). Unfortunately, regardless of their nice practical behavior, large-update methods have worse theoretical iteration bounds than small-update methods. This phenomenon, i.e., IPMs with nice theoretical properties are inefficient in practice and the other way around, has been called the *irony* of IPMs [93, page 51].

1.9 Outline

Based on the aforementioned motivation, we designed a class of infeasible PFMs. Our IIPMs can be considered as large-update variants of Roos' full-Newton IIPM. However, they differ from feasible large-update PFMs, in the sense that θ is no longer arbitrary; it has to be computed at each iteration. In practice, our algorithm has the advantage that the parameter θ is larger than $O(1/n)$, even $\theta = O(1)$, which yields a larger amount of reduction on the quantity $\epsilon(x, y, s)$ at a so-called *outer* iteration of the algorithm. However, the above mentioned irony of IPMs is still present in our algorithm, meaning that despite its nice practical behavior, it has worse theoretical iteration complexity than its full-Newton step counterpart. For a variant, we obtain the bound

$$O\left(n\sqrt{n}(\log n)^3 \log \frac{\epsilon(x^0, y^0, s^0)}{\varepsilon}\right),$$

which is a factor $\sqrt{n}(\log n)^3$ worse than the iteration bound of Roos' full-Newton step IIPM, presented in [97]. The best-known iteration bound for large-update

[13] We would like to mention that a simplified version of Roos' algorithm was given by Mansouri and Roos [66], to be followed by a slightly improved version which was given by Gu, Mansouri, Zangiabadi, Bai and Roos [46]. Both versions, i.e., those presented in [46] and [66], have the convergence rate $O(n)$.

[14] Throughout this thesis, we call displacements Δx, Δy and Δs (strictly) feasible if the triple $(x + \Delta x, y + \Delta y, s + \Delta s)$ is primal-dual (strictly) feasible.

IIPMs is

$$O\left(n\sqrt{n}(\log n)\log\frac{\epsilon(x^0,y^0,s^0)}{\varepsilon}\right), \tag{1.22}$$

which is due to Salahi, Peyghami and Terlaky [101]. Note that the iteration complexity of our algorithm is a factor $(\log n)^2$ worse than (1.22).

Before dealing with the new algorithm, in Chapter 2, we first survey theoretical properties of some IIPMs presented by several authors within the last two decades, with the hope of getting some clue which might be helpful in improving the iteration complexity of our algorithm. We studied global convergence and polynomiality of the IIPMs, starting from Lustig's algorithm [62], to the infeasible *potential-reduction* methods of Mizuno, Kojima and Todd [76].

Our algorithm is inspired by a slightly improved version of Roos' algorithm which was given by Gu et al. [46]. We devote Chapter 3 to a description of this algorithm as a preparation to our large-update algorithm. As we mentioned above, Roos' algorithm and our algorithm share the property that they approximately follow the homotopy path to find an optimal solution. In Chapter 4 we introduce the notion of the homotopy path and argue that if (P) and (D) are both feasible, then the homotopy path converges to an optimal solution of these problems. Chapter 4 is based on [8]. The main result, namely a class of large-update IIPMs for LO, is presented in Chapters 5 and 6. Chapter 5 deals exclusively with the theoretical properties of our algorithms which amount to obtaining a default barrier-updating parameter θ and estimating the total number of iterations. Unfortunately, the outcome of Chapter 5 is disappointing as the best convergence rate of a variant is $O(n\sqrt{n}(\log n)^3)$. This is because we were not able to get rid of n in the expression of θ. One might ask then why we call our algorithm a large-update algorithm. To justify this we rely on our numerical test. We ran the algorithm to solve a subset of the NETLIB problems and compared the iteration numbers with those of the well-known LIPSOL package, the best existing software for the solution of LO problems. The iteration numbers of our algorithm seem promising; the outcome was in favor of LIPSOL, though. We obtain substantial larger values of θ than its default (theoretical) value, even $\theta = O(1)$, during our implementations. Moreover, we often observe that $\theta = 1$ within a few iterations after the start of the algorithm, which means that we have reached feasibility. Obviously, one then may proceed with a feasible large-update approach. Details regarding the implementation of the algorithm can be found in Chapter 6. Chapters 5 and 6 are based on [9]. We offer some concluding remarks and topics for further research in Chapter 7.

2

The state-of-the-art in IIPMs

2.1 Introduction

As we made clear in Chapter 1, FIPMs assume that some strictly feasible point is at hand and generate a sequence of strictly feasible points converging to an optimal solution. In real-life problems it more often happens that no strictly feasible point is known a priori; moreover, the problem may not be feasible at all. Therefore, most existing practical algorithms allow positive but infeasible starting points[1]. These algorithms are referred to as infeasible interior-point methods (IIPMs). The current chapter deals extensively with the state-of-the-art in the theory of IIPMs.

As we mentioned in Chapter 1, IIPMs generate a sequence of infeasible triples (x, y, s) with $(x, s) > 0$. As usual, the duality gap is measured by $x^T s$ and the primal and the dual infeasibility by, respectively, $\|r_b\|$ and $\|r_c\|$ where r_b and r_c are given by (1.3). The infeasibility and the duality gap are decreased at about the same rate, i.e., the quantity $\epsilon(x, y, s)$, given by (1.20), is monotonically decreasing. For a prescribed $\varepsilon > 0$, a triple (x, y, s) is an ε-solution of (P) and (D) if (1.21) is satisfied.

Starting from initials (x^0, y^0, s^0) with $x^0 > 0$ and $s^0 > 0$, most IIPMs for LO, e.g., those studied by Lustig [62], Kojima, Megiddo and Mizuno [54], Mizuno, Kojima and Todd [76], Mizuno [75], Potra [91] and Roos [97], use, implicitly or explicitly, the following system to solve the pair (P) and (D):

$$b - Ax = \nu r_b^0, \quad x \geq 0, \tag{2.1a}$$

$$c - A^T y - s = \nu r_c^0, \quad s \geq 0, \tag{2.1b}$$

$$xs = \mu \mathbf{e}, \tag{2.1c}$$

[1] E.g., LIPSOL package of Zhang [120], PCx package of Czyzyk et al. [18], LOQO package of Vanderbei [112] and etc., all of which are available on the web site of Network Enabled Optimization Server (NEOS Server: `http://neos.mcs.anl.gov/neos/solvers/index.html`).

where $\mu > 0$, $\nu \in (0,1]$, and r_b^0 and r_c^0 are, respectively, the primal and the dual initial residual vectors. It can be noticed from the system (2.1) that the parameters ν and μ control the feasibility and the optimality, respectively. For $\mu = 0$ and $\nu = 0$, the system coincides with the KKT system (1.1). Now consider the *perturbed problem* (P_ν) (see e.g., [78, 97]), defined as

$$(\mathrm{P}_\nu) \qquad \min\left\{ \left(c - \nu r_c^0\right)^T x : \ Ax = b - \nu r_b^0, \ x \geq 0 \right\},$$

and its dual (D_ν), given by

$$(\mathrm{D}_\nu) \qquad \max\left\{ \left(b - \nu r_b^0\right)^T y : \ A^T y + s = c - \nu r_c^0, \ s \geq 0 \right\}.$$

It can be easily verified that the system (2.1) is the parameterized KKT system, with respect to μ, of the perturbed pair (P_ν) and (D_ν). Because triple (x^0, y^0, s^0) is strictly feasible for the pair (P_ν) and (D_ν) if $\nu = 1$, then (P_1) and (D_1) satisfy the IPC. As a result, if $\nu = 1$, the system (2.1) has a unique solution for any $\mu > 0$. It has been proven (see e.g., [97, Lemma 3.1]) that (P) and (D) are feasible if and only if the perturbed pair (P_ν) and (D_ν) satisfies the IPC for any $\nu \in (0,1]$. It follows that (P) and (D) are feasible if and only if the system (2.1) has a unique solution for any $\nu \in (0,1)$ and $\mu > 0$. Denoting this unique solution by $(x(\mu,\nu), y(\mu,\nu), s(\mu,\nu))$, $x(\mu,\nu)$ is called the μ-center of (P_ν), and $(y(\mu,\nu), s(\mu,\nu))$ the μ-center of (D_ν). The set of these μ-centers of (P_ν) and (D_ν), for all $0 < \nu \leq 1$ and $\mu > 0$, form a 2-dimensional surface outside the feasible region of (P) and (D) which is called the *surface of centers* (see [78]). In order to improve the feasibility and the duality gap with the same speed, it is assumed that the ratio $\frac{\mu}{\nu}$ is a constant. Under this assumption, the above mentioned surface is reduced to a path of centers which is called the homotopy path of (P) and (D). Moreover, for the sake of notational simplicity, we denote the μ-centers $(x(\mu,\nu), y(\mu,\nu), s(\mu,\nu))$ by $(x(\nu), y(\nu), s(\nu))$. It is worth mentioning that existence of the homotopy path does not require (P) and (D) to satisfy the IPC. In Chapter 4 we establish that as $\mu \to 0$ (and $\nu \to 0$), the homotopy path converges to an optimal solution of (P) and (D).

2.2 A brief overview of IIPMs

In this section we briefly discuss some existing IIPMs that will be discussed more in detail in the subsequent sections. Table 2.1 shows the methods that we discuss. The origin of modern IIPMs is due to Lustig [62]. Lustig's algorithm was inspired by the "big M" method for the initialization of IPMs. The "big M" method for IPMs was introduced by Megiddo [72] and inspired by a well-known initialization method for the simplex method, introduced by Charnes, Cooper and Henderson [16]. In the "big M" method, some huge coefficients are introduced and they may cause severe numerical instabilities. The motivation of Lustig for designing his

Author	Contribution of the paper	Conv. rate	Ref.	Sec.
Lustig	primal-dual IIPM (1990/91)	–	[62]	2.4
Kojima Megiddo Mizuno	global convergence of IIPMs (1993)	–	[54]	2.5
Zhang	polynomiality of IIPMs (1994)	$O(n^2)$	[119]	2.6
Mizuno	improvement of the convergence rate of IIPM by a factor of $O(n)$	$O(n)$	[75]	2.7
Potra	PC IIPM (1996)	$O(n)$	[91]	2.8
Mizuno Kojima Todd	three potential-reduction IIPMs (1995)	Alg. I: $O(n^2\sqrt{n})$ Alg. II: $O(n^2\sqrt{n})$ Alg. III: $O(n)$	[76]	2.9

Table 2.1: Progress in IIPMs

algorithm was to overcome this issue. Lustig's achievement was elimination of the role of M in his computations by taking M infinitely large. We explain this in Section 2.4. To facilitate understanding of Lustig's method, we briefly describe Megiddo's "big M" method in Section 2.3.

Lustig implicitly solves the system (2.1) at an iteration. The Newton directions corresponding to this system are calculated from the following system:

$$\begin{aligned} A\Delta x &= \nu r_b^0, \\ A^T\Delta y + \Delta s &= \nu r_c^0, \\ s\Delta x + x\Delta s &= \mu\mathbf{e} - xs. \end{aligned} \tag{2.2}$$

By choosing some proper step size $\alpha \in (0, 1]$, the parameter[2] ν reduces by a factor $1 - \alpha$. Although, from this monotonic reduction of ν, one can expect convergence of the algorithm to a feasible point, but no theoretical proof was given concerning global convergence because it was difficult to deal with the following two situations:

(i) when the optimality occurs before the feasibility,

[2] It is worth mentioning that, Lustig's algorithm is a two-phase approach. Phase I cares more about the feasibility than the optimality. This means that the algorithm attempts to reduce the parameter ν while the parameter μ may be decreasing or increasing (the optimality may improve or worsen). Once a (strictly) feasible point has been found (ν is small enough or $\nu = 0$), the algorithm enters Phase II which is a FIPM.

(ii) when $\mathcal{P}^\circ = \emptyset$ and/or $\mathcal{D}^\circ = \emptyset$.

In the case (i), some infeasible iterates may be obtained at which the duality gap is zero, and in the case (ii), the parameter ν can not be set to zero at any iteration.

It is worth to mention that despite its drawback of having no theoretical convergence proof, practically efficient variants of Lustig's algorithm were given by several authors such as Lustig, Marsten and Shanno [63, 64] and Choi, Monma and Shanno [17].

Inspired by the primal-dual FIPM, studied in [56], Kojima, Megiddo and Mizuno [54] designed a variant of Lustig's algorithm which has a global convergence proof. In this variant, the Newton steps are obtained from the system (2.1) with the parameter μ set a priori to a fraction of μ_g, i.e., $\beta x^T s/n$ for some $\beta \in (0,1)$. The use of this μ yields some improvement of the duality gap at an iteration. Different step sizes $\alpha_P \in (0,1]$ and $\alpha_D \in (0,1]$ are used in the primal and the dual spaces, respectively. They are calculated in such a way that the following two properties are guaranteed: first, the feasibility improves with the same speed or slightly faster than the duality gap at an iteration of the algorithm[3], i.e., the iterates satisfy

$$\max\{\|r_b\|, \|r_c\|\} \leq \lambda x^T s \text{ for some } \lambda > 0; \tag{2.3}$$

this is useful to overcome the issue described in (i). Second, the iterates stay always away from the boundary (either outside or inside the feasible region of (P) and (D)) until the optimality is obtained. Precisely speaking, the iterates satisfy

$$xs \geq \gamma \frac{x^T s}{n} \mathbf{e}, \text{ for a } \gamma \in (0,1). \tag{2.4}$$

This is useful when $\mathcal{P}^\circ = \emptyset$ and/or $\mathcal{D}^\circ = \emptyset$, which concerns the issue described in (ii). In the FIPM, presented in [56], strictly feasible points are generated which satisfy (2.4). The set of strictly feasible points of a pair (P) and (D) satisfying (2.4) is referred to as the *infinity neighborhood* of the central path of (P) and (D) and denoted by $\mathcal{N}_{-\infty}(\gamma)$:

$$\mathcal{N}_{-\infty}(\gamma) := \left\{ (x,s) : x \in \mathcal{P},\, s \in \mathcal{D} \quad \& \quad xs \geq \gamma \frac{x^T s}{n}, \text{ with } \gamma \in (0,1) \right\}.$$

The IIPM presented in [54], generates iterates in the infinity neighborhood of the central path[4]. If the feasibility is satisfied, the algorithm reduces to the FIPM, presented in [56]. Kojima et al. [54] prove that the duality gap[5] is monotonically decreasing by a constant factor, whence global convergence of the algorithm

[3] Unlike Lustig's algorithm, the optimality and the feasibility both improve at an iteration of Kojima et al.'s algorithm [54].

[4] In the sequel of the current chapter, unless otherwise stated, by 'central path' we refer to the central path of either the original pair (P) and (D) or some perturbed pair (P$_\nu$) and (D$_\nu$).

[5] As we mentioned in Chapter 1, throughout this thesis, unless otherwise stated, by 'duality gap', we refer to the duality gap with respect to either the original pair (P) and (D) or to some perturbed pair (P$_\nu$) and (D$_\nu$).

follows. In the case of infeasibility, the algorithm detects a region where no primal-dual feasible solution is available. We explain Kojima et al.'s IIPM in Section 2.5.

The first polynomial-time IIPM was presented by Y. Zhang [119]. He considers a variant of the algorithm studied in [54] for solving the so-called *Horizontal Linear Complementarity Problem* (HLCP). The convergence proof of this algorithm is more or less the same as [54]. However, these algorithms are different in some minor aspects: in Zhang's algorithm, identical step sizes in a primal and the dual spaces are used, i.e., $\alpha_P = \alpha_D = \alpha$ with some $\alpha \in (0, 1]$; and moreover, the duality gap and the infeasibility are reduced with the same speed, i.e., $1 - \alpha$. It is established that the step size is inversely proportional to the initial residual norms. Hence, by introducing an initial point at which the residual norms are $O(n)$, Zhang obtains $\alpha = O(1/n^2)$ which gives rise to the convergence rate $O(n^2)$ for the algorithm[6]. We describe this algorithm for the case of LO (which is a special case of the HLCP) in Section 2.6.

Inspired by the algorithm of Zhang [119], Mizuno [75] presented a modification of the algorithm studied in [54] with $O(n^2)$ convergence rate. As in [54], this variant reduces the infeasibility slightly faster that the optimality. Later on, Mizuno realized that by further tightening the neighborhood of the central path, a larger fraction of the Newton steps could be used. Precisely speaking, by replacing the the neighborhood $\mathcal{N}_{-\infty}(\gamma)$ by $\mathcal{N}_2(\gamma)$, defined as

$$\mathcal{N}_2(\gamma) := \{(x, s) > 0 : \quad \|xs - \mu_g \mathbf{e}\| \le \gamma \mu_g\}, \text{ with a } \gamma \in (0, 1),$$

he established that the step size can improve to $\alpha = O(1/n)$. This yielded the convergence rate $O(n)$. We explain this algorithm in Section 2.7.

After the release of Mizuno's $O(n)$ IIPM [75], Potra [91] also published an $O(n)$ predictor-corrector IIPM[7]. Potra's algorithm uses the same neighborhood as the algorithm of Mizuno [75]. However, they are different in some aspects. The predictor step of Potra's method consists of two types of Newton steps rather than one. The first predictor step which is an affine-scaling step improves the duality gap preserving the current feasibility, and the second one improves the feasibility and has a tiny impact on the duality gap. As in Mizuno's PC algorithm, the outcome of the predictor steps is a point in the neighborhood $\mathcal{N}_2(2\gamma)$. The corrector step leaves the duality gap and the residual norms unchanged and brings the iterates generated during the predictor step to the neighborhood $\mathcal{N}_2(\gamma)$. An

[6] Roughly speaking, the polynomiality of the algorithm is obtained by using some narrower neighborhood than the one used in [54]. This can be justified as follows. To prove polynomiality one needs to set the step size to its lower bound, i.e., $O(1/n^2)$. Therefore, the neighborhood of the central path which covers the iterates becomes smaller than the one used by Kojima et al. [54].

[7] After the release of the first version of Mizuno's paper [75], Potra [90] introduced a predictor-corrector (PC) IIPM with the convergence rate $O(n\sqrt{n})$. Before Potra's result got to be published, Mizuno published the second version of the paper [75] which contained an $O(n)$ PC algorithm. After that, Potra realized that by a slight modification of his algorithm, he could prove the convergence rate $O(n)$.

iteration of the algorithm improves the feasibility and the duality gap with the same speed. We describe Potra's algorithm in Section 2.8.

So far, the algorithms we mentioned above are path-following methods. Precisely speaking, the generated iterates stay in a certain neighborhood of the homotopy path of (P) and (D) which consists of the μ-centers of the perturbed pairs (P_ν) and (D_ν) as the parameter ν and the barrier parameter μ tend to zero, simultaneously. It can be established (see e.g., [98, 115]) that these μ-centers are the minimizers of a barrier function which depends on the iterates and μ. Potential-reduction methods (PRMs) also use a barrier function which is called *potential function*. A potential function depends only on the iterates and has no minimizer. It is mainly used to determine the step size and an upper bound for the number of iterations. The step size is chosen such that the potential function decreases by some positive value per iteration. As the potential function approaches $-\infty$, the iterates converge to an optimal solution. We refer to [51, 57, 117] for some feasible PRMs. Mizuno, Kojima and Todd [76] presented an infeasible potential-reduction method for LO problem. They introduce three variants of such algorithms, namely Algorithm I, II and III. Algorithm I decreases the potential function of Tanabe-Todd-Ye [104, 108] under a condition which makes the duality gap to improve not faster than the feasibility. This condition has been already used in the polynomial-time IIPMs presented by Zhang [119] and Mizuno [75]. Algorithm I is called a constrained potential-reductrion IIPM. Mizuno et al. prove the convergence rate $O(n^2\sqrt{n})$ for this variant. Algorithm II decreases a new potential function which is obtained by embedding the constraint of Algorithm I into the Tanabe-Todd-Ye function. The iteration bound for variant II is the same as for variant I. Algorithm III is an $O(n)$ variant of Algorithm II. Unlike Mizuno's $O(n)$ PFM, Algorithm III does not confine the iterates to any neighborhood of the homotopy path. We refer to Section 2.9 for a detailed description of these algorithms.

2.3 The "big M" method in IPMs

Megiddo [72] was the first who applied the logarithmic barrier approach for simultaneous solution of the primal and the dual problem. He proposed to reformulate the problem using an artificial variable so that a starting point became available. To this end, a "big M" multiplier of an artificial variable was added to the objective function along with a new constraint with right-hand side M. This method was developed to a primal-dual algorithm by Kojima, Mizuno and Yoshise [56]. For a sufficiently large M, as in the simplex method, an optimal solution for the original pair can be obtained if and only if the optimal value of the artificial variable is zero. For arbitrary initials $x^0 > 0$, y^0 and $s^0 > 0$, the primal artificial problem (P_M) is defined as

$$\begin{array}{lllll} \min & c^T x & +Mx_{n+1} & & \\ \text{s.t.} & Ax & +r_b^0 x_{n+1} & & = b, \\ & -{r_c^0}^T x & & +x_{n+2} & = M, \\ & x, & x_{n+1}, & x_{n+2} & \geq 0, \end{array} \tag{P_M}$$

and its dual problem (D_M) is given by

$$\begin{array}{llllll} \max & b^T y & +My_{m+1} & & & \\ \text{s.t.} & A^T y & -r_c^0 y_{m+1} & +s & & = c, \\ & {r_b^0}^T y & & +s_{n+1} & & = M, \\ & & y_{m+1} & & +s_{n+2} & = 0, \\ & & & s, \quad s_{n+1}, & s_{n+2} & \geq 0, \end{array} \tag{D_M}$$

where x_{n+1} and x_{n+2} are primal artificial real variables, y_{m+1}, s_{n+1} and s_{n+2}, dual artificial variables, r_b^0 and r_c^0, the primal and dual initial residual vectors and M is a sufficiently large real number. The latter means that M is so large that the inequality

$$M > \max\{-{r_c^0}^T x^0, {r_b^0}^T y^0\}, \tag{2.5}$$

is satisfied, according to Kojima et al. [56].

It is worth noting that the pair (P_M) and (D_M) was also used later by Kojima, Mizuno and Yoshise [56] and Monteiro and Adler [80].

We recall from [56, Theorem 2.3] that if, in addition to the condition (2.5), M satisfies the condition

$$M > \max\{-{r_c^0}^T x^*, {r_b^0}^T y^*\},$$

where (x^*, y^*, s^*) is an optimal solution of (P) and (D), then one has

(a) A feasible solution (x, x_{n+1}, x_{n+2}) of (P_M) is optimal if and only if x is optimal solution of (P) and $x_{n+1} = 0$.

(b) A feasible solution (y, y_{m+1}) and (s, s_{n+1}, s_{n+2}) of (D_M) is optimal if and only if (y, s) is optimal solution of (D) and $y_{m+1} = 0$.

A nice feature of the pair (P_M) and (D_M) is that strictly feasible solutions are at hand. Taking

$$x = x^0,\ x_{n+1} = 1 \quad \text{and} \quad x_{n+2} = M + {r_c^0}^T x^0, \tag{2.6}$$

we have a primal strictly feasible solution for (P_M), and taking

$$s = s^0,\ s_{n+1} = M - {r_b^0}^T y^0 \quad \text{and} \quad s_{n+2} = 1, \tag{2.7}$$

we have dual strictly feasible solutions for (D_M).

An implementation of the algorithm of Kojima et al. [56] was first presented by McShane, Monma and Shanno [71] which proved to be favorably comparable with the past implementations like that of Monma and Morton [79] or MINOS[8] [9] [10], a software based on the simplex method designed by Murtagh and Saunders [82].

2.4 Lustig's algorithm

Lustig [62] aims at solving the pair (P) and (D), starting from an arbitrary triple (x^0, y^0, s^0) with $x^0 > 0$ and $s^0 > 0$. His algorithm is a two-phase approach. Phase I is devoted to obtain the feasibility and Phase II improves the duality gap. As we mentioned in Section 2.1, he implicitly solves the system (2.1) at an iteration of Phase I. The Newton step corresponding to this system is obtained from the system (2.2). Using some step size $\alpha \in (0, 1]$, he obtains strictly feasible iterates $x + \alpha\Delta x > 0$, $y + \alpha\Delta y$ and $s + \alpha\Delta s > 0$ for the new perturbed pair (P_{ν^+}) and (D_{ν^+}), with $\nu^+ = (1 - \alpha)\nu$. In other words, the feasibility improves by a factor $1 - \alpha$ at an iteration of Phase I. Once the parameter ν is small enough the algorithm enters Phase II which is a FIPM.

A nice feature of Lustig's algorithm is that Phase I of the algorithm does not involve any computations with huge coefficients like 'big M', and therefore it is more stable than other implementations of IPMs such as the one given in [71]. He achieved this by applying some FIPM to solve the artificial pair (P_M) and (D_M) and then taking M infinitely large. In this way, he managed to eliminate 'M' from the Newton steps. He named the new Newton directions *the limiting search directions*; as we will show in Subsection 2.4.2, these directions coincide with the Newton steps obtained from (2.2).

The next subsection deals with the definition of the Newton directions of the artificial pair (P_M) and (D_M).

2.4.1 Newton steps for the artificial pair

Throughout this section, we will make frequent use of the following notations:

$$\bar{A} = \begin{pmatrix} A & r_b^0 & 0 \\ -r_c^0 & 0 & 1 \end{pmatrix}, \quad \bar{b} = \begin{pmatrix} b \\ M \end{pmatrix}, \quad \bar{c} = (c; M; 0),$$

and the vector of variables

$$\bar{x} = (x; x_{n+1}; x_{n+2}), \quad \bar{y} = (y; y_{m+1}) \quad \text{and} \quad \bar{s} = (s; s_{n+1}; s_{n+2}).$$

[8] User's guide: `http://www.stanford.edu/group/SOL/guides/minos55.pdf`

[9] `http://www.stanford.edu/group/SOL/minos.htm`

[10] `http://www-neos.mcs.anl.gov/neos/solvers/nco:MINOS/AMPL.html`

Using these notations, the pair (P_M) and (D_M) can be rewritten as follows:

$$\min\left\{\bar{c}^T\bar{x}: \quad \bar{A}\bar{x}=\bar{b}, \quad \bar{x}\geq 0\right\}, \tag{2.8}$$

and

$$\max\left\{\bar{b}^T\bar{y}: \quad \bar{A}^T\bar{y}+\bar{s}=\bar{c}, \quad \bar{s}\geq 0\right\}. \tag{2.9}$$

As we mentioned in the previous section, strictly feasible solutions are available for the pair (P_M) and (D_M) given as in (2.6) and (2.7). Thus, Lustig [62] applies the generic feasible interior-point algorithm, introduced by Megiddo [72] and further studied by Kojima et al. [56] and Monteiro and Adler [80], to this pair. It turns out the Newton search directions $(\Delta_M\bar{x}, \Delta_M\bar{y}, \Delta_M\bar{s})$, defined as

$$\Delta_M\bar{x}=\begin{pmatrix}\Delta x\\ \Delta x_{n+1}\\ \Delta x_{n+2}\end{pmatrix}, \quad \Delta_M\bar{y}=\begin{pmatrix}\Delta y\\ \Delta y_{m+1}\end{pmatrix}, \quad \text{and} \quad \Delta_M\bar{s}=\begin{pmatrix}\Delta s\\ \Delta s_{n+1}\\ \Delta s_{n+2}\end{pmatrix},$$

can be calculated from the system

$$\begin{aligned}\bar{A}\Delta_M\bar{x} &= 0, && (2.10a)\\ \bar{A}^T\Delta_M\bar{y}+\Delta_M\bar{s} &= 0, && (2.10b)\\ \bar{s}\Delta_M\bar{x}+\bar{x}\Delta_M\bar{s} &= r_{xs}, && (2.10c)\end{aligned}$$

where

$$r_{xs}=\mu\mathbf{e}-\bar{x}\bar{s}, \quad \text{for some } \mu>0.$$

Lustig does not deal with the last system directly. Instead, by solving this system for $r_{xs}=-\bar{x}\bar{s}$ to obtain $(\Delta_M^a\bar{x}, \Delta_M^a\bar{y}, \Delta_M^a\bar{s})$, and for $r_{xs}=\mathbf{e}$ to get $(\Delta_M^c\bar{x}, \Delta_M^c\bar{y}, \Delta_M^c\bar{s})$, he calculates the solution of the system (2.10) by using the following relation:

$$(\Delta_M\bar{x}, \Delta_M\bar{y}, \Delta_M\bar{s})=(\Delta_M^a\bar{x}, \Delta_M^a\bar{y}, \Delta_M^a\bar{s})+\mu\,(\Delta_M^c\bar{x}, \Delta_M^c\bar{y}, \Delta_M^c\bar{s}).$$

In the next subsection, we describe how Lustig succeeded in eliminating M.

2.4.2 Limiting search directions

Defining

$$\bar{D}=\operatorname{diag}\left(\sqrt{\frac{\bar{x}}{\bar{s}}}\right),$$

it can be verified that

$$\bar{A}\bar{D}^2\bar{A}^T\Delta_M^a\bar{y}=\bar{b}.$$

Substituting $\bar{A}$, $\bar{b}$ and $\bar{D}$, after some simplifications and reductions, the last equation turns out to be equivalent to

$$\begin{pmatrix} AD^2A^T + \frac{x_{n+1}}{s_{n+1}} r_b^0 {r_b^0}^T & -AD^2 r_c^0 \\ -{r_c^0}^T D^2 A^T & {r_c^0}^T D^2 r_c^0 + \frac{x_{n+2}}{s_{n+2}} \end{pmatrix} \Delta_M^a \bar{y} = \begin{pmatrix} b \\ M \end{pmatrix},$$

where $D := \operatorname{diag}(\sqrt{x/s})$. It can be verified that the last system is equivalent to the following system of equations:

$$\begin{pmatrix} AD^2A^T + \frac{x_{n+1}}{s_{n+1}} r_b^0 {r_b^0}^T - \frac{AD^2 r_c^0 {r_c^0}^T D^2 A^T}{{r_c^0}^T D^2 r_c^0 + \frac{x_{n+2}}{s_{n+2}}} & 0 \\ \frac{-{r_c^0}^T D^2 A^T}{{r_c^0}^T D^2 r_c^0 + \frac{x_{n+2}}{s_{n+2}}} & 1 \end{pmatrix} \Delta_M^a \bar{y} = \begin{pmatrix} b + \frac{MAD^2 r_c^0}{{r_c^0}^T D^2 r_c^0 + \frac{x_{n+2}}{s_{n+2}}} \\ \frac{M}{{r_c^0}^T D^2 r_c^0 + \frac{x_{n+2}}{s_{n+2}}} \end{pmatrix}. \tag{2.11}$$

Note that M appears also in the artificial variables x_{n+2} and s_{n+1}, and this dependence is:

$$x_{n+2} = M + {r_c^0}^T x \quad \text{and} \quad s_{n+1} = M - {r_b^0}^T y.$$

Hence, when driving M to infinity, the system (2.11) boils down to

$$\begin{pmatrix} AD^2A^T & 0 \\ 0 & 1 \end{pmatrix} \lim_{M\to\infty} \Delta_M^a \bar{y} = \begin{pmatrix} b + s_{n+2} AD^2 r_c^0 \\ s_{n+2} \end{pmatrix}. \tag{2.12}$$

Denoting

$$\lim_{M\to\infty} \Delta_M^a \bar{y} = (\Delta_\ell^a y, \Delta_\ell^a y_{m+1})^T,$$

the relation (2.12) implies that

$$\Delta_\ell^a y = \left(AD^2A^T\right)^{-1} \left(b + s_{n+2} AD^2 r_c^0\right) \quad \text{and} \quad \Delta_\ell^a y_{m+1} = s_{n+2}.$$

Obviously, the expressions for $\Delta_\ell^a y$ and $\Delta_\ell^a y_{m+1}$ do not depend on M. Using (2.10b), one may write

$$\lim_{M\to\infty} \Delta_M^a \bar{s} = \lim_{M\to\infty} -\bar{A}^T \Delta_M^a \bar{y} = -\begin{pmatrix} A^T & -r_c^0 \\ {r_b^0}^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta_\ell^a y \\ s_{n+2} \end{pmatrix}$$
$$= \begin{pmatrix} -A^T \Delta_\ell^a y + s_{n+2} r_c^0 \\ -{r_b^0}^T \Delta_\ell^a y \\ -s_{n+2} \end{pmatrix}.$$

In a similar way, we eliminate M from the primal direction. Using (2.10c) with $r_{xs} = -\bar{x}\bar{s}$, one has

$$\Delta_{M}\bar{x} + \bar{D}^2\Delta_{M}\bar{s} = \bar{s}^{-1}r_{xs} = -\bar{x}.$$

Thus, one gets

$$\Delta^a_{M}\bar{x} = -\bar{x} - \bar{D}^2\Delta^a_{M}\bar{s}.$$

Substitution of $\bar{D}$, $\bar{x}$ and $\Delta^a_{M}\bar{s}$ implies that

$$\Delta^a_{M}\bar{x} = \begin{pmatrix} -x + D^2(A^T\Delta^a_\ell y - s_{n+2}r^0_c) \\ -x_{n+1} + \frac{x_{n+1}}{s_{n+1}}{r^0_b}^T\Delta^a_\ell y \\ 0 \end{pmatrix}.$$

By driving M to infinity, one gets

$$\lim_{M\to\infty} \Delta^a_{M}\bar{x} = \begin{pmatrix} -x - D^2\Delta^a_\ell s \\ -x_{n+1} \\ 0 \end{pmatrix}, \tag{2.13}$$

where $\Delta^a_\ell s = -A^T\Delta^a_\ell y - y_{m+1}r^0_c$. Since, by (2.9), $s_{n+2} = -y_{m+1}$, the search directions $(\Delta^a_\ell x, \Delta^a_\ell y, \Delta^a_\ell s)$ are defined as follows:

$$\begin{aligned} \Delta^a_\ell x &= -x - D^2\Delta^a_\ell s, \\ \Delta^a_\ell y &= \left(AD^2A^T\right)^{-1}\left(b - y_{m+1}AD^2r^0_c\right), \\ \Delta^a_\ell s &= -A^T\Delta^a_\ell y - y_{m+1}r^0_c. \end{aligned}$$

Lustig named these directions the *limiting search directions.* It can be easily verified that these limiting directions satisfy the relations

$$\begin{aligned} A\Delta^a_\ell x &= x_{n+1}r^0_b, \\ A^T\Delta^a_\ell y + \Delta^a_\ell s &= -y_{m+1}r^0_c, \\ s\Delta^a_\ell x + x\Delta^a_\ell s &= -xs. \end{aligned} \tag{2.14}$$

Similarly, denoting

$$\lim_{M\to\infty} \Delta^c_{M}\bar{y} = (\Delta^c_\ell y; \Delta^c_\ell y_{m+1}),$$

it can be verified that

$$\Delta^c_\ell y = -\left(AD^2A^T\right)^{-1}As^{-1} \quad \text{and} \quad \Delta^c_\ell y_{m+1} = 0. \tag{2.15}$$

Substitution in (2.10b), after setting $r_{xs} = \mathbf{e}$ in (2.10), implies that

$$\lim_{M\to\infty} \Delta_M^c \bar{s} = \left(-A^T \Delta_\ell^c y; -r_b^{0^T} \Delta_\ell^c y; 0\right).$$

By replacing $\Delta_M^c \bar{s}$ in (2.10c), after setting $r_{xs} = \mathbf{e}$ in (2.10), we arrive at

$$\lim_{M\to\infty} \Delta_M^c \bar{x} = \left(s^{-1} - D^2 \Delta_\ell^c s; 0; \frac{1}{s_{n+2}}\right), \tag{2.16}$$

where

$$\Delta_\ell^c s = -A^T \Delta_\ell^c y.$$

As a result, the *limiting centering* steps $(\Delta_\ell^c x, \Delta_\ell^c y, \Delta_\ell^c s)$ are calculated as follows:[11]

$$\begin{aligned}
\Delta_\ell^c x &= s^{-1} - D^2 \Delta_\ell^c s,\\
\Delta_\ell^c y &= -\left(AD^2A^T\right)^{-1} A s^{-1},\\
\Delta_\ell^c s &= -A^T \Delta_\ell^c y.
\end{aligned}$$

It can be easily verified that these limiting centering directions satisfy

$$\begin{aligned}
A\Delta_\ell^c x &= 0,\\
A^T \Delta_\ell^c y + \Delta_\ell^c s &= 0,\\
s\Delta_\ell^c x + x\Delta_\ell^c s &= \mathbf{e}.
\end{aligned} \tag{2.17}$$

Now given the directions $(\Delta_\ell^a x, \Delta_\ell^a y, \Delta_\ell^a s)$ and $(\Delta_\ell^c x, \Delta_\ell^c y, \Delta_\ell^c s)$ as defined, respectively, by (2.14) and (2.17), Lustig calculates the parameters μ and α such that new iterates (x^+, y^+, s^+) defined as

$$\begin{aligned}
x^+ &= x + \alpha(\Delta_\ell^a x + \mu \Delta_\ell^c x),\\
y^+ &= y + \alpha(\Delta_\ell^a y + \mu \Delta_\ell^c y),\\
s^+ &= s + \alpha(\Delta_\ell^a s + \mu \Delta_\ell^c s),
\end{aligned} \tag{2.18}$$

satisfy $(x^+, s^+) > 0$ and

$$\|r_b\| \le 100\|r_c\|. \tag{2.19}$$

The condition (2.19) serves to assure that the primal and dual feasibility are achieved at a close rate. Moreover, in some LO problems with empty dual interior, some dual variables tend to zero while their complementary variables blow up. This causes some numerical instabilities. This condition serves to overcome this

[11] We dub the search directions obtained in this way "limiting centering steps" to make it clear that they are obtained by driving $M \to \infty$. This term was not used by Lustig [62].

issue too, by keeping the amount of primal and dual infeasibility close to each other.

By (2.13) and (2.16), one has $\Delta_l^a x_{n+1} = -x_{n+1}$ and $\Delta_l^c x_{n+1} = 0$. This implies that after an iteration of the algorithm the artificial variable x_{n+1} is updated as follows:

$$x_{n+1}^+ = (1-\alpha)x_{n+1}. \tag{2.20}$$

Moreover, (2.9) and (2.12) imply that $\Delta_l^a y_{m+1} = s_{n+2} = -y_{m+1}$. By (2.15), $\Delta_l^c y_{m+1} = 0$. Therefore, after an iteration of the algorithm the artificial variable y_{m+1} is updated as follows:

$$y_{m+1}^+ = (1-\alpha)y_{m+1}. \tag{2.21}$$

Because $x_{n+1}^0 = -y_{m+1}^0 = 1$, $r_b^0 \neq 0$ and $r_c^0 \neq 0$, the variables x_{n+1} and $-y_{m+1}$ remain equal throughout the algorithm according to (2.20) and (2.21). Thus, one may assume $x_{n+1} = -y_{m+1} = \nu$ for some $\nu \in [0, 1]$.

Note that according to the definition of (P_M) and (D_M), the iterates always satisfy the relations

$$\begin{aligned} Ax &= b - \nu r_b^0, \quad x \geq 0, \\ A^T y + s &= c - \nu r_c^0, \quad s \geq 0, \end{aligned} \tag{2.22}$$

which means that the iterates (x, y, s) are strictly feasible for some perturbed pair (P_ν) and (D_ν). On the other hand, the parameter ν reduces by a factor $(1-\alpha)$. It follows that after a Newton step $(\Delta x, \Delta y, \Delta s)$ of Lustig's algorithm, defined as

$$\Delta x = \Delta_\ell^a x + \mu \Delta_\ell^c x, \quad \Delta y = \Delta_\ell^a y + \mu \Delta_\ell^c y \quad \text{and} \quad \Delta s = \Delta_\ell^a s + \mu \Delta_\ell^c s, \tag{2.23}$$

the new iterates (x^+, y^+, s^+), given by (2.18), are strictly feasible for the perturbed pairs (P_{ν^+}) and (D_{ν^+}) with $\nu^+ := (1-\alpha)\nu$. This means that Lustig's algorithm generates a sequence of strictly feasible triples (x, y, s) of the perturbed pairs (P_ν) and (D_ν) for decreasing values of ν. If $\alpha = 1$ occurs at some iteration, then one has $\nu^+ = 0$ which means that the primal and the dual feasibility have been obtained.

By (2.14) and (2.17), it can be easily verified that the Newton step $(\Delta x, \Delta y, \Delta s)$, given by (2.23), solves the system (2.2) which defines the Newton step corresponding to the perturbed pair (P_ν) and (D_ν).

By (2.22), at an iteration of the algorithm, the residual vectors r_b and r_c satisfy

$$r_b = \nu r_b^0 \quad \text{and} \quad r_c = \nu r_c^0. \tag{2.24}$$

Using these, the system (2.2) can be rewritten as follows:

$$\begin{aligned} A\Delta x &= r_b, \\ A^T \Delta y + \Delta s &= r_c, \\ s\Delta x + x\Delta s &= \mu \mathbf{e} - xs, \end{aligned} \tag{2.25}$$

which coincides with the Newton search directions for the system (1.2).

Although Lustig established a monotonic reduction of the parameter ν which seems to be sufficient to prove global convergence, however, as mentioned in Section 2.1, he could not present any theoretical convergence proof because of two reasons: first, in the case where (P) or (D) is infeasible or unbounded or has empty interior, the parameter ν could not be set to zero at any iteration; and second, it was not easy to guarantee simultaneous occurrence of feasibility and optimality. In the next section, we explain the algorithm of Kojima, Megiddo and Mizuno which is a globally convergent variant of Lustig's algorithm. They present some solutions to the above two issues which guarantee global convergence.

2.5 The algorithm of Kojima et al.

Kojima, Megiddo and Mizuno [54] consider the algorithm of Lustig [62], described in the previous section. Just as Lustig, they allow different step sizes α_P and α_D along the primal and dual directions, respectively. They establish that there is an $\alpha^* \in (0,1)$ for which $\min\{\alpha_P, \alpha_D\} \geq \alpha^*$. Without loss of generality, for the sake of simplicity, we feel free to assume that $\alpha_P = \alpha_D = \alpha$ for some $\alpha \in (0,1)$. As mentioned in Section 2.1, α is chosen such that the iterates satisfy (2.3) and (2.4). The condition (2.3) serves to overcome the issue (i), described in Section 2.1 and the condition (2.4) is useful when dealing with the issue (ii). In short, in the algorithm presented in [54], the iterates belong to the *modified* infinity neighborhood $\mathcal{N}$ of the central path[12] of (P) and (D), defined as

$$\begin{aligned}\mathcal{N} := \{(x,y,s) : x, s \in \mathbb{R}^n_{++},\ & xs \geq \gamma\mu_g \mathbf{e}, \\ & \|r_b\| \leq \varepsilon_p \text{ or } \gamma_p \|r_b\| \leq x^T s, \\ & \|r_c\| \leq \varepsilon_d \text{ or } \gamma_d \|r_c\| \leq x^T s, \\ & \gamma \in (0,1),\ (\gamma_p, \gamma_d, \varepsilon_p, \varepsilon_d) > 0\}.\end{aligned} \tag{2.26}$$

Under the assumption that a primal-dual optimal solution exists, global convergence follows by showing that in each iteration both the infeasibility and the duality gap are decreasing by a constant factor. Their algorithm is able to detect a region where no feasible point exists.

They also presented an extension of their algorithm to LCP; this is discarded because it is beyond the scope of this thesis.

We proceed with explaining how the authors obtain a lower bound α^* of the step size α.

[12] It is worth mentioning that Kojima et al. call $\mathcal{N}$ the neighborhood of the central path of (P) and (D). This statement may not be correct in general because if (P) and (D) have empty interior then there is no central path. However, one can find some $\nu \in (0,1)$ for which the iterates belong to the neighborhood $\mathcal{N}_{-\infty}(\gamma)$ of the central path of the perturbed pair (P$_\nu$) and (D$_\nu$).

2.5.1 Global convergence

As Lustig [62], Kojima et al. calculate the Newton directions $(\Delta x, \Delta y, \Delta s)$ from the system (2.25). A minor difference between Kojima et al.'s directions and Lustig's is that in Kojima et al.'s algorithm, the barrier parameter μ is set a priori to

$$\mu = \beta_1 \mu_g, \quad 0 < \beta_1 < 1.$$

with μ_g given by (1.13). This guarantees improvement of both the feasibility and the duality gap at an iteration. Recall that, in [62], as long as the infeasibility insists, Lustig cares more about the feasibility than the optimality and the parameter μ may increase or decrease, whilst in [54], the authors push the algorithm to reduce both the duality gap and the infeasibility at an iteration.

Kojima et al. reduce the duality gap slightly slower than the infeasibility. To this end, they choose the step size α such that the following two relations hold:

$$(x, y, s) + \alpha(\Delta x, \Delta y, \Delta s) \in \mathcal{N}, \tag{2.27a}$$

$$(x + \alpha \Delta x)^T (s + \alpha \Delta s) \leq (1 - \alpha (1 - \beta_2)) x^T s, \tag{2.27b}$$

for a $\beta_2 \in (\beta_1, 1)$. As in Lustig's algorithm [62], one has:

$$\|r_b^+\| = (1 - \alpha)\|r_b\| \quad \text{and} \quad \|r_c^+\| = (1 - \alpha)\|r_c\|, \tag{2.28}$$

where r_b^+ and r_c^+ are the primal and the dual residual vectors at (x^+, y^+, s^+), with

$$(x^+, y^+, s^+) := (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s). \tag{2.29}$$

This means that the residual norms are reduced by a factor $1 - \alpha$ which is smaller than $1 - \alpha (1 - \beta_2)$.

If $\alpha^* > 0$ is a lower bound for the step size α (see Theorem 2.5.1), then (2.27b) implies that the duality gap is decreased as follows:

$$(x + \alpha^* \Delta x)^T (s + \alpha^* \Delta s) \leq (1 - \alpha^* (1 - \beta_2)) x^T s. \tag{2.30}$$

Assuming that an optimal solution exists, this monotonic reduction on the infeasibility and the duality gap implies that after a finite number of iterations they reach or bypass a prescribed accuracy. Precisely speaking, there exists some K such that $\forall k > K$ one has

$$\left\|b - Ax^k\right\| \leq \varepsilon_p \quad \text{and} \quad \left\|c - A^T y^k - s^k\right\| \leq \varepsilon_d \quad \text{and} \quad \left(x^k\right)^T s^k \leq \varepsilon, \tag{2.31}$$

for some positive $\varepsilon_p, \varepsilon_d, \varepsilon$, where (x^k, y^k, s^k) are the iterates generated at the k-th iteration of the algorithm.

The authors implicitly prove the following theorem which says that after a finite number of iterations the algorithm arrives at either a primal-dual optimal ε-solution or at some iterates whose 1-norm is very large. We briefly explain its proof because it yields a positive lower bound α^* for the step size α which is needed in the proof of convergence.

Theorem 2.5.1. *Let $\varepsilon > 0$ and $\omega^* > 0$ be given. There exists a K such that $\forall k \geq K$, the iterates $\left(x^k, y^k, s^k\right)$ generated by the algorithm satisfy either* (2.31) *or*

$$\left\|\left(x^k, s^k\right)\right\|_1 > \omega^*. \tag{2.32}$$

Proof. The proof goes by contradiction. Assume that neither (2.31) nor (2.32) is satisfied throughout the algorithm. Then for any k, one should have

$$\left(x^k\right)^T s^k \geq \varepsilon^* \quad \text{and} \quad \left\|\left(x^k, s^k\right)\right\|_1 \leq \omega^*, \tag{2.33}$$

with

$$\varepsilon^* = \min\left\{\gamma_p \varepsilon_p, \gamma_d \varepsilon_d, \varepsilon\right\}. \tag{2.34}$$

Hence, the sequence $\left\{\left(x^k, y^k, s^k\right)\right\}_{k=1}^{\infty}$ lies in the following compact set:

$$\mathcal{N}^* = \left\{(x, y, s): \quad x^T s \geq \varepsilon^* \quad \text{and} \quad \|(x, s)\|_1 \leq \omega^*\right\}.$$

On the other hand, both the left-hand side coefficient matrix and the right-hand side vector of the system (2.25) are continuous over $\mathcal{N}^*$ for any (x, y, s). As a consequence, the Newton search directions $(\Delta x, \Delta y, \Delta s)$ are bounded above over $\mathcal{N}^*$. That is,

$$\exists \eta \quad \text{s.t.} \quad \left|\Delta x_i \Delta s_i - \gamma \frac{\Delta^T x \Delta s}{n}\right| \leq \eta \quad \text{and} \quad \left|\Delta^T x \Delta s\right| \leq \eta. \tag{2.35}$$

Kojima et al. derive an α^* for which the iterates certainly lie in $\mathcal{N}$ and the duality gap is decreasing to zero. But this is inconsistent with (2.33), which says that the duality gap is always strictly positive. This leads to a contradiction. They obtain α^* as follows.

The iterates belong to $\mathcal{N}$ if the step size α satisfies (i), (ii) and (iii) below:

$$\text{(i)} \;\; f_i(\alpha) = (x_i + \alpha\Delta x_i)(s_i + \alpha\Delta s_i) - \gamma \tfrac{(x_i+\alpha\Delta x_i)^T (s_i+\alpha\Delta s_i)}{n} \geq 0, \quad i = 1, \cdots, n,$$

$$\text{(ii)} \;\; g_p(\alpha) = (x + \alpha\Delta x)^T (s + \alpha\Delta s) - \gamma_p(1-\alpha)\|r_b\| \geq 0 \;\; \text{or} \;\; (1-\alpha)\|r_b\| \leq \varepsilon,$$

$$\text{(iii)} \;\; g_d(\alpha) = (x + \alpha\Delta x)^T (s + \alpha\Delta s) - \gamma_d(1-\alpha)\|r_c\| \geq 0 \;\; \text{or} \;\; (1-\alpha)\|r_c\| \leq \varepsilon_d.$$

Assuming that $g_p(0) \geq 0$ and $g_d(0) \geq 0$, Kojima et al. prove that $f_i, \;\; i = 1, \cdots, n$, g_p and g_d are bounded below as follows:[13]

$$\begin{aligned} f_i(\alpha) &\geq \beta_1 (1-\gamma)(\varepsilon^*/n)\alpha - \eta\alpha^2, \quad i = 1, \cdots, n, \\ g_p(\alpha) &\geq \beta_1 \varepsilon^* \alpha - \eta\alpha^2, \\ g_d(\alpha) &\geq \beta_1 \varepsilon^* \alpha - \eta\alpha^2. \end{aligned} \tag{2.36}$$

[13] Due to their less relevance to the goal of the section, the details regarding the proof of the inequalities (2.36) have been omitted here.

If either $g_p(0) \geq 0$ or $g_d(0) \geq 0$ does not hold, they consider $(1-\alpha)\|r_b\| \leq \varepsilon$ or $(1-\alpha)\|r_c\| \leq \varepsilon_d$, instead of the second or the third inequality, respectively.

One needs α such that the relations (2.27) are satisfied. It can be easily verified that the right hand side expressions in (2.36) are nonnegative for the following α:

$$\hat{\alpha} = \min\left\{1, \frac{\beta_1(1-\gamma)\varepsilon^*}{n\eta}, \frac{\beta_1\varepsilon^*}{\eta}\right\}.$$

This means that by putting $\alpha = \hat{\alpha}$, the iterates certainly lie in $\mathcal{N}$, i.e., (2.27a) holds. On the other hand, the authors establish that

$$(1-\alpha(1-\beta_2))\,x^T s - (x+\alpha\Delta x)^T(s+\alpha\Delta s) \geq (\beta_2-\beta_1)\,\varepsilon^*\alpha - \eta\alpha^2.$$

It can be easily verified that for all α satisfying

$$\alpha \leq \frac{(\beta_2-\beta_1)\,\varepsilon^*}{\eta},$$

the right-hand side is nonnegative. Thus (2.27b) is satisfied for this α.

As a result, by choosing

$$\alpha^* = \min\left\{\hat{\alpha},\ \frac{(\beta_2-\beta_1)\,\varepsilon^*}{\eta}\right\}, \tag{2.37}$$

since $0 < \alpha^* \leq 1$, the equation (2.30) implies that the duality gap is converging to zero. This is in inconsistent with the first condition of (2.33). Thus the proof is complete. □

Now we explain the algorithm in a more formal way in the next subsection.

2.5.2 The algorithm

Algorithm 2.1 Globally convergent IIPM of Kojima et al. [54]

```
Input:
    parameters: $\varepsilon > 0,\ 0 < \gamma < 1, \gamma_p, \gamma_d > 0,\ 0 < \beta_1 < 1,\ \omega^* > 0$;
begin
    initial points: $x^0 > 0$, $y^0$ and $s^0 > 0$;
    while $(\|r_b\| > \varepsilon_p$ or $\|r_c\| > \varepsilon_d$ or $x^T s > \varepsilon)$ & $\|(x,s)\|_1 \leq \omega^*$
        $\mu$-update: $\mu = \beta_1\mu_g$;
        $(x,y,s) = (x,y,s) + \alpha(\Delta x, \Delta y, \Delta s)$;
    endwhile
end
```

As it can be noticed from the description of Algorithm 2.1, it starts at (x^0, y^0, s^0) with $x^0 > 0$ and $s^0 > 0$. The parameters γ, γ_p and γ_d are chosen such that (x^0, y^0, s^0) lie in $\mathcal{N}$. The system (2.25) with $\mu = \beta_1 \mu_g$ is solved to obtain the Newton search directions. After calculating the step sizes α, the new iterates (x^+, y^+, s^+) are obtained using (2.29). This process is repeated until the condition for the **while** statement is violated. If the algorithm stops because of (2.31), then a primal-dual optimal solution has been found; if it stops because of (2.32), then one arrives at a region which does not contain any primal-dual feasible solution. For more details on the latter we refer to the next subsection.

2.5.3 Detecting infeasibility

We start with a theorem.

Theorem 2.5.2. ([54, Theorem 4.1]) *Let $\sigma > 0$ and $\omega > 0$ be such that*

$$(x^0, y^0, s^0) \in S(\sigma, \omega) := \{(x, y, s) : \quad (x; s) \geq \sigma\,(\mathbf{e}; \mathbf{e}) \;\; \& \;\; \|(x; s)\|_1 \leq \omega\}\,.$$

Let ω^* satisfy

$$\frac{\omega^2 + \left(x^0\right)^T s^0}{\sigma} \leq \omega^*.$$

If the algorithm terminates by (2.32), then the region $S(\sigma, \omega)$ contains no primal-dual feasible solution.

Theorem 2.5.2 does not necessarily imply infeasibility as there may be a feasible solution outside the given region. In particular, if (P) or (D) has empty interior, there is no $\sigma > 0$ for which $S(\sigma, \omega)$ contains a feasible solution. To overcome this deficiency of Theorem 2.5.2, Kojima et al. modify their algorithm as follows. If at an iteration, one has $\|r_b\| \leq \varepsilon_p$ and/or $\|r_c\| \leq \varepsilon_d$, then the system (2.25) is used with $r_b = 0$ and/or $r_c = 0$ during the subsequent iterations. In this way, one avoids further improving the residual norm(s). Therefore, Theorem 2.5.2 continues to satisfy. The global convergence of the modified algorithm follows after a slight modification of the proof of Theorem 2.5.1 [54, Section 5].

Assume that the algorithm stops at the k-th iteration because of condition (2.32). Then

$$\|(x^z, s^z)\| \geq \omega^* \quad \text{and} \quad (x^z)^T s^z \geq \varepsilon^*,$$

with ε^* given by (2.34). As a result, one may have several options: if both residuals are less than or equal to the prescribed accuracy parameters, then using a feasible IPM the duality gap is reduced until an approximate optimal solution to the original pair (P) and (D) has been obtained. But, if either of the primal or the dual residuals (or both) are larger than the prescribed tolerances, then by way of some theorems the authors derive some region where no feasible point is available [54, Section 5].

Ongoing efforts on theoretical aspects of the IIPMs were pursued by Zhang [119] who was the first to establish that IIPMs could be polynomial-time. He applied the algorithm studied by Lustig [62] and Kojima et al. [54], to the HLCP and proved that the algorithm has an $O(n^2)$ convergence rate. Since we focus on LO, we present in the next subsection a restriction of this algorithm to LO.

2.6 The algorithm of Y. Zhang

Zhang [119] considers the HLCP, defined as follows:

$$\begin{aligned} Mx + Ns &= h, \\ xs &= 0, \\ (x, s) &\geq 0, \end{aligned} \tag{2.38}$$

where $x, s, h \in \mathbb{R}^n$ and $M, N \in \mathbb{R}^{n \times n}$. It can be easily verified that by setting

$$M = \begin{pmatrix} A \\ 0 \end{pmatrix}, \quad N = \begin{pmatrix} 0 \\ B \end{pmatrix} \quad \text{and} \quad h = \begin{pmatrix} b \\ Bc \end{pmatrix},$$

where $B \in \mathbb{R}^{(n-m)\times n}$ is full row rank and $BA^T = 0$, the system (2.38) becomes equivalent to the KKT system (1.1).

Following Kojima et al. [54], Zhang deals with the system (2.25) with

$$\mu = \sigma\mu_g, \quad \text{for a} \quad \sigma \in (0,1).$$

He uses an initial point which has a special property. Let $(\bar{x}^0, \bar{y}^0, \bar{s}^0)$ be such that

$$A\bar{x}^0 = b \quad \text{and} \quad A^T\bar{y}^0 + \bar{s}^0 = c,$$

but not necessarily $(\bar{x}^0, \bar{s}^0) \geq 0$. The initials (x^0, y^0, s^0), with $x^0 > 0$ and $s^0 > 0$, are taken such that

$$\big(x^0, s^0\big) > (\bar{x}^0, \bar{s}^0). \tag{2.39}$$

It is worth mentioning that, later on, the above restriction on the starting point was relaxed by Stephen Billups and Michael Ferris [14].

Unlike Lustig [62] and Kojima et al. [54], Zhang [119] uses equal primal and dual step sizes. The common step size α is chosen so that new iterates (x^+, y^+, s^+), given by (2.29), satisfy $(x^+, s^+) > 0$ along with the following condition:

$${x^+}^T s^+ \geq (1-\alpha)\, x^T s. \tag{2.40}$$

Following the general policy in IIPMs, the condition (2.40) is considered to assure that the duality gap is reducing not faster than the infeasibility. It can be verified as follows. Using (2.28), one may say

$$\|(r_b^+, r_c^+)\| = (1-\alpha)\|(r_b, r_c)\| \le \frac{{x^+}^T s^+}{x^T s}\|(r_b, r_c)\| \le {x^+}^T s^+ \frac{\|(r_b^0, r_c^0)\|}{{x^0}^T s^0}.$$

This means that the residual norms are always bounded above by a fixed multiple of the duality gap and, therefore, the optimality can not be achieved before the feasibility has been obtained. Therefore, the condition (2.40) coincides with the condition (2.3), used by Kojima et al.'s [54].

To prevent the iterates from premature approach to the boundary, the author imposes the following condition on the iterates:

$$\min(xs) \ge \gamma\mu_g, \quad \text{for a } 0 < \gamma \le \tfrac{\min(x^0 s^0)}{(x^0)^T s^0/n}, \tag{2.41}$$

which is the condition (2.4) of Kojima et al. algorithm [54]. It is worth mentioning that the iterates of Zhang's algorithm are actually captured by a neighborhood $\mathcal{N}$ resembling that of Kojima et al. (see Section 2.5) with $\gamma_p = \gamma_d = (x^0)^T s^0/\|(r_b^0, r_c^0)\|$.

We proceed with showing how Zhang proved global convergence of the algorithm.

2.6.1 Global convergence

The goal is to achieve complementarity and feasibility of the iterates, simultaneously. Note that the most popular tools to measure the duality gap and the infeasibility are, respectively, $x^T s$ and the residual norm $\|(r_b, r_c)\|$. By embedding these two quantities in a so-called *merit function* Φ, defined as

$$\Phi(x,y,s) = x^T s + \|(r_b, r_c)\|\,, \tag{2.42}$$

the author tries to fulfill his goal by assuring some constant reduction on Φ. Note that as Φ tends to zero, so do the residual norms and the duality gap. Let us explain this. The value of the merit function after each Newton step is given by

$$\Phi(x^+, y^+, s^+) = (1-\alpha+\alpha\sigma)x^T s + \alpha^2 \Delta x^T \Delta s + (1-\alpha)\,\|(r_b, r_c)\|\,.$$

This can be rewritten as follows:

$$\Phi(x^+, y^+, s^+) = (1-\varphi(\alpha))\Phi(x,y,s) \tag{2.43}$$

where

$$\varphi(\alpha) = \frac{\alpha\left((1-\sigma)x^T s - \alpha\Delta x^T \Delta s + \|(r_b, r_c)\|\right)}{\Phi(x,y,s)}. \tag{2.44}$$

Naturally, the value of α for which the function $\varphi(\alpha)$ reaches its maximum value in $(0,1)$ may be the best option. This is equivalent to obtaining the maximizer of $\varphi(\alpha)$ subject to the conditions (2.40) and (2.41). Existence of such an α was established in [119, Lemma 5.1].

Lemma 2.6.1. (cf. [119, Lemma 6.3]) *Let* α *be the maximizer of* $\varphi(\alpha)$ *with respect to* (2.40) *and* (2.41). *Then, one has*

(1) $\alpha \geq \min\left\{1, (1-\gamma)\bar{\sigma}\mu_g/\omega^2\right\}$ *where* $\bar{\sigma} = \min\{\sigma, 1-\sigma\}$ *and*

$$\omega := \sqrt{\|D\Delta x\|^2 + \|D^{-1}\Delta s\|^2}\ ;$$

(2) $\varphi(\alpha) \geq \alpha(1-\sigma-\alpha|\Delta^T x\Delta s|/x^T s)$.

By [119, Lemma 6.2], at each iteration one has

$$\omega^2 \leq t x^T s, \tag{2.45}$$

for some $t > 0$ and that the sequence $\{t_k\}$ satisfies

$$\bar{t} = \limsup_{k\to\infty} t_k < \infty. \tag{2.46}$$

It can be easily verified that

$$|\Delta^T x\Delta s| \leq \|D\Delta x\|\|D^{-1}\Delta s\| \leq \frac{\omega^2}{2}. \tag{2.47}$$

By Lemma 2.6.1, and (2.45), one has

$$\alpha \geq \frac{(1-\gamma)\bar{\sigma}}{n\bar{t}}. \tag{2.48}$$

Moreover, by (2.45) and (2.47), one has

$$\frac{|\Delta^T x\Delta s|}{x^T s} \leq \frac{\bar{t}}{2}.$$

Using this, Lemma 2.6.1, part (2), implies that

$$\varphi(\alpha) \geq \alpha\left(1-\sigma-\frac{\alpha\bar{t}}{2}\right).$$

It is assumed that σ satisfies

$$0 < \bar{\sigma} < \sigma < \frac{1}{2}.$$

Then it follows that $1-\sigma > \frac{1}{2}$. As a result, one has

$$\varphi(\alpha) \geq \frac{1}{2}\alpha(1-\alpha\bar{t}).$$

Substituting α from (2.48), and noting that α is a maximizer of $\varphi(\alpha)$, one concludes that

$$\varphi(\alpha) \geq \varphi\left(\frac{(1-\gamma)\bar{\sigma}}{n\bar{t}}\right) \geq \bar{\varphi} := \left(1-\frac{\bar{\sigma}(1-\gamma)}{n}\right)\frac{\bar{\sigma}(1-\gamma)}{2n\bar{t}} > 0. \tag{2.49}$$

This proves that the merit function is reduced by a factor $\bar{\varphi} \in (0,1)$ at each iteration. Thus, the global convergence of the algorithm follows.

Polynomiality of the algorithm is obtained by slightly modifying the starting point. The next subsection deals with this subject.

2.6.2 Polynomiality of the algorithm

Letting

$$(\bar{x}^0, \bar{y}^0, \bar{s}^0) = \arg \min_{(x,y,s)} \left(\sum_{j=1}^{m} |b_j - A_j x|^2 + \sum_{j=1}^{n} |c_j - (A^T)_j y - s_j|^2 \right),$$

where A_j, for $j = 1, \cdots, m$, is the j-th row of the matrix A, the author chooses the starting point (x^0, y^0, s^0) as follows:

$$x^0 = s^0 = \bar{\zeta}\mathbf{e} \quad \text{and} \quad y^0 = \bar{y}^0, \tag{2.50}$$

with $\bar{\zeta}$ satisfying

$$\bar{\zeta} \geq \left\| (\bar{x}^0, \bar{s}^0) \right\|.$$

Note that for an optimal solution (x^*, y^*, s^*), one has $\left\| (\bar{x}^0, \bar{s}^0) \right\| \leq \| (x^*, s^*) \|$. The following lemma is critical.

Lemma 2.6.2. (cf. [119, Lemma 7.2]) *Let*

$$\zeta = \min \left\{ \| (x^*, s^*) \| : \ (x^*, y^*, s^*) \text{ is an optimal solution} \right\},$$

and $\bar{\zeta}$ be such that

$$\bar{\zeta} \geq \frac{\zeta}{\lambda \sqrt{n}}, \tag{2.51}$$

with λ independent from n. Then $\bar{t}$, defined as in (2.46)*, turns out to be $O(n)$.*

Using Lemma 2.6.2, (2.49) implies that

$$\bar{\varphi} = O\left(\frac{1}{n^2}\right),$$

which means that the merit function is decreasing by a factor of $O(1/n^2)$. Therefore, one concludes that if a primal-dual optimal solution exists, after at most

$$O\left(n^2 \log \frac{1}{\varepsilon}\right), \tag{2.52}$$

iterations of the algorithm, the following condition holds:

$$\|\Phi\| \leq \varepsilon. \tag{2.53}$$

Thus, the latter is used as the stopping criterion.

But what if no optimal solution exists? Because the sequence $\{\Phi_k\}_{k=1}^{\infty}$ of the merit function values then converges to a positive number instead of zero, the condition (2.53) will never be satisfied. Zhang proposes some other stopping criterion in addition to (2.53), similar to that of Kojima et al. [54], which gives some information about the region where no feasible solution exists. In short, the extra stopping criterion is constructed as follows. In the proof Lemma 2.6.2, the author proves that if an optimal solution (x^*, y^*, s^*) satisfying $\|(x^*, s^*)\| \leq \zeta$ exists, then the following inequality certainly holds:

$$\frac{\|(r_b, r_c)\|}{\|(r_b^0, r_c^0)\|} \frac{(x^0 - \bar{x}^0)^T s + (s^0 - \bar{s}^0)^T x}{x^T s} \leq 4\lambda + 5.$$

He uses this in getting information about the infeasibility by terminating the algorithm if the latter does not hold at some iteration [119, Theorem 8.1].

So far, it has been proven that IIPMs are polynomial, i.e., there is a variant whose iteration bound is given by (2.52). This bound was improved later by a factor of $O(n)$ by Mizuno [75]. In the next section, we describe the algorithm of Mizuno [75] which enjoys an $O(n)$ convergence rate.

2.7 The algorithm of Mizuno

Mizuno [75] considers the algorithm of Kojima et al. [54] described in Figure 2.1. By slightly modifying the starting point and the stopping criteria, he proves that the algorithm of Kojima et al. is of $O(n^2)$ convergence rate. His starting point is the same as that of Zhang, namely (2.50). A variant with $O(n)$ convergence rate is also presented.

2.7.1 Another IIPM with the convergence rate $O(n^2)$

Mizuno starts from the following point:

$$x^0 = s^0 = \gamma_0 \zeta \mathbf{e} \quad \text{and} \quad y^0 = 0,$$

where $\gamma_0 \in (0, 1]$ is a constant and ζ is such that

$$\zeta \geq \min \left\{ \|(x, s)\|_\infty : \, Ax = b, \; A^T y + s = c \text{ for some } y \right\}.$$

The iterates are forced to stay in the neighborhood $\mathcal{N}$ defined by (2.26) in which the parameters γ_p and γ_d are specified as follows:

$$\gamma_p = \frac{{x^0}^T s^0}{\|r_b^0\|} \quad \text{and} \quad \gamma_d = \frac{{x^0}^T s^0}{\|r_c^0\|}.$$

Following Zhang [119], Mizuno allows equal primal and dual step sizes, i.e., $\alpha_p = \alpha_d = \bar{\alpha}$ where $\bar{\alpha}$ is obtained as described in Subsection 2.5.1.

Let η be defined as in (2.35). In [75, Section 3], the author proves that

$$\eta = O(n) x^T s.$$

By substituting in (2.37), one gets

$$\bar{\alpha} \geq \alpha^* = \min\left\{1, \frac{\beta_1 (1-\gamma)\,\varepsilon^*}{n\eta}, \frac{\beta_1 \varepsilon^*}{\eta}, \frac{(\beta_2 - \beta_1)\,\varepsilon^*}{\eta}\right\} = O\left(\frac{1}{n^2}\right).$$

Recall from Subsection 2.5.1 that the primal and dual feasibility are controlled by a parameter ν which is updated by the factor $(1-\alpha^*)$. Precisely speaking, at each iteration (2.24) is satisfied. Moreover, $(x, y, s) \in \mathcal{N}$ and (2.24) imply that

$$x^T s \geq \frac{(x^0)^T s^0}{\|r_b^0\|}\|r_b\| = \nu (x^0)^T s^0 \quad \text{and} \quad x^T s \geq \frac{(x^0)^T s^0}{\|r_c^0\|}\|r_c\| = \nu (x^0)^T s^0.$$

Besides, by (2.30) the duality gap is decreasing by a factor of $(1 - \alpha^* (1-\beta_2))$ at each iteration. Thus, the algorithm decreases the amount of infeasibility and the duality gap by a factor of $O(1/n^2)$. As a result, if a primal-dual optimal solution exists, after at most $O(n^2 L')$ iterations, with

$$L' = \max\left\{\log\frac{x^T s}{\varepsilon}, \log\frac{\|r_b^0\|}{\varepsilon_p}, \log\frac{\|r_c^0\|}{\varepsilon_d}\right\},$$

the algorithm obtains an ε-solution of (P) and (D) [75, Theorem 2.1].

2.7.2 An IIPM with convergence rate $O(n)$

Mizuno [75, Section 4] obtains a variant of the algorithm described in the last subsection which has $O(n)$ convergence rate. This algorithm is inspired by the predictor-corrector (PC) FIPM of Mizuno et al. [77]. The iterates are captured by the following neighborhood:

$$\mathcal{N}_2(\gamma_1) := \{(x, y, s) : \ x > 0, s > 0, \ \|xs - \mu_g \mathbf{e}\| \leq \gamma_1 \mu_g\}, \tag{2.54}$$

for a $\gamma_1 \in (0, 1)$. Mizuno takes $\gamma_1 = \frac{1}{4}$. An iteration of the new variant is as follows. First, Mizuno calculates the step size α such that new iterates $(\bar{x}, \bar{y}, \bar{s})$, defined as

$$\bar{x} := x + \alpha\Delta x, \quad \bar{y} := y + \alpha\Delta y \quad \text{and} \quad \bar{s} := s + \alpha\Delta s,$$

satisfy

$$\begin{aligned} (\bar{x}, \bar{y}, \bar{s}) &\in \mathcal{N}_2(2\gamma_1), \\ \bar{x}^T \bar{s} &\leq (1 - \alpha\,(1-\beta_2)) x^T s, \\ \bar{x}^T \bar{s} &\geq (1-\alpha)\nu {x^0}^T s^0, \end{aligned} \tag{2.55}$$

where the parameter ν is controlling the feasibility and decreases by the factor $(1-\alpha)$ in each iteration. Next, putting $\mu = \bar{x}^T \bar{s}/n$, the system (1.6) is applied

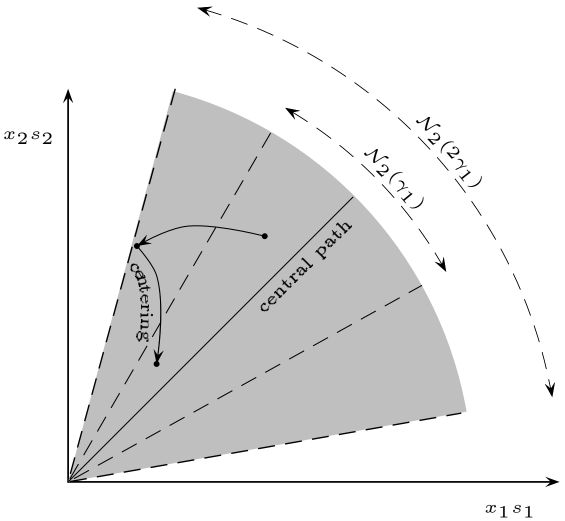


Figure 2.1: Mizuno's $O(n)$-variant IIPM: $\gamma_1 = \frac{1}{4}$.

to obtain a centering step $(\Delta^c x, \Delta^c y, \Delta^c s)$. The new iterates (x^+, y^+, s^+) are calculated as follows:

$$(x^+, y^+, s^+) := (\bar{x}, \bar{y}, \bar{s}) + (\Delta^c x, \Delta^c y, \Delta^c s).$$

In [75, Lemma 4.2] is it established that $(x^+, y^+, s^+) \in \mathcal{N}_2(0.25)$.

It is worth mentioning that, as we made it clear in the last subsection, the feasibility is improving at a rate $(1 - \alpha)$; while due to (2.55), the optimality is improving at slightly slower rate, namely, $(1 - \alpha\,(1 - \beta_2))$.

An illustration of an iteration of the algorithm is given in Figure 2.1. The algorithm starts with a point inside the narrow neighborhood $\mathcal{N}_2(0.25)$ and then derives a point inside the wider one, namely $\mathcal{N}_2(0.5)$. After that using a full-Newton centering step the point is restored into the narrow neighborhood.

In [75, Lemma 4.3], the author proves that the feasibility step size is bounded below as follows:

$$\alpha \geq \alpha^* := \min\left\{\frac{1}{2}, \sqrt{\frac{x^T s}{8n\eta}}, \frac{\beta_1 x^T s}{\eta}, \frac{(\beta_2 - \beta_1) x^T s}{\eta}\right\},$$

where $\eta = O(n)x^T s$. This implies $\alpha^* = O(1/n)$. By [75, Theorem 4.1], the convergence rate the algorithm turns out to be $O(n)$.

So far it has been established that the IIPMs are globally convergent and polynomial and the best known convergence rate, i.e., $O(n)$, is due to Mizuno [75]. We proceed with a PC IIPM with the same convergence rate which is due to Potra [91].

2.8 A PC algorithm of F. Potra

Inspired by Mizuno's PC IIPM, described in the previous section, Potra [91] designed another PC algorithm with the same convergence rate. An iteration of Potra's algorithm consists of three types of Newton steps: two predictor steps and one corrector step. The first predictor step uses an affine-scaling direction which decreases the duality gap while keeping the feasibility unchanged whilst the second one serves to improve feasibility at the same rate as the optimality. The predictor step sizes are chosen in such a way that the generated iterates belong to some wider neighborhood than $\mathcal{N}_2(0.25)$. The corrector step serves to restore these iterates to the narrower neighborhood $\mathcal{N}_2(0.25)$.

2.8.1 The algorithm

The algorithm designed by Potra [91] is presented in Algorithm 2.2. The starting point slightly differs from that of Mizuno and is as follows:

$$x^0 = \xi \mathbf{e}, \quad s^0 = \sigma \mathbf{e} \quad \text{and} \quad y^0 = 0,$$

where the parameters ξ and σ satisfy

$$\left\| A^\dagger b \right\|_\infty \leq \xi \quad \text{and} \quad \|c\|_\infty \leq \sigma,$$

with $A^\dagger$ denoting the pseudoinverse of A, which is defined as $A^\dagger = A^T \left(AA^T\right)^{-1}$.

Notice that the initials (x^0, y^0, s^0) given above are perfectly centered; i.e.,

$$x^0 s^0 = \mu \mathbf{e} \quad \text{with} \quad \mu = \frac{\left(x^0\right)^T s^0}{n}.$$

The first predictor affine-scaling search direction $(\Delta^a x, \Delta^a y, \Delta^a s)$ is calculated from the following system:

$$\begin{aligned} A\Delta^a x &= 0, \\ A^T \Delta^a y + \Delta^a s &= 0, \\ s\Delta^a x + x\Delta^a s &= -xs, \end{aligned} \tag{2.56}$$

along which only the optimality improves. Next, the second predictor step $(\Delta^f x, \Delta^f y, \Delta^f s)$ is calculated from the system

$$\begin{aligned} A\Delta^f x &= r_b, \\ A^T \Delta^f y + \Delta^f s &= r_c, \\ s\Delta^f x + x\Delta^f s &= 0, \end{aligned} \tag{2.57}$$

which improves feasibility. Their effect on the optimality is not significant. The step sizes $\alpha > 0$ and $\theta \in (0,1]$ are calculated such that the iterates $(\bar{x},\bar{y},\bar{s})$, defined as

$$(\bar{x},\bar{y},\bar{s}) := (x,y,s) + \alpha(\Delta^a x, \Delta^a y, \Delta^a s) + \theta(\Delta^f x, \Delta^f y, \Delta^f s), \tag{2.58}$$

belong to $\mathcal{N}_2(\beta)$ where $\beta \in (0.25, 0.5]$. In Subsection 2.8.2 we discuss in detail how to obtain these parameters.

The centering step $(\Delta^c x, \Delta^c y, \Delta^c s)$ is given by

$$\begin{aligned} A\Delta^c x &= 0, \\ A^T\Delta^c y + \Delta^c s &= 0, \\ \bar{s}\Delta^c x + \bar{x}\Delta^c s &= \bar{\mu}\mathbf{e} - \bar{x}\bar{s}, \end{aligned} \tag{2.59}$$

with $\bar{\mu} := \frac{\bar{x}^T\bar{s}}{n}$. The new iterates (x^+, y^+, s^+), defined by

$$(x^+, y^+, s^+) := (\bar{x},\bar{y},\bar{s}) + (\Delta^c x, \Delta^c y, \Delta^c s),$$

are such that $(x^+, y^+, s^+) \in \mathcal{N}_2(0.25)$. See [91, Proposition 2.2] for a proof.

This procedure is repeated until an ε-solution (x,y,s) is obtained.

Algorithm 2.2 The PC algorithm of Potra [91]

```
Input:
  accuracy parameter: $\varepsilon > 0$;
  constants: $\sigma > 0$, $\xi > 0$;
begin
  $x^0 := \xi\mathbf{e}$; $s^0 := \sigma\mathbf{e}$; $y^0 := 0$; $\mu^0 := {x^0}^T s^0/n = \sigma\xi$;
  while $\epsilon(x,y,s) > \varepsilon$
    $(x,y,s) : = (x,y,s) + \alpha(\Delta^a x, \Delta^a y, \Delta^a s) + \theta(\Delta^f x, \Delta^f y, \Delta^f s)$;
    $\mu$-update: $\mu := \frac{x^T s}{n}$;
    centering step:
    $(x,y,x) := (x,y,s) + (\Delta^c x, \Delta^c y, \Delta^c s)$;
  endwhile
end
```

In the next subsection, we explain how the step sizes θ and α are obtained and how convergence of his algorithm is derived.

2.8.2 Convergence of the algorithm

Define

$$\begin{aligned} x(\alpha,\theta) &:= x + \alpha\Delta^a x + \theta\Delta^f x, \\ y(\alpha,\theta) &:= y + \alpha\Delta^a y + \theta\Delta^f y, \\ s(\alpha,\theta) &:= s + \alpha\Delta^a s + \theta\Delta^f s. \end{aligned} \tag{2.60}$$

Denoting the primal and dual residual vectors at $(x(\alpha,\theta), y(\alpha,\theta), s(\alpha,\theta))$ by $r_b(\alpha,\theta)$ and $r_c(\alpha,\theta)$, respectively, it can be verified that

$$r_b(\alpha,\theta) = (1-\theta)r_b \quad \text{and} \quad r_c(\alpha,\theta) = (1-\theta)r_c.$$

Moreover,

$$\mu(\alpha,\theta) = \frac{x(\alpha,\theta)^T s(\alpha,\theta)}{n} = \left(1 - \alpha + \gamma\alpha\theta + \tau\theta^2\right)\mu,$$

with

$$\gamma = \frac{(\Delta^a x)^T \Delta^f x + (\Delta^a s)^T \Delta^f s}{n\mu} \quad \text{and} \quad \tau = \frac{(\Delta^f x)^T \Delta^f s}{n\mu}. \tag{2.61}$$

It follows that

$$\mu(\alpha,\theta) = (1-\theta)\mu, \tag{2.62}$$

if and only if

$$\alpha = \chi(\theta) := \frac{1+\tau\theta}{1-\tau\theta}\theta. \tag{2.63}$$

On the other hand, one has

$$x(\alpha,\theta)s(\alpha,\theta) = (1-\alpha)xs + \alpha^2 h + \alpha\theta g + \theta^2 t,$$

where

$$h = \Delta^a x \Delta^a s, \quad g = \Delta^a x \Delta^f s + \Delta^a s \Delta^f x \quad \text{and} \quad t = \Delta^f x \Delta^f s.$$

Thus, one may write

$$x(\alpha,\theta)s(\alpha,\theta) - \mu(\alpha,\theta) = (1-\alpha^2)f + \alpha^2 h + \alpha\theta\bar{g} + \theta^2\bar{t},$$

where

$$f = xs - \mu\mathbf{e}, \quad \bar{g} = g - \frac{\mathbf{e}^T g}{n} g \quad \text{and} \quad \bar{t} = t - \frac{\mathbf{e}^T t}{n} t.$$

In order to obtain $(x(\alpha,\theta), y(\alpha,\theta), s(\alpha,\theta)) \in \mathcal{N}_2(\beta)$, we need to have

$$\|x(\alpha,\theta)s(\alpha,\theta) - \mu(\alpha,\theta)\mathbf{e}\| \le \beta\mu(\alpha,\theta).$$

Potra shows that by replacing "$\le$" by "$=$" in the last inequality, the resulting equation is equivalent to the following:

$$\theta P_7(\theta) = \beta^2\mu^2 - \|f\|^2 \tag{2.64}$$

where $P_7(\theta)$ is a polynomial of order seven in θ. Since $\|f\| \leq 0.25\ \mu$, one has

$$\beta^2\mu^2 - \|f\|^2 \geq (\beta^2 - 1/16)\mu^2 > 0.$$

On the other hand, $\theta P_7(\theta) = 0$ for $\theta = 0$. This means that there exists an $\theta_0 > 0$ such that

$$\forall\, \theta \in [0, \theta_0], \quad \theta P_7(\theta) \leq \beta^2\mu^2 - \|f\|^2. \tag{2.65}$$

If the equation (2.64) has no zero point in the interval $(0, 1)$, then (2.65) holds for all $\theta \in [0, 1]$. Hence, the largest value of θ is

$$\hat{\theta} := \min\left\{1, \tilde{\theta}\right\} \tag{2.66}$$

where $\tilde{\theta}$ solves the equation (2.64). This means that, (2.65) holds for all $\theta \in [0, \hat{\theta}]$.

Due to (2.63), for the sake of simplicity, the author denotes

$$(x(\theta), y(\theta), s(\theta)) := (x(\alpha, \theta), y(\alpha, \theta), s(\alpha, \theta)), \quad \text{and} \quad \mu(\theta) := \mu(\alpha, \theta), \tag{2.67}$$

where $(x(\alpha, \theta), y(\alpha, \theta), s(\alpha, \theta))$ and $\mu(\alpha, \theta)$ are given by (2.60) and (2.62), respectively. The goal is to show that

$$\forall \theta \in [0, \hat{\theta}], \quad (x(\theta), y(\theta), s(\theta)) \in \mathcal{N}_2(\beta). \tag{2.68}$$

Note that (2.65) is equivalent to

$$\|x(\theta)s(\theta) - \mu(\theta)\mathbf{e}\| \leq \beta\mu(\theta).$$

On the other hand, it has been proven in [91, page 25] that

$$\forall \theta \in [0, \hat{\theta}], \quad x(\theta)s(\theta) > 0,$$

meaning that $(x(\theta), s(\theta)) > 0$. Thus (2.68) holds.

In [91, Theorem 2.4] is it established that the algorithm generates a sequence of points inside $\mathcal{N}_2(0.25)$ and the residual vectors, r_b and r_c, and the barrier parameter μ, satisfy

$$r_b = \nu r_b^0, \quad r_c = \nu r_c^0 \quad \text{and} \quad \mu = \nu\mu^0, \quad \text{with} \quad \mu^0 = \frac{{x^0}^T s^0}{n}, \tag{2.69}$$

where the parameter ν is reducing at least by a factor $(1 - \theta)$.

The following lemma confirms that the generated sequence of iterates is bounded.

Lemma 2.8.1. (cf. [91, Lemma 3.1]) *If (P) and (D) are feasible then the sequence* $\left\{(x^k, y^k, s^k)\right\}_{k=1}^{\infty}$ *generated by the algorithm satisfies*

$$\xi \left\|s^k\right\|_1 + \sigma \left\|x^k\right\|_1 \leq 2n\mu^0 + \xi \left\|x^*\right\| + \sigma \left\|x^*\right\|_1,$$

for an optimal pair (x^*, s^*).

He proceeds with obtaining a lower bound for the step size θ which is a consequence of a couple of lemmas.

Lemma 2.8.2. (cf. [91, Lemma 3.4]) *Let (x^*, y^*, s^*) be a primal-dual optimal solution for (P) and (D). Consider*

$$\sigma^* = \frac{1}{n\sigma}\left\|s^*\right\|_1, \quad \xi^* = \frac{1}{n\xi}\left\|x^*\right\|_1, \quad \gamma_1 = \frac{1}{\xi}\left\|A^\dagger b\right\|_\infty \quad \text{and} \quad \gamma_2 = \frac{1}{\sigma}\left\|c\right\|_\infty .$$

One has

$$\begin{aligned} \left\|D^{-1}\Delta^f x\right\| &\leq (\eta_1 + \eta_2)\sqrt{\mu}, \\ \left\|D\Delta^f s\right\| &\leq (\eta_1 + \eta_2)\sqrt{\mu}, \end{aligned}$$

where $D = \operatorname{diag}\left(\sqrt{x/s}\right)$ and

$$\eta_1 = \frac{\gamma_1 + 1}{\sqrt{1-\beta}}(2 + \sigma^* + \xi^*)n, \tag{2.70a}$$

$$\eta_2 = \frac{\gamma_2 + 1}{\sqrt{1-\beta}}(2 + \sigma^* + \xi^*)n. \tag{2.70b}$$

Lemma 2.8.3. (cf. [91, Lemma 3.5]) *Let (x^*, y^*, s^*) be a primal-dual optimal solution for (P) and (D). Then the step size θ satisfies*

$$\theta \geq \bar{\theta} := \min\left\{1, \frac{1}{2\left|\gamma\right|}, \frac{1}{\sqrt{\left|\tau\right|}}, \gamma_9\right\},$$

with γ and τ given by (2.61)*, and*

$$\gamma_9 = \frac{4(\beta - 0.25)}{\beta - 0.25 + \sqrt{(\beta - 0.25)^2 + 4(\beta - 0.25)\gamma_8}},$$

where

$$\gamma_8 = 2\left|\gamma + \tau\right| + 4\gamma_4 + 2\gamma_5 + \gamma_6,$$

with

$$\begin{aligned} \gamma_4 &= \sqrt{2}n(1 + \sqrt{\left|\tau\right|})^2, \\ \gamma_5 &= 2\sqrt{n}(\eta_1 + \eta_2)(1 + \sqrt{\left|\tau\right|}), \\ \gamma_6 &= \eta_1^2 + \sqrt{2}\eta_1\eta_2 + \eta_2^2, \end{aligned} \tag{2.71}$$

with η_1 and η_2 given by (2.70)*.*

Lemma 2.8.3 along with (2.69) imply

$$\epsilon(x^+, y^+, s^+) \leq (1 - \bar{\theta})\epsilon(x, y, s).$$

We conclude that, the duality gap and the infeasibility are reduced at least by the factor $(1 - \bar{\theta})$. Thus, the following theorem follows.

Theorem 2.8.4. (cf. [91, Theorem 3.6]) *Suppose that (P) and (D) are feasible. Then after at most*

$$\left\lceil \frac{\left|\log \frac{\varepsilon}{\epsilon_0}\right|}{\left|\log(1-\bar{\theta})\right|} \right\rceil,$$

iterations with $\epsilon_0 = \epsilon(x^0, y^0, s^0)$, *the algorithm ends up with an* ε*-solution of the pair (P) and (D).*

2.8.3 Polynomial iteration bound

In [91, Section 4], the author shows that for large enough ξ and σ one gets a polynomial algorithm. He chooses the quantities ξ^* and σ^*, defined in Lemma 2.8.2, such that

$$\xi^* = O(1), \quad \sigma^* = O(1). \tag{2.72}$$

By (2.70) and (2.72),

$$\exists \lambda_1 \geq 1, \quad \eta_1 \leq \lambda_1 \, n, \quad \eta_2 \leq \lambda_1 \, n. \tag{2.73}$$

In [91, page 29], Potra proves that

$$|\gamma| \leq \frac{\eta_1 + \eta_2}{\sqrt{n}}, \quad \text{and} \quad \tau \geq -\frac{\eta_1^2 + \eta_2^2}{n},$$

which implies by (2.73) that

$$|\gamma| \leq 2\lambda_1\sqrt{n}, \quad -2\lambda_1^2 n \leq \tau \leq 0.$$

Due to (2.71), this implies that

$$\gamma_4 = O(n^2), \quad \gamma_5 = O(n^2), \quad \gamma_6 = O(n^2).$$

Hence, for γ_8 and γ_9, defined in Lemma 2.8.3, one can say

$$\gamma_8 = O(n^2) \quad \text{and} \quad \gamma_9 = O(n^{-1}).$$

According to this, we deduce that

$$\theta \geq \bar{\theta} = O(n^{-1}).$$

Using $\bar{\theta} \geq -\log(1-\bar{\theta})$ and Theorem 2.8.4, the iteration bound for the algorithm turns out to be

$$\left\lceil O(n) \log \frac{\epsilon_0}{\varepsilon} \right\rceil.$$

It is worth mentioning that the argument given in this subsection does not imply polynomiality of the algorithm as to calculate θ involves the solution of the equation (2.64) for which there is no polynomial algorithm yet. In [91, Section 4], the author proposes to set θ to a lower bound $\tilde{\theta} = \Omega(1/n)$ that can be obtained in polynomial time. Such a bound can be obtained as follows. Let $w \in \{\eta_1, \eta_2, |\tau|, |\gamma|\}$ and I_w be the smallest integer such that $w \le I_w$. Then, as we establish in Subsection 2.8.3, one has

$$\eta_1 \le I_{\eta_1} = O(n), \;\; \eta_2 \le I_{\eta_2} = O(n), \;\; |\tau| \le I_{|\tau|} = O(n) \text{ and } |\gamma| \le I_{|\gamma|} = O(\sqrt{n}).$$

By replacing $w \in \{\eta_1, \eta_2, |\tau|, |\gamma|\}$ by I_w in (2.71), a lower bound $\tilde{\theta} = \Omega(1/n)$ can be obtained in polynomial time.

Note that if there is no optimal solution, then the inequality (1.21) will never be satisfied. This causes the algorithm to hang. To overcome this issue, he proposes the following solution through which the algorithm detects a region where no optimal solution exists.

Theorem 2.8.5. (cf. [91, Theorem 4.2]) *Let $\bar{\xi}$ and $\bar{\sigma}$ be positive constants. If*

$$\xi \left\| s \right\|_1 + \sigma \left\| x \right\|_1 > 2n\xi\sigma + \xi\bar{\sigma} + \sigma\bar{\xi},$$

holds at some iteration, then there is no optimal solution (x^*, y^*, s^*) such that

$$\left\| x^* \right\|_1 \le \bar{\xi}, \quad \left\| s^* \right\|_1 \le \bar{\sigma}.$$

Although Potra [91] could not improve the convergence rate $O(n)$ of Mizuno's algorithm [75], however, Potra's algorithm has the advantage that the feasibility and the duality gap improve at the same rate.

So far, we have been dealing with IIPMs which are classified as PFMs. We proceed with presenting a potential-reduction IIPM which is due to Mizuno, Kojima and Todd [76].

2.9 Potential-reduction IIPMs of Mizuno et al.

Before going into the potential-reduction algorithm of Mizuno et al. [76], we would like to explain briefly a difference between PFMs and PRMs. As we mentioned in Subsection 1.3.3, PFMs follow the central path which consists of the unique solutions of the system (1.2) for $\mu > 0$. It can be verified (see e.g., [115]) that the central path is nothing else as the set of the minimizers of the classical primal-dual *logarithmic barrier function*, defined as

$$f(x, s; \mu) := \frac{x^T s}{\mu} - \sum_{i=1}^{n} \log(x_i s_i), \tag{2.74}$$

for $\mu > 0$. In PRMs, the potential function has the property that it does not have a minimizer, i.e., it goes to $-\infty$ as the iterates approach an optimal solution.

Tanabe-Todd-Ye [104, 108] introduced the following potential function:

$$\phi(x,s) := \rho \log x^T s - \sum_{i=1}^{n} \log x_i s_i, \quad \text{for a } \rho > n. \tag{2.75}$$

The following lemma is relevant.

Lemma 2.9.1. *If (P) and (D) satisfy the IPC, then $\phi(x,s) \to -\infty$ if and only if the iterates (x,s) approach a primal-dual optimal solution.*

Proof. We define the logarithmic barrier function $\Phi(x,s;\mu)$ as follows:

$$\Phi(x,s;\mu) := \frac{x^T s}{\mu} - n - \sum_{i=1}^{n} \log \frac{x_i s_i}{\mu},$$

which is up to the term $n \log \mu - n$, equal to $f(x,s;\mu)$ as defined by (2.74). For any fixed $\mu > 0$, Φ is strictly convex with the minimum value of 0 and its minimizer occurs at the μ-center, i.e., a primal-dual strictly feasible (x,s) satisfying $xs = \mu\mathbf{e}$. The function ϕ can be rewritten as below:

$$\phi(x,s) = \Phi(x,s;\mu) + \rho \log x^T s - \frac{x^T s}{\mu} + n - n \log \mu, \tag{2.76}$$

for some $\mu > 0$. On the central path, we have $\Phi(x,s;\mu) = 0$ and hence $x^T s = n\mu$. Thus (2.76) implies that

$$\phi(x,s) = \rho \log n + (\rho - n) \log \mu.$$

This implies that ϕ goes to $-\infty$ if and only if $\mu \to 0$, showing that the central path converges to an optimal solution when $\mu \to 0$.

If the iterates are not on the central path, their strict feasibility implies that there is a μ for which the following is satisfied:

$$\Phi(x,s;\mu) \le \tau \quad \text{for some} \quad 0 < \tau < \infty. \tag{2.77}$$

Let us assume that this property is always maintained for some fixed τ. In other words, we assume that the iterates stay in a certain neighborhood of the central path.

It can be easily verified that Φ is strictly convex in μ with the minimizer $\mu = \mu_g := x^T s/n$. Thus, for any μ satisfying (2.77), the following is certainly true:

$$0 \le \Phi(x,s,\mu_g) \le \Phi(x,s;\mu) \le \tau.$$

As a result, without loss of generality, one may take $\mu = \mu_g$ in (2.76). Hence, one gets:

$$\phi(x,s) = \Phi(x,s,\mu_g) + \rho \log n + (\rho - n) \log \mu_g.$$

Because $0 \leq \Phi(x, s, \mu_g) \leq \tau$, the following is always true:

$$\rho \log n + (\rho - n) \log \mu_g \leq \phi(x, s) \leq \tau + \rho \log n + (\rho - n) \log \mu_g.$$

As τ and ρ are constant, this implies that ϕ goes to $-\infty$ if and only if $\mu_g \to 0$, showing that the iterates approach an optimal solution if $\mu \to 0$. □

We would also like to mention that global convergence of both path-following methods and potential-reduction methods has been established by Kojima, Noma and Yoshise [58] for monotone LCP.

We proceed with explaining the algorithm of Mizuno et al. [76]. Basically, the idea is more or less the same as that of the previously described algorithms in the current chapter, specially those described in [75, 91, 119]. The authors use the following starting point:

$$(x^0, y^0, s^0) = \zeta\gamma_0(\mathbf{e}, 0, \mathbf{e}),$$

in which $\gamma_0 \in (0, 1]$ and ζ is a number satisfying

$$\|x^* + s^*\| \leq \zeta, \tag{2.78}$$

for some optimal pair (x^*, s^*). Moreover, the duality gap is decreasing with at most the same speed as the infeasibility. The Newton search directions $(\Delta x, \Delta y, \Delta s)$ are slightly different and obtained from the system (2.25) with $\mu = \mu_\eta$ where

$$\mu_\eta := \frac{x^T s}{n + \eta} \quad \text{with} \quad \eta \geq \sqrt{n}. \tag{2.79}$$

It is worth mentioning that according to [57] and [106], by this choice of μ, the gradient vectors of the potential function ϕ, given by (2.75), and the logarithmic barrier function f, given by (2.74), coincide. As a result, one can make sure that the potential function ϕ is decreasing along the Newton directions (2.25). It was also established that if $\eta = O(\sqrt{n})$, then the convergence rate of the feasible potential-reduction algorithm studied in [57] becomes $O(\sqrt{n})$.

They present two $O(n^2\sqrt{n})$-variants of the algorithm, namely, Algorithm I and II, and an $O(n)$-variant, namely, Algorithm III.

In Algorithm I, the step size α is calculated such that the potential function $\phi_1(x, s)$, defined as[14]

$$\phi_1(x, s) := (n + \eta) \log x^T s - \sum_{i=1}^{n} \log x_i s_i - n \log n, \quad \text{for a } \eta \geq \sqrt{n}. \tag{2.80}$$

decreases by at least a constant and, in addition, the following condition holds:

$$(x + \alpha\Delta x)(s + \alpha\Delta s) \geq (1 - \alpha) x^T s. \tag{2.81}$$

[14] Notice that ϕ_1 is, up to the term $-n \log n$, precisely the Tanabe-Todd-Ye potential-function (2.75).

As we make clear in the sequel, this condition guarantees that the infeasibility decreases at least as much as the duality gap.

In Algorithm II, the step size α is obtained in such a way that a modified version of the above function, defined by

$$\phi_2(x, y, s) := \phi_1(x, s) + \log x^T s - \log\left(x^T s - \sigma \left\|r_b, r_c\right\|\right), \tag{2.82}$$

decreases by at least a constant. In this variant no extra condition is considered; in other word, by adding the extra term $\log x^T s - \log\left(x^T s - \sigma \left\|r_b, r_c\right\|\right)$ to ϕ_1, the extra condition (2.81) was relaxed.

The variant III is actually an $O(n)$-variant of the previous variants.

We proceed with explaining the three variants in more detail.

2.9.1 Algorithm I: a constrained potential-reduction IIPM

In this section, we discuss Algorithm I in a more detail. As described above, the step size α is calculated such that (2.81) holds and the value of the potential function ϕ_1 decreases with a constant $\delta > 0$, i.e., one has

$$\phi_1(x + \alpha\Delta x, s + \alpha\Delta s) - \phi_1(x, s) \leq -\delta. \tag{2.83}$$

If such an α does not exist, then it is concluded that there is no optimal pair (x^*, s^*) satisfying (2.78).

By this amount of reduction, Mizuno et al. [76] establish that after a finite number of iterations, the algorithm arrives at either a point (x, y, s) such that $x^T s \leq \varepsilon$, or a region where there is no optimal solution. See Theorem 2.9.2.

Letting $\{\nu^k\}_{k=1}^{\infty}$ be such that

$$\nu^1 = 1 \quad \text{and} \quad \nu^{k+1} = (1 - \alpha^k)\nu^k,$$

where α^k is the step size in k-th iteration, the residual vectors r_b and r_c satisfy

$$(r_b^k, r_c^k) = \nu^k(r_b^0, r_c^0),$$

and, by (2.81), the duality gap satisfies

$${x^k}^T s^k \geq \nu^k {x^0}^T s^0.$$

This means that the iterates generated by the algorithm have the property that the infeasibility is decreasing at least as much as the duality gap.

It is assumed that at k-th iteration the following holds

$${x^k}^T s^k \geq \gamma_1 \nu^k {x^0}^T s^0,$$

for a $\gamma_1 \in (0, 1]$. Then, they establish that by taking $\alpha = \bar{\alpha}$ where

$$\bar{\alpha} := \frac{\gamma_0^4 \gamma_1^2 \min(xs)}{100n(n + \eta)x^T s},$$

the potential function ϕ_1 decreases with at least the amount of δ where

$$\delta = \frac{\gamma_0^4 \gamma_1^2}{300\,(n+\eta)^2},$$

after each Newton iteration with size $\bar{\alpha}$.

It can be easily verified that $\phi_1(x,s) \geq \eta \log x^T s$. Thus, assuming that a primal-dual optimal solution exists, $x^T s \leq \varepsilon$ holds if $\phi_1(x,s) \leq \frac{1}{\eta}\log\varepsilon$. This will be the case after a finite number of iterations because the potential function is decreasing by δ. As a result, one may conclude that after at most

$$\left\lceil \frac{1}{\delta}\left(\phi_1(x^0,s^0) - \frac{1}{\eta}\log\varepsilon\right)\right\rceil,$$

iterations, the condition $\phi_1(x,s) \leq \frac{1}{\eta}\log\varepsilon$ holds. The following theorem gives a more accurate iteration bound for Algorithm I by imposing some further restrictions on the parameters ζ and ε.

Theorem 2.9.2. (cf. [76, Theorem 1]) *Let $L \geq \log n$ (L may be the binary size of the input data), $\eta \geq \sqrt{n}$, $\gamma_0 \in (0,1]$, and assume that there exists an optimal pair (x^*, s^*) satisfying* (2.78) *with $\log\zeta = O(L)$. Letting ε be such that $\log\frac{1}{\varepsilon} = O(L)$ and assuming that the potential function is decreasing at least with δ, then after at most $O(\eta(n+\eta)^2 L)$ iterations, Algorithm I stops with either a point (x,y,s) for which $x^T s \leq \varepsilon$, or with an optimal pair or we may conclude that that no optimal solution x^* of (P) and (y^*, s^*) of (D) exist satisfying* (2.78).

By (2.79), letting $\eta = O(\sqrt{n})$, the last theorem implies that Algorithm I has $O(n^2\sqrt{n})$ convergence rate.

2.9.2 Algorithm II: a pure potential-reduction algorithm

In this variant, the step size α is calculated such that the function ϕ_2 decreases with at least δ. By replacing the function ϕ_1 by ϕ_2, the condition (2.81) is relaxed. In [76, Lemma 8], the authors establish that if α satisfies (2.81) and (2.83), then one has

$$\phi_2(x+\alpha\Delta x, y+\alpha\Delta y, s+\alpha\Delta s) - \phi_2(x,y,s) \leq -\delta.$$

Moreover, like Algorithm I, it is established that if $\phi_2(x^0,y^0,s^0) = O(\eta L)$ then $\phi_2(x,y,s) \leq \eta\log\varepsilon$ implies $x^T s \leq \varepsilon$ [76, Lemma 7].

In [76, Theorem 6], the authors prove that putting

$$\sigma := \frac{\gamma_1 (x^0)^T s^0}{\|r_b^0, r_c^0\|},$$

in (2.82), Theorem 2.9.2 can be applied to this version of the algorithm as well.

Unfortunately, the iteration complexity of the algorithms I and II is worse than the other IIPMs, described in the previous sections, i.e., $O(n^2\sqrt{2})$ vs. $O(n^2)$ or $O(n)$. Mizuno et al. [76] present variants of the algorithms I and II which have an $O(n)$ convergence rate. In the next subsection, we describe an $O(n)$ variant of Algorithm II.

2.9.3 Algorithm III: a potential-reduction IIPM with $O(n)$ convergence rate

An $O(nL)$-variant of Algorithm II is described in Algorithm 2.3. A difference between Mizuno's PC algorithm [75] and Algorithm 2.3 is that the iterates are not confined to stay in any neighborhood of the homotopy path (see Section 2.7). As it can be noticed, each iteration of the algorithm consists of two types of steps. Step A serves to improve the optimality and the feasibility, and Step B is a centering step.

Algorithm 2.3 The potential-reduction algorithm of Mizuno et al. [76]

```
Input:
   a tolerance $\varepsilon > 0$;
   constants $\gamma_0 \in (0,1]$, $\lambda \in (0,1]$, $\delta_1 > 0, \delta_2 > 0, \eta > 0$;
begin
   consider $x^0 = s^0 = \gamma_0 \zeta \mathbf{e}$ with $\zeta$ given by (2.78), $y^0 = 0$;
   while $x^T s > \varepsilon$
      if $\min(xs) > \lambda x^T s/n$
         Step A:
         $\mu$-update: $\mu := \mu_\eta$;
         $(x, y, s) := (x, y, s) + \alpha(\Delta^f x, \Delta^f y, \Delta^f s)$;
      else
         Step B:
         $\mu$-update: $\mu := \mu_g$;
         $(x, y, s) := (x, y, s) + \alpha(\Delta^c x, \Delta^c y, \Delta^c s)$;
      endif
   endwhile
end
```

If the following condition holds:

$$\min(xs) \geq \lambda \mu_g, \tag{2.84}$$

for a $\lambda \in (0,1]$, the algorithm enters Step A by calculating the Newton search directions $(\Delta^f x, \Delta^f y, \Delta^f s)$ from the system (2.25) for $\mu = \mu_\eta$ with μ_η as defined in (2.79). Then, a step size α is computed in such a way that the following holds:

$$\phi_2(x + \alpha\Delta^f x, y + \alpha\Delta^f y, s + \alpha\Delta^f s) \leq \phi_2(x, y, s) - \delta_1 \tag{2.85}$$

for some $\delta_1 > 0$. In [76, Lemma 11], it is proven that if there exists an optimal pair (x^*, s^*) satisfying (2.78), then there is a step size α for which (2.85) holds with $\delta_1 = 0.001\lambda^2\gamma_0^4\gamma_1^2$.

If (2.84) does not hold, the algorithm performs Step B by calculating Newton search directions $(\Delta^c x, \Delta^c y, \Delta^c s)$ from the system (1.6) with $\mu = \mu_g$. Note that Step B maintains the current infeasibility as well as the duality gap $x^T s$. A step size α is chosen such that the following holds:

$$\phi_2(x + \alpha\Delta^c x, y + \alpha\Delta^c y, s + \alpha\Delta^c s) \leq \phi_2(x, y, s) - \delta_2, \tag{2.86}$$

for some $\delta_2 > 0$. By [76, Lemma 12], (2.86) holds for $\delta_2 = \frac{(1-\lambda)^2}{4}$.

The performance of the algorithm is summarized in the following theorem.

Theorem 2.9.3. (cf. [76, Theorem 9]) *Let $L \geq \log n$ and $\gamma_0 \in (0,1]$ and $\gamma_1 \in (0,1)$. Suppose that $\log\zeta = O(L)$, $\log\frac{1}{\varepsilon} = O(L)$, $\eta \geq n$,*

$$\sigma := \frac{\gamma_1 {x^0}^T s^0}{\|r_b^0, r_c^0\|}, \quad \delta_1 := 0.001\lambda^2\gamma_0^4\gamma_1^2, \quad \text{and} \quad \delta_2 := \frac{(1-\lambda)^2}{4}. \tag{2.87}$$

Then Algorithm III terminates in $O(\eta L)$ iterations.

By setting $\eta = O(n)$, the convergence rate of Algorithm III turns out to be $O(n)$ which coincides with the best known convergence rate obtained by Mizuno [75].

2.9.4 Detecting infeasibility

It has been established in [76, proof of Lemma 4] that if there are optimal solutions x^* of (P) and (y^*, s^*) of (D) satisfying $\|x^* + s^*\|_\infty \leq \zeta$, then the following holds:

$$\theta\gamma_0\zeta \left\|(x,s)\right\|_1 \leq \frac{2x^T s}{\gamma_0\gamma_1}.$$

This means that if the last inequality is violated then there is no optimal solution satisfying $\|x^* + s^*\|_\infty \leq \zeta$.

2.10 Conclusion

This chapter dealt with the theoretical aspects of the IIPMs. After the release of Lustig's algorithm [62], several researchers attempted to derive some globally convergent or polynomial-time variants of Lustig's algorithm. Kojima et al. [54] managed to design a globally convergent variant of Lustig's algorithm by capturing the iterates by the infinity neighborhood of the homotopy path. Using a suitable starting point and a slightly narrower neighborhood, Zhang [119] derived a polynomial-time IIPM. After that Mizuno [75] realized that by further tightening the neighborhood such that the iterates stay very close to the homotopy path, the convergence rate could be improved to $O(n)$. This motivated Potra [91] and Mizuno et al. [76] to design predictor-corrector and potential-reduction IIPMs, respectively, with the convergence rate $O(n)$.

The IIPMs described in this chapter use damped Newton steps. Recently, Roos [97] introduced a primal-dual path-following IIPM which has the advantage that a full-Newton step improves the feasibility and the duality gap with the same speed. Roos achieved this by restricting the iterates to some small neighborhood of the μ-centers of the pairs (P_ν) and (D_ν) as μ and ν reduce with the same speed. The convergence rate of this algorithm coincides with the best known convergence rate for IIPMs, i.e., $O(n)$. Another nice feature of this algorithm is that it has a simple analysis compared with those presented in this chapter. Unfortunately, in practice Roos's algorithm is painfully slow. A simplified version of Roos' algorithm was given by Mansouri and Roos [66] and slightly improved by Gu et al. [46]. Because our large-update IIPM, presented in Chapter 5 is a large-update variant of Roos' algorithm, we describe Roos' algorithm in detail in Chapter 3.

3

A full-Newton step IIPM for LO

3.1 Introduction

Inspired by the full-Newton step FIPM described in Section 1.7, C. Roos [97] presented a full-Newton step IIPM for LO. The strategy is more or less the same as for the other IIPMs, namely to decrease the infeasibility and the duality gap with the same speed. The algorithm is designed in such a way that this can be done using full-Newton steps. The algorithm is also able to detect infeasibility and/or unboundedness. This chapter is devoted to the slightly improved version of Roos' algorithm which is introduced by Gu et al. [46].

Without loss of generality, we assume that both (P) and (D) are feasible. We discuss infeasibility or unboundedness in Section 3.5.

As in other polynomial-time IIPMs, e.g., those studied in [75, 119], Gu et al. use the initials (x^0, y^0, s^0), given by

$$x^0 = s^0 = \zeta \mathbf{e}, \quad y^0 = 0, \tag{3.1}$$

where ζ is a number satisfying

$$\|(x^*, s^*)\| \leq \zeta, \tag{3.2}$$

for some primal-dual optimal solution (x^*, y^*, s^*) of (P) and (D).

This algorithm generates a sequence of triples (x, y, s) in a small neighborhood of the μ-centers of the perturbed pairs (P_ν) and (D_ν) where the parameters μ and ν are simultaneously reduced by a factor $1 - \theta$ with a $\theta \in (0, 1)$. The iterates are obtained by approximately solving the system (2.1).

As we mentioned in Section 2.1, if the original pair (P) and (D) is feasible then, for any $\nu \in (0, 1)$, the perturbed pair (P_ν) and (D_ν) satisfies the IPC and hence the system (2.1) has a unique solution for any $\nu \in (0, 1)$ and $\mu > 0$. This is the content of the following lemma.

Lemma 3.1.1. (cf. e.g., [97, 118]) *The original pair (P) and (D) is feasible if and only if for each* ν *satisfying* $0 < \nu \leq 1$, *the perturbed pair* (P_ν) *and* (D_ν) *satisfy the IPC.*

In order to improve the feasibility and the duality gap with the same speed, throughout the algorithm, the parameters μ and ν are related as follows:

$$\mu = \nu\mu^0 \quad \text{with} \quad \mu^0 = \frac{(x^0)^T s^0}{n} = \zeta^2.$$

As in Algorithm 1.1, the closeness of a triple (x, y, s) to a μ-center is measured by $\delta(x, s; \mu)$, as defined in Section 1.6.

3.2 The algorithm

In this section, we describe an iteration of the algorithm. At the beginning of each iteration, it is assumed that a strictly feasible triple (x, y, s) of (P_ν) and (D_ν), with $\nu \in (0, 1]$, is given which satisfies $x^T s = n\mu$ and $\delta(x, s; \mu) \leq \tau$ for $\mu = \nu\zeta^2$ and a (small) threshold[1] $\tau > 0$. Roos [97] uses $\tau = \frac{1}{8}$ while in Gu et al.'s variant $\tau = \frac{1}{16}$ is used. It can be verified that the initials (x^0, y^0, s^0), given by (3.1), are strictly feasible for (P_1) and (D_1) and $\delta(x^0, s^0, \mu^0) = 0$ which means that at the beginning, $\delta(x, s; \mu) \leq \tau$ certainly holds.

An iteration of the algorithm consists of two types of full-Newton steps: a feasibility step and some centering steps. A feasibility step generates a triple (x^f, y^f, s^f) in the region of quadratical convergence of the μ^+-center of (P_{ν^+}) and (D_{ν^+}) with $(\mu^+, \nu^+) := (1 - \theta)(\mu, \nu)$, in the following sense[2]:

$$\delta(x^f, s^f; \mu^+) \leq \frac{1}{\sqrt[4]{2}}. \tag{3.3}$$

A few centering steps restore the iterates to the τ-neighborhood of the μ^+-center of (P_{ν^+}) and (D_{ν^+}). This procedure is repeated until an ε-solution is obtained. See Algorithm 3.1 for a formal description of the algorithm.

A graphical illustration of an iteration is given by Figure 3.1. The straight lines represent the central paths of the pairs (P_ν) and (D_ν) and (P_{ν^+}) and (D_{ν^+}). The dark gray circles depict the τ-neighborhoods of the μ and μ^+-centers. The region in light gray shows the quadratically convergent region of the μ^+-center of (P_{ν^+}) and (D_{ν^+}). The Newton steps are shown by the arrows and the iterates by the circlets. Each iteration starts at a point inside the τ-neighborhood of the μ-centers of (P_ν) and (D_ν). Using a feasibility step one obtains iterates inside the light gray region. After using some centering steps we get iterates in the dark gray neighborhood of the μ^+-center of (P_{ν^+}) and (D_{ν^+}).

[1] In the sequel we call the set of triples (x, y, s) satisfying $\delta(x, s; \mu) \leq \tau$, the τ-neighborhood of the μ-center.

[2] By (1.16), if $\delta(x, s; \mu) \leq 1/\sqrt[4]{2}$ then after a full Newton step, new triple (x^+, y^+, s^+) satisfies $\delta(x^+, s^+; \mu) \leq \delta(x, s; \mu)^2$ which means that Newton's method is quadratically convergent.

Algorithm 3.1 The full-Newton step IIPM due to Gu et al. [46]

```
Input:
    a threshold parameter $\tau > 0$;
    an accuracy parameter $\varepsilon > 0$;
    a fixed barrier update parameter $\theta$, $0 < \theta < 1$;
    initials $(x^0, y^0, s^0) = \zeta(\mathbf{e}, 0, \mathbf{e})$ for a $\zeta > 0$.
begin
    $x = x^0$, $y = y^0, s = s^0$, and $\nu = 1$; $\mu = \zeta^2$;
    while $x^T s \geq \varepsilon$
        feasibility step:
        $(x, y, s) := (x, y, s) + (\Delta^f x, \Delta^f y, \Delta^f s)$;
        update of $\mu$ and $\nu$:
        $\mu := (1 - \theta)\mu$;
        $\nu := (1 - \theta)\nu$;
        centering steps:
        while $\delta(x, s; \mu) > \tau$
            $(x, y, s) := (x, y, s) + (\Delta x, \Delta y, \Delta s)$;
        endwhile
    endwhile
end
```

The next section deals with the analysis of the algorithm. The hard part of the analysis is the analysis of the feasibility step which yields a suitable value of the barrier updating parameter θ.

3.3 Feasibility step

At the start of an iteration we have a triple (x, y, s), strictly feasible for a perturbed pair (P_ν) and (D_ν), satisfying $\delta(x, s; \mu) \leq \tau$ and $x^T s = n\mu$ where $\mu = \nu\zeta^2$. The algorithm seeks for displacements $(\Delta^f x, \Delta^f y, \Delta^f s)$ for which new iterates (x^f, y^f, s^f), defined as

$$x^f := x + \Delta^f x, \quad y^f := y + \Delta^f y \quad \text{and} \quad s^f := s + \Delta^f s,$$

are as close as possible to the μ^+-center of the pair (P_{ν^+}) and (D_{ν^+}). In other words, we want the displacements to satisfy the system

$$\begin{aligned} A(x + \Delta^f x) &= b - \nu^+ r_b^0, \\ A^T(y + \Delta^f y) + (s + \Delta^f s) &= c - \nu^+ r_c^0, \\ (x + \Delta^f x)(s + \Delta^f s) &= \mu^+ \mathbf{e}. \end{aligned}$$

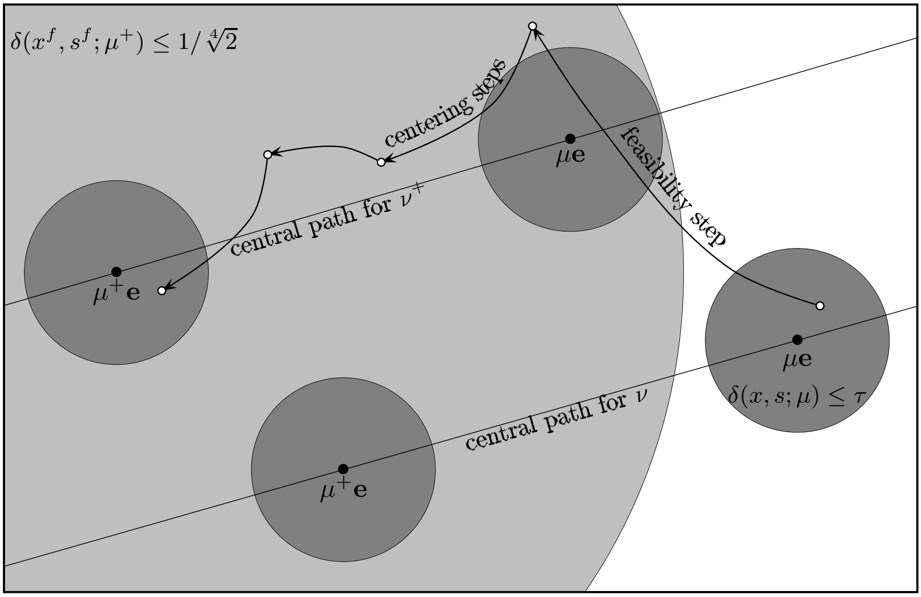


Figure 3.1: An illustration of an iteration of Algorithm 3.1.

The first two equations guarantee that the new triple (x^f, y^f, s^f) is feasible for (P_{ν^+}) and (D_{ν^+}), provided that x^f and s^f are positive. The third equation indicates that we target at the μ^+-center of (P_{ν^+}) and (D_{ν^+}). Using (2.1a) and (2.1b) and after rearranging and linearizing, one obtains the following system:

$$\begin{aligned} A\Delta^f x &= \theta\nu r_b^0, \\ A^T\Delta^f y + \Delta^f s &= \theta\nu r_c^0, \\ s\Delta^f x + x\Delta^f s &= \mu^+\mathbf{e} - xs. \end{aligned} \tag{3.4}$$

This is the system that is used to calculate the Newton directions in the feasibility step.

We proceed with explaining how Gu et al. obtain θ such that after updating μ to μ^+, the triple (x^f, y^f, s^f) satisfies (3.3). To this end, we first need to define the scaled search directions d_x^f and d_s^f:

$$d_x^f := \frac{v\Delta^f x}{x} \quad \text{and} \quad d_s^f := \frac{v\Delta^f s}{s}, \tag{3.5}$$

where v is the variance vector of the iterates (x, y, s) with respect to μ, defined by (1.8).

The following lemma gives a condition on θ which guarantees strict feasibility of (x^f, y^f, s^f) for (P_{ν^+}) and (D_{ν^+}).

Lemma 3.3.1. (cf. [46, Lemma 4.1]) *The iterates (x^f, y^f, s^f) are strictly feasible for (P_{ν^+}) and (D_{ν^+}) if and only if $(1-\theta)\mathbf{e} + d_x^f d_s^f > 0$.*

The next lemma provides an upper bound for $\delta(x^f, s^f; \mu^+)$.

Lemma 3.3.2. (cf. [46, Lemma 4.2]) *Denoting $\delta(v^f) := \delta(x^f, s^f; \mu^+)$ where v^f is the variance vector of the iterates (x^f, y^f, s^f) with respect to μ^+, i.e.,*

$$v^f := \sqrt{\frac{x^f s^f}{\mu^+}}, \tag{3.6}$$

and assuming that $\|d_x^f d_s^f\|_\infty \leq 1-\theta$, then one has

$$4\delta(v^f)^2 \leq \frac{\left\|\frac{d_x^f d_s^f}{1-\theta}\right\|^2}{1 - \left\|\frac{d_x^f d_s^f}{1-\theta}\right\|_\infty}. \tag{3.7}$$

In the sequel, we denote

$$\omega := \tfrac{1}{2}\sqrt{\|d_x^f\|^2 + \|d_s^f\|^2}.$$

One has

$$\|d_x^f d_s^f\| \leq \|d_x^f\|\|d_s^f\| \leq \tfrac{1}{2}\left(\|d_x^f\|^2 + \|d_s^f\|^2\right) = \tfrac{1}{2} \times 4\omega^2 = 2\omega^2$$

and

$$\|d_x^f d_s^f\|_\infty \leq \|d_x^f d_s^f\| \leq 2\omega^2.$$

Due to this, the right-hand side expression of (3.7) is bounded above by

$$\frac{\frac{4\omega^4}{(1-\theta)^2}}{1 - \frac{2\omega^2}{1-\theta}}.$$

Now assuming for the moment that

$$\frac{2\omega^2}{1-\theta} < 1, \tag{3.8}$$

one has

$$\left\|d_x^f d_s^f\right\|_\infty \leq 2\omega^2 < 1-\theta.$$

Thus, by Lemma 3.3.2, (3.3) certainly holds if

$$\frac{\frac{4\omega^4}{(1-\theta)^2}}{1 - \frac{2\omega^2}{1-\theta}} \leq 2\sqrt{2}, \tag{3.9}$$

which is the case if

$$\frac{2\omega^2}{1-\theta} \leq \sqrt{2}\left(\sqrt{1+\sqrt{2}}-1\right) \approx 0.783. \tag{3.10}$$

Note that (3.10) implies (3.8). We conclude that if (3.10) holds, so does (3.3), i.e., $\delta(x^f, s^f; \mu^+) \leq 1/\sqrt[4]{2}$.

In order to proceed we need an upper bound for ω. Before dealing with this issue, we restate the system (3.4) in terms of the scaled Newton directions d_x^f and d_s^f. Using (3.5), it can be easily verified that the system (3.4) can be rewritten as follows:

$$\begin{aligned} \bar{A}d_x^f &= \theta\nu r_b^0, \\ \tfrac{1}{\mu}\bar{A}^T\Delta^f y + d_s^f &= \theta\nu v s^{-1} r_c^0, \\ d_x^f + d_s^f &= r_v, \end{aligned} \tag{3.11}$$

where

$$\bar{A} = AV^{-1}X, \;\; V = \operatorname{diag}(v), \;\; X = \operatorname{diag}(x) \quad \text{and} \quad r_v = (1-\theta)v^{-1} - v.$$

Now we have the following lemma which was proven implicitly by Gu et al. [46][3].

Lemma 3.3.3. *Let ζ be defined as in* (3.2) *and v as in* (1.8)*. Then one has*

$$4\omega^2 \leq 2\|r_v\|^2 + \frac{3\theta^2[\mathbf{e}^T(x+s)]^2}{\zeta^2 v_{\min}^2}, \tag{3.12}$$

where $v_{\min} := \min(v)$.

Gu et al. also proved that

$$2\|r_v\|^2 \leq 8(1-\theta)^2\delta^2 + 2\theta^2 n, \tag{3.13}$$

where $\delta := \delta(x, s; \mu)$.

We recall the following lemma[4] which defines an upper bound for $\mathbf{e}^T(x+s)$.

Lemma 3.3.4. (cf. [46, Lemma 4.3]) *Let (x, y, s) be strictly feasible for (P_ν) and (D_ν), with $x^T s = n\mu$, and ζ as defined in* (3.2)*. Then one has*

$$\mathbf{e}^T(x+s) \leq 2n\zeta. \tag{3.14}$$

[3] If $r_v = 0$, one may obtain a remarkably simple proof for this lemma. The system (3.11) with $r_v = 0$ is used during the feasibility step of the large-update IIPM in Chapter 5. Since the large-update IIPM is the main focus of this thesis, we found it more relevant to present this simplified proof in Chapter 5 and discard the proof of Lemma 3.3.3 here.

[4] Lemma 3.3.4 assumes that an iteration of the algorithm starts with a triple (x, y, s) and a μ which satisfy $x^T s = n\mu$. In general, one may have $x^T s \neq n\mu$. This occurs when damped Newton steps are used. Therefore, we postpone the proof of this lemma to Chapter 5 where we introduce a large-update variant of Gu et al.'s algorithm which uses damped Newton steps and $x^T s = n\mu$ may not be assumed at the beginning of an iteration.

The following lemma gives lower and upper bounds for $v_{\min}$.

Lemma 3.3.5. (cf. [98, Theorem II.62]) *Let* $\rho(\delta) = \delta + \sqrt{\delta^2+1}$ *where* $\delta := \delta(x, s; \mu)$. *Then one has*

$$\frac{1}{\rho(\delta)} \leq v_i \leq \rho(\delta), \quad i = 1, \cdots, n.$$

Using Lemma 3.3.5, and substituting (3.13) and (3.14) in (3.12), after some elementary reductions, one gets the following bound on ω:

$$4\omega^2 \leq 8\delta^2 + 2\theta^2 n + 12n^2\theta^2\rho(\delta)^2. \tag{3.15}$$

Recall that according to Lemma 3.3.2, (3.9) and (3.10), the inequality (3.3) holds if

$$\frac{2\omega^2}{1-\theta} \leq 0.783.$$

Substituting ω from (3.15) in the latter, one gets:

$$8\delta^2 + 2\theta^2 n + 12n^2\theta^2\rho(\delta)^2 \leq 1.566(1-\theta). \tag{3.16}$$

It can be verified that the left-hand side expression in this inequality is increasing in δ. Gu et al. established that by setting

$$\tau = \frac{1}{16} \quad \text{and} \quad \theta = \frac{1}{4n}, \tag{3.17}$$

the inequality (3.16) is satisfied.

3.4 Iteration bound

We established in the last section that if the parameters τ and θ are given by (3.17), after the feasibility step the new iterates (x^f, y^f, s^f) satisfy $\delta(x^f, s^f; \mu^+) \leq 1/\sqrt[4]{2}$. In order to estimate the number of iterations, we need to count the number of centering steps required to obtain new iterates (x^+, y^+, s^+) satisfying $\delta(x^+, s^+; \mu^+) \leq \tau$. This can be done as follows. Recall that (x^f, y^f, s^f) are in the quadratically convergent region of the μ^+-center. Thus, by (1.16), after k centering steps one has

$$\delta(x^+, s^+; \mu^+) \leq \delta(x^f, s^f; \mu^+)^{2^k}.$$

As a result, $\delta(x^+, s^+; \mu^+) \leq \frac{1}{16}$ is satisfied if

$$\delta(x^f, s^f; \mu^+)^{2^k} \leq \left(\frac{1}{\sqrt[4]{2}}\right)^{2^k} \leq \frac{1}{16}.$$

The last inequality implies that $k \leq 4$.

At each iteration of this algorithm, the quantity $\epsilon(x, y, s)$, given by (1.20), is decreasing by the factor $1 - \theta$ with θ given by (3.17). Thus, letting K be the number of μ-updates before an ε-solution is obtained, one has

$$\epsilon(x, y, s) \leq (1-\theta)^K \epsilon(\zeta \mathbf{e}, 0, \zeta \mathbf{e}) \leq \varepsilon.$$

This is equivalent to

$$K \leq -\frac{1}{\log(1-\theta)} \log \frac{\epsilon(\zeta \mathbf{e}, 0, \zeta \mathbf{e})}{\varepsilon}.$$

Using $\theta - \log(1-\theta) > 0$ for any $\theta \in (0, 1)$, this certainly holds if

$$K \leq \frac{1}{\theta} \log \frac{\epsilon(\zeta \mathbf{e}, 0, \zeta \mathbf{e})}{\varepsilon}.$$

Setting $\theta = \frac{1}{4n}$ and considering the fact that at most 5 so-called inner iterations are done per μ-update (one feasibility step and 4 centering steps), the total number of iterations does not exceed

$$\left\lceil 20n \log \frac{\epsilon(\zeta \mathbf{e}, 0, \zeta \mathbf{e})}{\varepsilon} \right\rceil. \tag{3.18}$$

3.5 Detecting infeasibility or unboundedness

We have shown that if (P) and (D) are feasible and ζ satisfies (3.2), then the algorithm finds an ε-solution of (P) and (D), and the number of iterations does not exceed (3.18). In this section, we deal with the case where ζ is too small or (P) and (D) do not have an optimal solution. Although in [46, Remark 4.1], the authors discuss this issue, we present the argument given by Roos [97] which is more comprehensive.

According to Lemma 3.1.1, if the original pair (P) and (D) does not have optimal solution, then there exists some $\bar{\nu} \in (0, 1)$ such that for any $\nu \in (0, \bar{\nu})$ the perturbed pair (P_ν) and (D_ν) does not satisfy the IPC. This means that if at some iteration of the algorithm, after the feasibility step the iterates (x^f, y^f, s^f) do not belong to the quadratically convergent region of the μ^+-center, i.e., $\delta(x^f, s^f; \mu^+) > 1/\sqrt[4]{2}$, then one may conclude that the pair (P) and (D) does not have any optimal solution (x^*, y^*, s^*) satisfying (3.2). To settle uncertainty about the existence of an optimal solution satisfying (3.2) with some larger ζ, Roos [97, Section 4.7] suggests to run the algorithm with $\zeta = 2\zeta$ and repeat if necessary. On the other hand, it is well-known that if (P) and (D) are feasible and their input data, i.e., A, b and c are rational numbers, then there exist a primal-dual optimal solution (x^*, y^*, s^*) which satisfies $\|x^* + s^*\|_\infty \leq 2^L$ with L denoting the size of the input data. Due to this, starting from $\zeta = 1$, after at most L times updating ζ, the algorithm ends up with an optimal solution or declares infeasibility or unboundedness of the problems (P) and (D).

4

Convergence of the homotopy path

4.1 Introduction

As we mentioned in the previous chapters, most IIPMs follow approximately the homotopy path to find an optimal solution of the pair (P) and (D). In this chapter, we establish that if (P) and (D) are feasible then the homotopy path converges to a strictly complementarity solution of (P) and (D).

As in (3.1), we assume that $x^0 = s^0 = \zeta \mathbf{e}$, and $y^0 = 0$, and that ζ is as in (3.2). Recall from Section 2.1 that the homotopy path consists of the μ-centers $(x(\nu), y(\nu), s(\nu))$, of the perturbed pairs (P_ν) and (D_ν), where $\mu = \nu\zeta^2$. In other words, each point on the path is uniquely defined by the following system:

$$\begin{aligned} b - Ax &= \nu(b - A\zeta\mathbf{e}), & x &\geq 0, \\ c - A^T y - s &= \nu(c - \zeta\mathbf{e}), & s &\geq 0, \\ xs &= \nu\zeta^2\mathbf{e}, & 0 &\leq \nu \leq 1, \end{aligned} \tag{4.1}$$

with ζ given by (3.2).

By applying the implicit function theorem, we may easily see that the μ-centers $(x(\nu), y(\nu), s(\nu))$, for $\nu \in (0, 1)$, depends analytically on ν and forms a continuous path. In the sequel, we will investigate the convergence properties of the homotopy path.

4.2 Convergence properties

We denote the *support* of any nonnegative vector x as $\sigma(x)$. So, if $x \in \mathbb{R}^n_+$ then

$$\sigma(x) = \{i : x_i > 0\}.$$

To simplify notation we denote $x = x(\nu)$, $y = y(\nu)$, and $s = s(\nu)$ in the next lemma.

Lemma 4.2.1. *Let (x^*, y^*, s^*) denote an arbitrary optimal solution of (P) and (D). Then we have for any $\nu \in (0,1]$:*

$$(1-\nu)\zeta\left[\sum_{i\in\sigma(x^*)}\frac{x_i^*}{x_i}+\sum_{i\in\sigma(s^*)}\frac{s_i^*}{s_i}\right]=(1-\nu)\mathbf{e}^T(x^*+s^*)-\mathbf{e}^T(x+s)+(1+\nu)\zeta n. \quad (4.2)$$

Proof. Since $b = Ax^*$ and $c = A^Ty^* + s^*$, the system (4.1) can be rewritten as

$$\begin{aligned} A(x^*-x) &= \nu A(x^*-\zeta\mathbf{e}), && x \geq 0,\\ A^T(y^*-y)+(s^*-s) &= \nu(A^Ty^*+s^*-\zeta\mathbf{e}), && s\geq 0,\\ xs &= \nu\zeta^2\mathbf{e}, && 0\leq\nu\leq 1.\end{aligned}$$

Using that the row space of A and its null space are orthogonal, we obtain

$$[(1-\nu)x^*-x+\nu\zeta\mathbf{e}]^T\,[(1-\nu)s^*-s+\nu\zeta\mathbf{e}]=0. \quad (4.3)$$

Since $(x^*)^Ts^* = 0$ we derive from this that

$$(1-\nu)(s^Tx^*+x^Ts^*)=\nu(1-\nu)\zeta\mathbf{e}^T(x^*+s^*)-\nu\zeta\mathbf{e}^T(x+s)+x^Ts+\nu^2\zeta^2\mathbf{e}^T\mathbf{e}.$$

By the definition of the sets $\sigma(x^*)$ and $\sigma(s^*)$ we have $x_i^* = 0$ if $i \notin \sigma(x^*)$ and $s_i^* = 0$ if $i \notin \sigma(s^*)$. Hence it follows that

$$\begin{aligned}(1-\nu)\left[\sum_{i\in\sigma(x^*)}s_ix_i^*+\sum_{i\in\sigma(s^*)}x_is_i^*\right] &= \nu(1-\nu)\zeta\mathbf{e}^T(x^*+s^*)-\\ &\qquad \nu\zeta\mathbf{e}^T(x+s)+x^Ts+\nu^2\zeta^2n.\end{aligned}$$

Using $xs = \nu\zeta^2\mathbf{e}$, from the third equation of (4.1), we get

$$\begin{aligned}(1-\nu)\left[\sum_{i\in\sigma(x^*)}\frac{\nu\zeta^2}{x_i}x_i^*+\sum_{i\in\sigma(s^*)}\frac{\nu\zeta^2}{s_i}s_i^*\right] &= \nu(1-\nu)\zeta\mathbf{e}^T(x^*+s^*)-\\ &\qquad \nu\zeta\mathbf{e}^T(x+s)+\nu\zeta^2n+\nu^2\zeta^2n.\end{aligned}$$

After dividing both sides by $\nu\zeta$ we obtain (4.2), thus completing the proof. □

Since the left-hand side of the identity in (4.2) is nonnegative, the following corollary follows trivially.

Corollary 4.2.2. *For any optimal solution (x^*, y^*, s^*) of (P) and (D) and for any $\nu \in (0,1]$ one has*

$$\mathbf{e}^T(x(\nu)+s(\nu))\leq(1-\nu)\mathbf{e}^T(x^*+s^*)+(1+\nu)\zeta n. \quad (4.4)$$

By Theorem 1.2.3, the problems (P) and (D) have a strictly complementary (optimal) solution $(\hat{x}, \hat{y}, \hat{s})$. Hence, when denoting the classes in the optimal partition of (P) and (D) as B_{opt} and N_{opt}, one has for each optimal solution (x^*, y^*, s^*) of (P) and (D) that

$$B_{opt} = \sigma(\hat{x}) \supseteq \sigma(x^*), \quad N_{opt} = \sigma(\hat{s}) \supseteq \sigma(s^*). \tag{4.5}$$

Lemma 4.2.3. *Let $(\hat{x}, \hat{y}, \hat{s})$ be any strictly complementary solution of (P) and (D). The homotopy path has an accumulation point in the set of optimal solutions of (P) and (D). Moreover, any such accumulation point $(\tilde{x}, \tilde{y}, \tilde{s})$ is strictly complementary and satisfies*

$$\zeta \left[\sum_{i \in B_{opt}} \frac{\hat{x}_i}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{\hat{s}_i}{\tilde{s}_i} \right] = \mathbf{e}^T(\hat{x} + \hat{s}) - \mathbf{e}^T(\tilde{x} + \tilde{s}) + \zeta n. \tag{4.6}$$

Proof. Since the right-hand side in (4.4) depends linearly on ν and $0 \le \nu \le 1$, we have

$$\mathbf{e}^T \left(x(\nu) + s(\nu) \right) \le \max(\mathbf{e}^T(x^* + s^*) + \zeta n, 2\zeta n).$$

Hence the homotopy path, i.e. the set $\{(x(\nu), y(\nu), s(\nu)) : 0 < \nu \le 1\}$, lies in the compact set $\mathbf{e}^T(x(\nu) + s(\nu)) \le \max(\mathbf{e}^T(x^* + s^*) + \zeta n, 2\zeta n)$, where $x(\nu) \ge 0$ and $s(\nu) \ge 0$.

Now let $\nu_1 = 1$ and $\{\nu_k\}_{k=1}^{\infty}$ be a strictly decreasing sequence converging to 0 if $k \to \infty$, and let $x^k = x(\nu_k)$, $y^k = y(\nu_k)$ and $s^k = s(\nu_k)$. Since the sequence (x^k, s^k) lies in a compact set, it has an accumulation point $(\tilde{x}, \tilde{s})$. It follows that a subsequence of the sequence (x^k, s^k) converges to $(\tilde{x}, \tilde{s})$. Without loss of generality we assume below that the sequence (x^k, s^k) itself converges to $(\tilde{x}, \tilde{s})$. Since $(x^k)^T s^k = \nu_k \zeta^2 n$, the sequence $\left\{(x^k)^T s^k\right\}_{k=1}^{\infty}$ is strictly decreasing, and converges to 0. Thus it follows that $\tilde{x}^T \tilde{s} = 0$. Since A has full rank, $\tilde{s}$ determines $\tilde{y}$ uniquely such that $(\tilde{x}, \tilde{y}, \tilde{s})$ is an optimal solution of (P) and (D).

Putting $(x^*, y^*, s^*) = (\hat{x}, \hat{y}, \hat{s})$, $\nu = \nu_k$ and $(x, y, s) = (x^k, y^k, s^k)$ in (4.2), while also using (4.5), we get

$$(1 - \nu_k)\zeta \left[\sum_{i \in B_{opt}} \frac{\hat{x}_i}{x_i^k} + \sum_{i \in N_{opt}} \frac{\hat{s}_i}{s_i^k} \right] = (1 - \nu_k)\mathbf{e}^T(\hat{x} + \hat{s}) - \mathbf{e}^T(x^k + s^k) + (1 + \nu_k)\zeta n,$$

for $k = 1, 2, \cdots$. Now letting k go to ∞, we have that ν_k goes to 0, x^k goes to $\tilde{x}$ and s^k to $\tilde{s}$. Thus we obtain the relation (4.6). Since the right-hand side expression in (4.6) is a real number, the left-hand side expression must be well-defined. Thus it follows that if $i \in B_{opt}$ then $\tilde{x}_i > 0$, and if $i \in N_{opt}$ then $\tilde{s}_i > 0$. Hence it follows that $\sigma(\tilde{x}) = B_{opt}$ and $\sigma(\tilde{s}) = N_{opt}$, proving that $(\tilde{x}, \tilde{y}, \tilde{s})$ is strictly complementary. This completes the proof of the lemma. □

The following lemma makes clear that the homotopy path has only one accumulation point, which implies that it converges.

Lemma 4.2.4. *The homotopy path has precisely one accumulation point in the optimal set.*

Proof. By Lemma 4.2.3 the homotopy path has an accumulation point $(\tilde{x}, \tilde{y}, \tilde{s})$ in the optimal set. Suppose we have another accumulation point $(\bar{x}, \bar{y}, \bar{s})$ of the homotopy path in the optimal set. By applying Lemma 4.2.3 two times, the first time with $(\tilde{x}, \tilde{y}, \tilde{s}) = (\tilde{x}, \tilde{y}, \tilde{s})$ and $(\hat{x}, \hat{y}, \hat{s}) = (\bar{x}, \bar{y}, \bar{s})$ and the second time with $(\tilde{x}, \tilde{y}, \tilde{s}) = (\bar{x}, \bar{y}, \bar{s})$ and $(\hat{x}, \hat{y}, \hat{s}) = (\tilde{x}, \tilde{y}, \tilde{s})$, we obtain

$$\zeta \left[\sum_{i \in B_{opt}} \frac{\bar{x}_i}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{\bar{s}_i}{\tilde{s}_i} \right] = \mathbf{e}^T(\bar{x} + \bar{s}) - \mathbf{e}^T(\tilde{x} + \tilde{s}) + \zeta n,$$

$$\zeta \left[\sum_{i \in B_{opt}} \frac{\tilde{x}_i}{\bar{x}_i} + \sum_{i \in N_{opt}} \frac{\tilde{s}_i}{\bar{s}_i} \right] = \mathbf{e}^T(\tilde{x} + \tilde{s}) - \mathbf{e}^T(\bar{x} + \bar{s}) + \zeta n.$$

By adding these relations, while defining

$$z_i = \begin{cases} \bar{x}_i / \tilde{x}_i, \text{ if } i \in B_{opt}, \\ \bar{s}_i / \tilde{s}_i, \text{ if } i \in N_{opt}, \end{cases}$$

we obtain

$$\zeta \sum_{i=1}^{n} (z_i + z_i^{-1}) = 2\zeta n. \tag{4.7}$$

Since each z_i is the quotient of two positive numbers, we have $z_i > 0$. Therefore,

$$z_i + z_i^{-1} = (z_i^{\frac{1}{2}} - z_i^{-\frac{1}{2}})^2 + 2 \geq 2,$$

with equality if and only if $z_i = 1$. Thus it follows from (4.7) that $z_i = 1$ for each i, which means that $\bar{x} = \tilde{x}$ and $\bar{s} = \tilde{s}$. This proves the lemma. □

We finally prove that the limit of the homotopy path is the analytic center of a subset of the set of optimal solutions.

Lemma 4.2.5. *Let $(\tilde{x}, \tilde{y}, \tilde{s})$ be the limit point of the homotopy path in the optimal set. Then it is the analytic center of the set of optimal solutions (x^*, y^*, s^*) of (P) and (D) satisfying $\mathbf{e}^T(x^* + s^*) \leq \mathbf{e}^T(\tilde{x} + \tilde{s})$.*

Proof. Let $(\tilde{x}, \tilde{y}, \tilde{s})$ be a strictly complementary solution of (P) an (D) that is an accumulation point of the homotopy path. Let $\mathcal{S}(\tilde{x}, \tilde{s})$ denote the set of optimal solutions of (P) and (D) such that $\mathbf{e}^T(x^* + s^*) \leq \mathbf{e}^T(\tilde{x} + \tilde{s})$, and $(x^*, y^*, s^*) \in \mathcal{S}(\tilde{x}, \tilde{s})$. Using similar arguments as in the proof of Lemma 4.2.3, replacing $(\hat{x}, \hat{y}, \hat{s})$ by (x^*, y^*, s^*) and using (4.5), one proves that

$$\zeta \left[\sum_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \right] = \mathbf{e}^T(x^* + s^*) - \mathbf{e}^T(\tilde{x} + \tilde{s}) + \zeta n. \tag{4.8}$$

Using $\mathbf{e}^T(x^* + s^*) \leq \mathbf{e}^T(\tilde{x} + \tilde{s})$, and upon diving both sides by ζ, this implies

$$\sum_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \leq n.$$

The left-hand side expression is a sum of (at most) n nonnegative numbers. Using the arithmetic-geometric-mean inequality we obtain

$$\left(\prod_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} \prod_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \right)^{1/n} \leq \frac{1}{n} \left(\sum_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \right) \leq 1.$$

Thus we have

$$\prod_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} \prod_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \leq 1.$$

It will be convenient to define the function

$$f(x^*, s^*) := \prod_{i \in B_{opt}} x_i^* \prod_{i \in N_{opt}} s_i^* \tag{4.9}$$

on the set of optimal solutions of (P) and (D). Then we have

$$f(x^*, s^*) \leq f(\tilde{x}, \tilde{s}), \quad \forall (x^*, y^*, s^*) \in \mathcal{S}(\tilde{x}, \tilde{s}). \tag{4.10}$$

This means that $(\tilde{x}, \tilde{y}, \tilde{s})$ maximizes the product $\prod_{i \in B_{opt}} x_i^* \prod_{i \in N_{opt}} s_i^*$ on the set $\mathcal{S}(\tilde{x}, \tilde{s})$. Note that $f(\tilde{x}, \tilde{s})$ is positive, because the pair $(\tilde{x}, \tilde{s})$ is strictly complementary. On the other hand, for optimal solutions that are not strictly complementary we have $f(x^*, s^*) = 0$. Hence the maximum of $f(x^*, s^*)$ occurs in a strictly complementary solution. The logarithmic function being strict monotonically increasing we can equally well maximize $\log f(x^*, s^*)$, which has the same maximizer(s) on the set of strictly complementary solutions in $\mathcal{S}(\tilde{x}, \tilde{s})$. However, when the pair (x^*, s^*) is strictly complementary, one has

$$\log f(x^*, s^*) := \sum_{i \in B_{opt}} \log x_i^* + \sum_{i \in N_{opt}} \log s_i^*.$$

Since the set $\mathcal{S}(\tilde{x}, \tilde{s})$ is convex, by definition (see, e.g., [98]) the maximizer of $f(x^*, s^*)$ on $\mathcal{S}(\tilde{x}, \tilde{s})$ is the analytic center of $\mathcal{S}(\tilde{x}, \tilde{s})$. □

A question that arises is wether the limit point of the homotopy path depends on the starting parameter ζ, or not. We answer this question by using the following example.

We consider the case where

$$A = \begin{bmatrix} \alpha & -\beta & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 3 \end{bmatrix},$$

where α and β are positive numbers. Then (P) and (D) are respectively given by

$$\min \{\alpha x_1 + \beta x_2 : \alpha x_1 - \beta x_2 = 0,\ x_1 + x_2 + x_3 = 3,\ x = (x_1; x_2; x_3) \geq 0\},$$
$$\max \{3y_2 : \alpha y_1 + y_2 \leq \alpha,\ -\beta y_1 + y_2 \leq \beta,\ y_2 \leq 0\}.$$

The feasible region of the dual problem is depicted in Figure 4.1. One may easily verify that the set of optimal solutions is given by

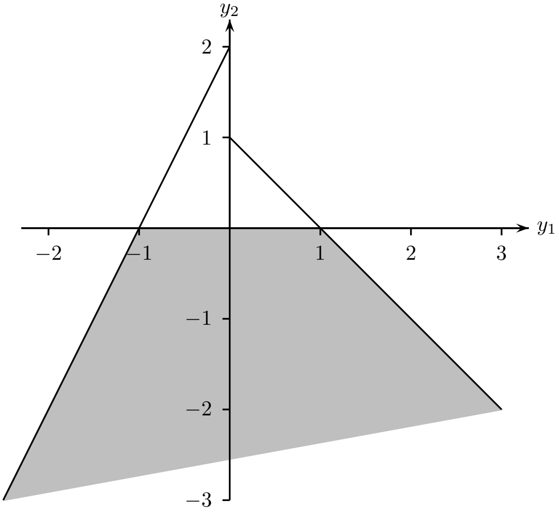


Figure 4.1: Feasible region of the dual problem for $\alpha = 1$ and $\beta = 2$.

$$\left\{ (x, y, s) = \left(\begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} y_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha(1 - y_1) \\ \beta(1 + y_1) \\ 0 \end{bmatrix} \right), -1 \leq y_1 \leq 1 \right\}. \tag{4.11}$$

We conclude from (4.11) that the classes in the optimal partition are given by

$$B_{opt} = \{3\}, \quad N_{opt} = \{1, 2\}.$$

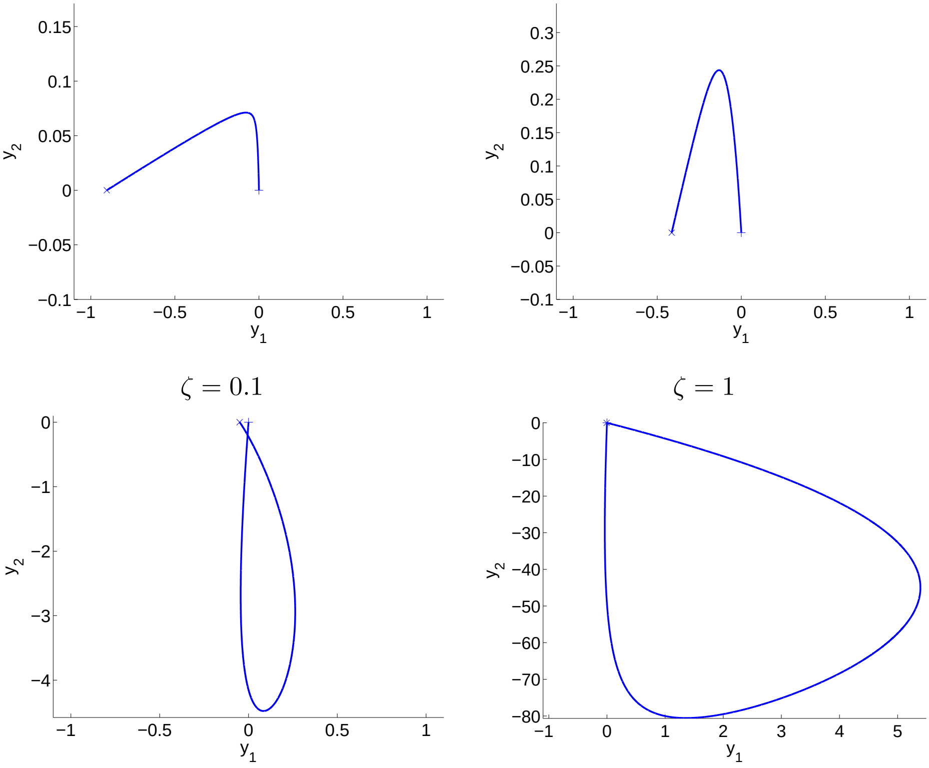
As a consequence we have

$$\prod_{i \in B_{opt}} x_i^* \prod_{i \in N_{opt}} s_i^* = 3 \cdot \alpha(1 - y_1) \cdot \beta(1 + y_1) = 3\alpha\beta(1 - y_1^2).$$

The last expression is maximal for $y_1 = 0$. Hence, putting $y_1 = 0$ in (4.11), we get that the analytic center is given by

$$(x, y, s) = \left(\begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} \right).$$

Now we turn to the homotopy path. We proceed by taking $\alpha = 1$ and $\beta = 2$. For that case we computed numerically the homotopy path for several values of ζ. The results are shown in Figure 4.2. The starting point of the homotopy path is the zero vector, which is drawn as a '+'. The limit point is drawn as a 'x'. The figure clearly demonstrates that the limit point depends highly on the value of ζ. It may be noted that in each of the four cases the limit point is such that $\tilde{y}_1 \leq 0$.



$\zeta = 0.1$

$\zeta = 1$

$\zeta = 10$

$\zeta = 100$

Figure 4.2: Homotopy path for $\alpha = 1$ and $\beta = 2$, and several values of ζ.

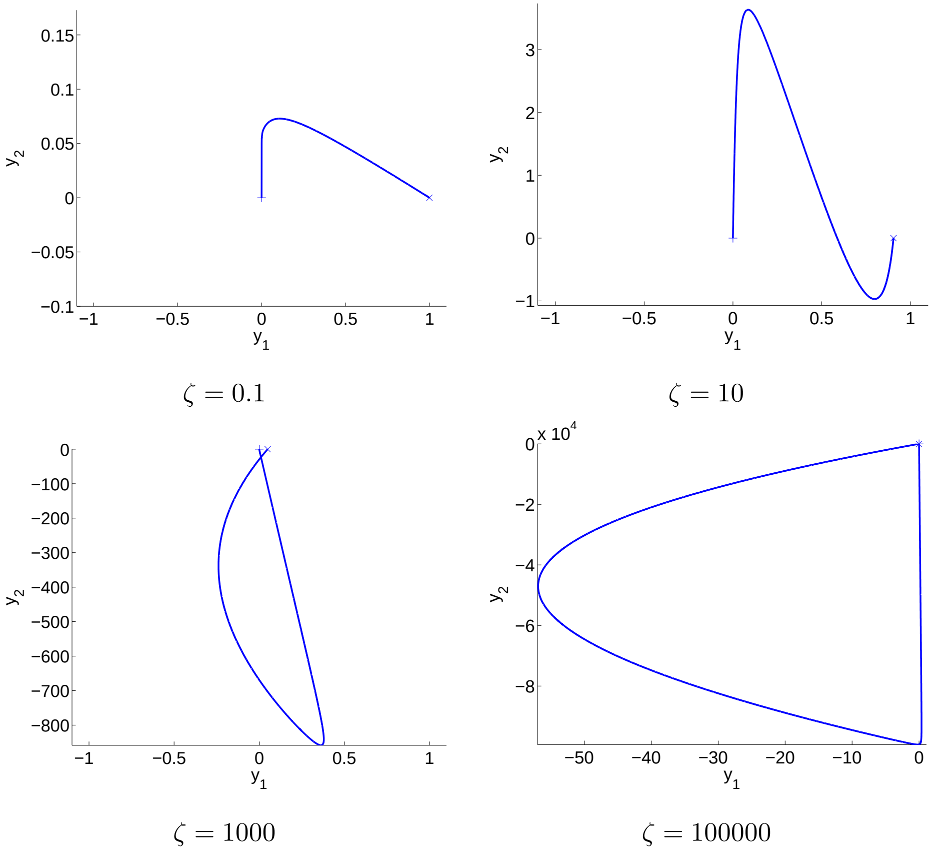
This also follows from Lemma 4.2.5. Because for any optimal solution (x, y, s) we have

$$\mathbf{e}^T(x+s) = 3+\alpha+\beta+(\beta-\alpha)y_1 = 6+y_1, \qquad \prod_{i\in B_{opt}} x_i \prod_{i\in N_{opt}} s_i = 6(1-y_1^2).$$

Hence, by Lemma 4.2.5 we should have

$$y_1 \le \tilde{y}_1 \quad \Rightarrow \quad y_1^2 \ge \tilde{y}_1^2.$$

This implication can be true only if $\tilde{y}_1 \le 0$. When $\alpha > \beta$ one proves in the same way that $\tilde{y}_1 \ge 0$. For an illustration we refer to Figure 4.3.



$\zeta = 0.1$

$\zeta = 10$

$\zeta = 1000$

$\zeta = 100000$

Figure 4.3: Homotopy path for $\alpha = 200$ and $\beta = 100$, and several values of ζ.

One might observe that in all cases it is true that the larger the value of ζ is, the closer the limit point is to the analytic center. This holds indeed in general, as we show in the next lemma.

Lemma 4.2.6. *Let $\zeta > 0$ and $(\tilde{x}, \tilde{y}, \tilde{s})$ be the limit point in the optimal set of the corresponding homotopy path. If ζ goes to infinity then $(\tilde{x}, \tilde{y}, \tilde{s})$ converges to the analytic center of the optimal set.*

Proof. Since $\tilde{x} \geq 0$ and $\tilde{s} \geq 0$ we have $\mathbf{e}^T(\tilde{x} + \tilde{s}) \geq 0$. Hence it follows from (4.8) that

$$\zeta \left[\sum_{i \in B_{opt}} \frac{x_i^*}{\tilde{x}_i} + \sum_{i \in N_{opt}} \frac{s_i^*}{\tilde{s}_i} \right] \leq \mathbf{e}^T(x^* + s^*) + \zeta n,$$

where (x^*, y^*, s^*) denotes some optimal triple. Dividing by ζ at both sides we get

$$\frac{1}{n} \left[\sum_{i \in B_{opt}} \frac{x_i^*}{\lambda \tilde{x}_i} + \sum_{i \in N_{opt}} \frac{s_i^*}{\lambda \tilde{s}_i} \right] \leq 1, \quad \lambda := \frac{\mathbf{e}^T(x^* + s^*)}{\zeta n} + 1.$$

Due to the geometric-arithmetic mean inequality this implies

$$\left[\prod_{i \in B_{opt}} \frac{x_i^*}{\lambda \tilde{x}_i} \prod_{i \in N_{opt}} \frac{s_i^*}{\lambda \tilde{s}_i} \right]^{\frac{1}{n}} \leq 1.$$

With f as defined in (4.9), this implies $f(x^*, s^*) \leq \lambda^n f(\tilde{x}, \tilde{s})$. When ζ goes to infinity, then λ approaches 1, making clear that $(\tilde{x}, \tilde{y}, \tilde{s})$ converges to the analytic center of the optimal set. □

5

A class of Large-update IIPMs for LO

5.1 Introduction

In Chapter 3 we described a full-Newton step IIPM due to Gu et al. [46] which has the property that the iterates stay in a narrow neighborhood of the homotopy path. This algorithm uses full-Newton steps. This enabled us to explore the local quadratic convergence property of Newton's method. As we established, the barrier-updating parameter θ should be small, namely $O(\frac{1}{n})$, which imposes an $O(n)$ convergence rate to the algorithm. This convergence rate coincides with the best known convergence rate for IIPMs. However, the algorithm has the disadvantage that it has a poor performance in practice.

In this chapter we attempt to design a more aggressive variant of the algorithm of Gu et al., i.e., which reduces $\epsilon(x, y, s)$ faster. We would like to mention that this is our aim and also what happens in practice (see Chapter 6). As we will see, however, our algorithm suffers from the same irony that occurs for FIPMs, namely that the theoretical convergence rate of large-update methods is much worse than that of full-Newton variant. In a first attempt we used the classical search direction for primal-dual methods, that is based on the well-known primal-dual logarithmic barrier function, and the theoretical convergence rate turned out to be $O(n^2)$. As we show, however, when using a different barrier function to define the search direction the convergence rate can be improved to $O(n\sqrt{n}(\log n)^3)$.

In order to obtain this result we use a so-called *kernel-function* based barrier function. Any such barrier function is based on a univariate function, called its kernel function. Such functions have been introduced in [10] and are closely related to the so-called self-regular functions introduced in [88]. In these references only FIPMs are considered, and it is shown that these functions are much more efficient for the process of re-centering, which is a crucial part in every FIPM, especially when an iterate is far from the central path. Not surprising, it turns out that these functions are also useful in our large-update IIPM, where re-centering is also a crucial ingredient.

In Section 5.2, we briefly introduce the notion of kernel function, which plays a crucial role in our algorithm. After that, in Section 5.3, as a preparation to our large-update IIPM, we briefly recall the use of kernel-based barrier functions in large-update FIPMs, as presented in [10]. It will become clear in this section that the convergence rate highly depends on the underlying kernel function. The best result is obtained for a specific kernel function, denoted ψ_3, which yields the convergence rate $O(\sqrt{n}\log n)$; this is a factor $\log n$ worse than for full-Newton step FIPMs.

In Section 5.4, we describe our large-update IIPM in detail. In our description we use a search direction based on general kernel functions. The algorithm uses two types of damped Newton steps: a so-called feasibility step and some centering steps. The feasibility step serves to reduce the residual norms, $\|r_b\|$ and $\|r_c\|$, whereas the centering steps keep the residual vectors fixed, but improve the duality gap x^Ts. This procedure is repeated until an ε-solution is obtained. Though many parts of our analysis are valid for general kernel function, at some places we restrict ourselves to the kernel function $\psi_3(t)$. In Section 5.5, we show that the algorithm based on this kernel function yields an $O(n\sqrt{n}(\log n)^3)$ convergence rate which is a factor $(\log n)^2$ worse than for the IIPM, obtained by Salahi et al. [101]. In Section 5.6, we argue how our algorithm detects infeasibility or unboundedness.

5.2 Kernel functions

In this section, we show that the μ-centers, i.e., the unique solutions of the system (1.2), can be characterized as the minimizers of a suitably chosen primal-dual barrier function. In fact we will define a wide class of such barrier functions, each of which is determined by a kernel function.

A kernel function is just a univariate nonnegative function $\psi(t)$, where $t > 0$, which is strictly convex, minimal at $t = 1$ and such that $\psi(1) = 0$, whereas $\psi(t)$ goes to infinity both when t goes to zero and when t goes to infinity.

Now let (x, y, s) be a primal-dual strictly feasible for (P) and (D). Observe that if v is the variance vector with respect to μ, as given by (1.8), then $v = \mathbf{e}$ holds if and only if (x, y, s) is the μ-center of (P) and (D). Given any kernel function ψ we extend its definition to $\mathbb{R}^n_{++}$ according to

$$\Psi(v) := \sum_{i=1}^{n} \psi(v_i). \tag{5.1}$$

It is obvious that $\Psi(v)$ is nonnegative everywhere, and $\Psi(\mathbf{e}) = 0$. Yet we can define a barrier function $\Phi(x, s, \mu)$ as follows:

$$\Phi(x, s, \mu) := \Psi(v). \tag{5.2}$$

It is now obvious that $\Phi(x,s,\mu)$ is well-defined, nonnegative for every primal-dual strictly feasible (x,y,s), and moreover,

$$\Phi(x,s,\mu)=0 \quad\Leftrightarrow\quad \Psi(v)=0 \quad\Leftrightarrow\quad v=\mathbf{e} \quad\Leftrightarrow\quad (x,y,s)=(x(\mu),y(\mu),s(\mu)).$$

This implies that $(x(\mu),y(\mu),s(\mu))$ is the (unique) minimizer of $\Phi(x,s,\mu)$.

We next give an important example, showing that the well-known logarithmic barrier function arises when taking as a kernel function

$$\psi(t):=\frac{t^2-1}{2}-\log t, \qquad t>0. \tag{5.3}$$

This follows since in that case we may write

$$\begin{aligned}\Phi(x,s,\mu)=\sum_{i=1}^n\psi(v_i)=\sum_{i=1}^n\left(\frac{v_i^2-1}{2}-\log v_i\right)&=\sum_{i=1}^n\left(\frac{\frac{x_is_i}{\mu}-1}{2}-\log\sqrt{\frac{x_is_i}{\mu}}\right)\\&=\frac{1}{2}\left(\frac{x^Ts}{\mu}-\sum_{i=1}^n\log x_i-\sum_{i=1}^n\log s_i+n\log\mu-n\right).\end{aligned}$$

Up to the term $n\log\mu-n$, which does not depend on (x,y,s), the expression within the brackets is precisely the classical primal-dual logarithmic barrier function; due to this term the minimal value of $\Phi(x,s,\mu)$ equals 0.

As in [10] we call the kernel function ψ ***eligible*** if it satisfies the following technical conditions.

$$\begin{aligned}t\psi''(t)+\psi'(t)&>0, \quad t<1,\\ t\psi''(t)-\psi'(t)&>0, \quad t>1,\\ \psi'''(t)&<0, \quad t>0,\\ 2\psi''(t)^2-\psi'(t)\psi'''(t)&>0, \quad t<1.\end{aligned}$$

In the sequel it is always assumed that ψ is an eligible kernel function. Properties of eligible kernel functions will be recalled from [10] without repeating their proofs.

5.3 Large-update FIPMs for LO

In this section we recall from [10] some results for a large-update FIPM for solving (P) and (D) using a kernel-function based barrier function. We assume, without loss of generality, that the triple

$$(x^0,y^0,s^0)=(\mathbf{e},0,\mathbf{e}), \tag{5.4}$$

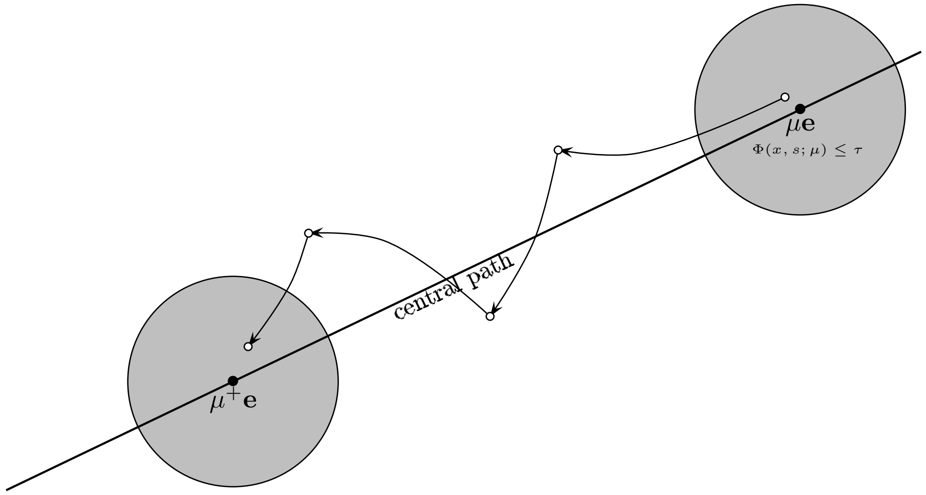


Figure 5.1: An illustration of large-update FIPM: $\mu^{+} = (1 - \theta)\mu$.

is primal-dual feasible.[1] We then have $x^0 s^0 = \mu^0 \mathbf{e}$ for $\mu^0 = 1$. This means that (x^0, y^0, s^0) is the 1-center, and hence $\Phi(x^0, s^0, \mu^0) = 0$. We use this triple to initialize our algorithm.

Each main (or *outer*) iteration of the algorithm starts with a strictly feasible triple (x, y, s) that satisfies $\Phi(x, s, \mu) \leq \tau$ for some $\mu \in (0, 1]$, where τ is a fixed positive constant. It then constructs a new triple (x^{+}, y^{+}, s^{+}) such that $\Phi(x^{+}, s^{+}, \mu^{+}) \leq \tau$ with $\mu^{+} < \mu$. When taking τ small enough, we obtain in this way a sequence of strictly feasible triples that belong to small neighborhoods of a sequence of μ-centers, for a decreasing sequence of μ's. As a consequence, the sequence of constructed triples (x, y, s) converges to an optimal solution of (P) and (D).

We will assume that $\mu^{+} = (1 - \theta)\mu$, where $\theta \in (0, 1)$ is a fixed constant, e.g., $\theta = 0.5$ or $\theta = 0.99$. The larger θ, the more aggressive is the algorithm. Especially when θ is large, each outer iteration will require several so-called inner iterations. See Fig.5.1. The straight line represents the central path of (P) and (D) and the gray circles depict the τ-neighborhoods of the μ-center and the μ^{+}-center. The curved arrows illustrate the damped Newton steps that are used to restore the iterates to the τ-neighborhoods of the μ^{+}-center of (P) and (D). The iterates are shown by the circlets.

[1] The problems (P) and (D) can be embedded into a self-dual problem for which the given triple is a feasible solution and that has an optimal solution that induces optimal solutions for (P) and (D).

Since $\mu^0 = 1$, Lemma 1.7.3 yields the following upper bound for the number of outer iterations of the algorithm:

$$\left\lceil \frac{1}{\theta} \log \frac{n}{\varepsilon} \right\rceil. \tag{5.5}$$

The main task is therefore to get a sharp upper estimate for the number of inner iterations during an outer iteration. We now describe how such an estimate is obtained. We go into some detail, though without repeating proofs, because the results that we recall below are relevant for the IIPM that we discuss in the next section.

As said before, at the start of each outer iteration we have a strictly feasible triple (x, y, s) and $\mu > 0$ such that $\Phi(x, s, \mu) \leq \tau$. We first need to estimate the increase in Φ when μ is updated to $\mu^+ = (1 - \theta)\mu$. For this we need the following lemma.

Lemma 5.3.1. (cf. [10, Theorem 3.2]) *Let $\varrho : [0, \infty) \to [1, \infty)$ be the inverse function of $\psi(t)$ for $t \geq 1$. Then we have for any positive vector v and any $\beta \geq 1$:*

$$\Psi(\beta v) \leq n\psi\left(\beta \varrho\left(\frac{\Psi(v)}{n}\right)\right).$$

Now let v be the variance vector of (x, y, s) with respect to μ. Then one easily understands that the variance vector v^+ of (x, y, s) with respect to μ^+ is given by $v^+ = v/\sqrt{1-\theta}$. Hence, using Lemma 5.3.1 with $\beta = 1/\sqrt{1-\theta}$ we may write

$$\Phi(x, s, \mu^+) = \Psi(v^+) = \Psi\left(\frac{v}{\sqrt{1-\theta}}\right) \leq n\psi\left(\frac{\varrho\left(\frac{\Psi(v)}{n}\right)}{\sqrt{1-\theta}}\right) \leq n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right),$$

where the last inequality holds because ϱ is monotonically increasing and $\Psi(v) = \Phi(x, s, \mu) \leq \tau$. Hence the number $\bar{\tau}$ defined by

$$\bar{\tau} := n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right), \tag{5.6}$$

is an upper bound for the value of Ψ after a μ-update. Note that this bound is independent of the triple (x, y, s); it depends only on the kernel function ψ and the parameters n, τ and θ.

To simplify the notation we redefine μ according to $\mu := \mu^+$. Thus we need to deal with the following question: given a triple (x, y, s) such that $\Phi(x, s, \mu) \leq \bar{\tau}$, how much inner iterations are needed to generate a triple (x, y, s) such that $\Phi(x, s, \mu) \leq \tau$. To answer this question we have to describe an inner iteration. It has been argued in Section 2.2. of [10] that it is natural to define search directions

$(\Delta x, \Delta y, \Delta s)$ by the system

$$\begin{aligned} A\Delta x &= 0, \\ A^T\Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= -\mu v \nabla \Psi(v). \end{aligned}$$

This system has a unique solution. It may be worth pointing out that if ψ is the logarithmic kernel function, as given by (5.3), then $-\mu v \nabla \Psi(v) = \mu \mathbf{e} - xs$, and hence the resulting direction is the primal-dual Newton direction that is used in all primal-dual FIPMs. By doing a line search in this direction with respect to Ψ we get new iterates

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

where α is the step size. According to [10, Lemma 4.4], we use the following default step size:

$$\alpha = \frac{1}{\psi''(\rho(2\delta(v)))},$$

where ρ is the inverse function of $-\frac{1}{2}\psi'(t)$, and

$$\delta(v) := \frac{1}{2} \left\| \nabla \Psi(v) \right\|.$$

Algorithm 5.1 shows a formal description of the algorithm. The closeness of

Algorithm 5.1 A large-update FIPM

```
Input:
  A threshold parameter $\tau > 0$;
  an accuracy parameter $\varepsilon > 0$;
  a fixed barrier update parameter $\theta$, $0 < \theta < 1$;
begin
  $x := \mathbf{e}$; $y := 0$; $s := \mathbf{e}$; $\mu := 1$;
  while $n\mu \geq \varepsilon$,
    $\mu := (1 - \theta)\mu$;
    while $\Psi(v) > \tau$,
      $(x, y, s) := (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$;
      $v := \sqrt{\frac{xs}{\mu}}$;
    endwhile
  endwhile
end
```

(x, y, s) to the μ-center is measured by $\Psi(v)$, where v is the variance vector of

(x, y, s) with respect to the current value of μ. The initial triple (x, y, s) is as given by (5.4) and $\mu = 1$. So we then have $\Psi(v) = 0 \le \tau$. After a μ-update we have $\Psi(v) \le \bar{\tau}$. Then a sequence of inner iterations is performed to restore the inequality $\Psi(v) \le \tau$. Then μ is updated again, and so on. This process is repeated until $n\mu$ falls below the accuracy parameter ε after which we have obtained an ε-solution.

To estimate the number of inner iterations we proceed as follows. Denoting the decrease in the value of Ψ as $\Delta\Psi$, it was shown in [10, Theorem 4.6] that

$$\Delta\Psi \ge \alpha\delta(v)^2 = \frac{\delta(v)^2}{\psi''(\rho(2\delta(v)))}. \tag{5.7}$$

Since the kernel function ψ is eligible, the last expression is increasing in $\delta(v)$ [10, Lemma 4.7]. Besides, by [10, Theorem 4.9], $\delta(v)$ is bounded from below as follows:

$$\delta(v) \ge \tfrac{1}{2}\psi'\left(\varrho\left(\Psi(v)\right)\right). \tag{5.8}$$

Combining (5.7) and (5.8), we arrive at

$$\Delta\Psi \ge \frac{\left(\psi'\left(\varrho\left(\Psi(v)\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(\Psi\left(v\right)\right)\right)\right)\right)}. \tag{5.9}$$

Following [10], let γ be the smallest number such that

$$\Psi(v)^{\gamma-1}\frac{\left(\psi'\left(\varrho\left(\Psi(v)\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(\Psi\left(v\right)\right)\right)\right)\right)} \ge \kappa \tag{5.10}$$

for some positive constant κ, whenever $\Psi(v) \ge \tau$. From the references in Table 5.1 we know that such constants κ and γ exist for the kernel functions in this table. When denoting the value of the barrier function after the μ-update as Ψ_0 and the value after the k-th inner iteration as Ψ_k, it follows from (5.9) and (5.10) that

$$\Psi_0 \le \bar{\tau}, \quad \Psi_k \le \Psi_{k-1} - \kappa\Psi_{k-1}^{1-\gamma},\ k = 1,\, 2\ \ldots, \tag{5.11}$$

with $\bar{\tau}$ as in (5.6). At this stage we may point out why the use of kernel functions other than the logarithmic kernel function may be advantageous. The reason is that if ψ is the logarithmic kernel function then $\gamma = 1$, whence we obtain $\Psi_k \le \Psi_{k-1} - \kappa$ for each $k \ge 1$, provided that $\Psi_{k-1} \ge \tau$. This resembles the well-known fact that the best lower bound for the decrease of the logarithmic barrier function is a fixed constant, no matter what the value of $\Psi(v)$ is. As we will see smaller values of γ can be obtained for other kernel functions, which leads to larger reductions of the barrier function value, and hence lower iteration numbers.

By [10, Lemma 5.1], (5.11) implies that the number of inner iterations will not exceed

$$\frac{\bar{\tau}^\gamma}{\kappa\gamma} = \frac{1}{\kappa\gamma}\left(n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right)\right)^\gamma. \tag{5.12}$$

Multiplying this number by the number of outer iterations, as given by (5.5), we obtain the following upper bound for the total number of iterations:

$$\frac{1}{\theta\kappa\gamma}\left(n\psi\left(\frac{\varrho\left(\frac{\tau}{n}\right)}{\sqrt{1-\theta}}\right)\right)^{\gamma}\log\frac{n}{\varepsilon}.$$

Given a kernel function ψ, it is now straightforward to compute the resulting iteration bound from this expression. Table 5.1 summarizes some results from the literature. In this chapter we consider an IIPM based on the use of a kernel function. Although many of the results below hold for any eligible kernel function, we will concentrate of the kernel ψ_3 in Table 5.1. The reason is that it gives the best possible result among the kernel functions in this table; another nice feature of ψ_3 is that if q approaches 1 then this function converges to the logarithmic kernel function.

i	$\psi_i(t)$	iteration bound	ref.
1	$\frac{t^2-1}{2}-\log t$	$O\left(n\right)\,\log\frac{n}{\varepsilon}$	e.g., [98]
2	$\frac{1}{2}\left(t-\frac{1}{t}\right)^2$	$O\left(n^{\frac{2}{3}}\right)\,\log\frac{n}{\varepsilon}$	[85]
3	$\frac{t^2-1}{2}+\frac{t^{1-q}-1}{q-1}$	$O(\sqrt{n}\log n)\log\frac{n}{\varepsilon},\ q=O(\log n)$	[86]
4	$\frac{t^2-1}{2}+\frac{t^{1-q}-1}{q(q-1)}-\frac{q-1}{q}(t-1)$	$O(\sqrt{n}\log n)\log\frac{n}{\varepsilon},\ q=O(\log n)$	[87]
5	$\frac{t^2-1}{2}+\frac{e^{\frac{1}{t}}-e}{e}$	$O\left(\sqrt{n}\ \log^2 n\right)\,\log\frac{n}{\varepsilon}$	[10]
6	$\frac{t^2-1}{2}-\int_1^t e^{\frac{1}{\xi}-1}d\xi$	$O\left(\sqrt{n}\ \log^2 n\right)\,\log\frac{n}{\varepsilon}$	[10]
7	$\frac{t^2-1}{2}+\frac{(e-1)^2}{e}\frac{1}{e^t-1}-\frac{e-1}{e}$	$O\left(n^{\frac{3}{4}}\right)\,\log\frac{n}{\varepsilon}$	[11]
8	$\frac{t^2-1}{2}+\frac{1}{q}\left(e^{t^{-q}-1}-1\right)$	$O\left(\frac{1}{q}\sqrt{n}\,(\log n)^{q+1}\right)\log\frac{n}{\varepsilon}$	[89]
9	$\frac{t^2-1}{2}-\int_1^t e^{\frac{1}{\xi^q}-1}d\xi$	$O\left(\frac{1}{q}\sqrt{n}\,(\log n)^{q+1}\right)\log\frac{n}{\varepsilon}$	[89]
10	$\frac{t^2-1}{2}-\int_1^t e^{q\left(\frac{1}{\xi}-1\right)}d\xi$	$O\left(\sqrt{n}\log n\right)\log\frac{n}{\varepsilon},\ q=O(\log n)$	[12]
11	$\frac{t^2-1}{2}+\frac{e^{\frac{q}{t}}-e^q}{qe^q}$	$O\left(\sqrt{n}\log n\right)\log\frac{n}{\varepsilon},\ q=O(\log n)$	[2]

Table 5.1: Iteration bounds of large-update FIPMs for several kernel functions.

5.4 Large-update IIPMs for LO

In this section, we attempt to design a large-update IIPM to solve (P) and (D), which is inspired by the full-Newton step algorithm described in Chapter 3. As in the full-Newton variant, our algorithm starts from the initials (x^0, y^0, s^0), given by (3.1) and (3.2), and generates a sequence of positive iterates in a small neighborhood of the homotopy path. Precisely speaking, the iterates belong to a small neighborhood of the μ-centers of the perturbed pairs (P_ν) and (D_ν) where μ and ν are related as $\mu = \nu\zeta^2$ with ζ given by (3.2). Moreover, each iteration reduces ν (and also μ) by a factor $1-\theta$, with $\theta \in (0,1)$. Since we are interested in some larger values of θ than in (3.17), we must expect that after a μ-update the iterates do not belong to the region of quadratic convergence of the new μ-centers. As a result the use of the quantity δ to measure the closeness of the iterates to the homotopy path becomes irrelevant. As for large-update FIPMs, we use a barrier function for this purpose. Let us explain an iteration of the algorithm in a more detail.

5.4.1 An outer iteration of the algorithm

As in Section 5.3, $\Psi(v)$ will denote the barrier function based on the kernel function $\psi(t)$, as given in (5.1). Here v denotes the variance vector of a triple (x, y, s) with respect to $\mu > 0$, and we define $\Phi(x, s, \mu)$ as in (5.2). The algorithm is designed in such a way that at the start of each outer iteration we have $\Psi(v) \leq \tau$ for some threshold value $\tau = O(1)$. As $\Psi(v) = 0$ at the starting points (3.1), the condition $\Psi(v) \leq \tau$ is certainly satisfied at the start of the first outer iteration.

Each outer iteration of the algorithm consists of a feasibility step and some centering steps. At the start of the outer iteration we have a triple (x, y, s) that is strictly feasible for (P_ν) and (D_ν), for some $\nu \in (0,1]$, and that belongs to the τ-neighborhood of the μ-center of (P_ν) and (D_ν), where $\mu = \nu\zeta^2$. We first perform a feasibility step during which we generate a triple (x^f, y^f, s^f) which is strictly feasible for the perturbed problems (P_{ν^+}) and (D_{ν^+}), with $\nu^+ = (1-\theta)\nu$ and, moreover, ***close enough*** to the μ^+-center of (P_{ν^+}) and (D_{ν^+}), with $\mu^+ = \nu^+\zeta^2$. i.e., $\Phi(x^f, s^f; \mu^+) \leq \tau^f$, for some suitable value of τ^f.

After the feasibility step we perform some centering steps to get a strictly feasible triple (x^+, y^+, s^+) of (P_{ν^+}) and (D_{ν^+}) in the τ-neighborhood of the μ^+-center of (P_{ν^+}) and (D_{ν^+}). During the centering steps the iterates stay feasible for (P_{ν^+}) and (D_{ν^+}). Hence for the analysis of the centering steps we can use the analysis presented in the previous section for FIPMs. From this analysis we derive that the number of centering steps will not exceed

$$\frac{(\Phi(x^f, s^f, \mu^+))^\gamma}{\kappa\gamma},$$

where the parameters γ and κ depend on the kernel function ψ. Hence we are left with the problem of defining a suitable search direction $(\Delta^f x, \Delta^f y, \Delta^f s)$ for the feasibility step and to determine θ such that after the feasibility step we have $\Phi(x^f, s^f, \mu^+) \leq \tau^f$ for some suitable value of τ^f. The number of outer iterations will be $\frac{1}{\theta}\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}$. Hence the total number of iterations will not exceed

$$\frac{(\tau^f)^\gamma}{\theta\kappa\gamma}\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}. \tag{5.13}$$

5.4.2 Feasibility step

For the search direction in the feasibility step we use the triple $(\Delta^f x, \Delta^f y, \Delta^f s)$ that is (uniquely) defined by the following system:

$$A\Delta^f x = \nu r_b^0, \tag{5.14a}$$

$$A^T\Delta^f y + \Delta^f s = \nu r_c^0, \tag{5.14b}$$

$$s\Delta^f x + x\Delta^f s = 0. \tag{5.14c}$$

Then defining the new iterates according to

$$x^f = x + \theta\Delta^f x, \quad y^f = y + \theta\Delta^f y, \quad s^f = s + \theta\Delta^f s,$$

we have, due to (5.14a),

$$b - Ax^f = b - A\left(x + \theta\Delta^f x\right) = b - Ax - \theta\nu r_b^0 = \nu r_b^0 - \theta\nu r_b^0 = (1-\theta)\nu r_b^0 = \nu^+ r_b^0.$$

In the same way one shows that $c - A^T y^f - s^f = \nu^+ r_c^0$. Hence it remains to find θ such that x^f and s^f are positive and $\Phi(x^f, s^f, \mu^+) \leq \tau^f$. This is the hard part of the analysis of our algorithm, which we leave to the subsection below. The algorithm is presented in Algorithm 5.2. A graphical illustration is given by Figure 5.2. The straight lines in Figure 5.2 depict the central paths of the pair (P_ν) and (D_ν) and the pair (P_{ν^+}) and (D_{ν^+}). The τ-neighborhoods of the μ- and μ^+-centers are shown by the gray circles. The light gray region specifies the τ^f-neighborhood of the μ^+-center of (P_{ν^+}) and (D_{ν^+}). The feasibility step is depicted by the first arrow at the right-hand side. The other arrows depict the centering steps. The iterates are shown by the circlets.

5.4.3 Analysis of the feasibility step

The feasibility step starts with some strictly feasible triple (x, y, s) for (P_ν) and (D_ν) and $\mu = \nu\zeta^2$ such that

$$\Psi(v) \leq \tau \quad \text{with} \quad v := \sqrt{\frac{xs}{\mu}}.$$

As mentioned in Subsection 5.4.1, our goal is to find θ such that after the feasibility step, with step size θ, the iterates (x^f, y^f, s^f) lie in the τ^f-neighborhood of the μ^+-center of the new perturbed pair (P_{ν^+}) and (D_{ν^+}). This means that (x^f, y^f, s^f) are such that

$$\Psi(v^f) \leq \tau^f \quad \text{where} \quad v^f := \sqrt{\frac{x^f s^f}{\mu^+}}, \quad \mu^+ = (1-\theta)\mu.$$

Algorithm 5.2 A generic primal-dual large-update IIPM

```
Input:
   accuracy parameter $\varepsilon > 0$;
   barrier update parameter $\theta \in (0,1)$;
   initialization parameter $\zeta > 0$;
   threshold parameter $\tau = O(1)$.
begin
   $x := \zeta e$; $y := 0$; $s := \zeta e$; $\nu := 1$; $\mu := \zeta^2$;
   while $\epsilon(x,y,s) \geq \varepsilon$
      feasibility step;
         $(x,\, y,\, s) := (x,\, y,\, s) + \theta(\Delta^f x,\, \Delta^f y,\, \Delta^f s)$;
      update of $\mu$ and $\nu$:
         $\nu := (1-\theta)\,\nu$;
         $\mu := \nu\zeta^2$;
      centering steps:
      while $\Phi(x,s;\mu) > \tau$
            $(x,\, y,\, s) := (x,\, y,\, s) + \alpha(\Delta x,\, \Delta y,\, \Delta s)$;
      endwhile
   endwhile
end
```

Using the scaled search directions d_x^f and d_s^f, given by (3.5), we may write

$$x^f = x + \theta\Delta^f x = x\left(\mathbf{e} + \theta\frac{\Delta^f x}{x}\right) = x\left(\mathbf{e} + \theta\frac{d_x^f}{v}\right) = \frac{x}{v}\left(v + \theta d_x^f\right), \quad (5.15)$$

$$s^f = s + \theta\Delta^f s = s\left(\mathbf{e} + \theta\frac{\Delta^f s}{s}\right) = s\left(\mathbf{e} + \theta\frac{d_s^f}{v}\right) = \frac{s}{v}\left(v + \theta d_s^f\right). \quad (5.16)$$

This shows that x^f and s^f are positive if and only if $v + \theta d_x^f$ and $v + \theta d_s^f$ are positive. On the other hand, using (3.5), we can reformulate (5.14c) as follows:

$$x\Delta^f s + s\Delta^f x = 0 \;\Leftrightarrow\; \frac{\Delta^f s}{s} + \frac{\Delta^f x}{x} = 0 \;\Leftrightarrow\; \frac{v\Delta^f s}{s} + \frac{v\Delta^f x}{x} = 0 \;\Leftrightarrow\; d_x^f + d_s^f = 0.$$

Therefore, $d_s^f = -d_x^f$. As a consequence, x^f and s^f are positive if and only if $v \pm \theta d_x^f > 0$. Since $v > 0$ this is equivalent to $v^2 - \theta^2 (d_x^f)^2 > 0$. We conclude that x^f and s^f are positive if and only if $0 \leq \theta < \theta_{\max}$, where

$$\theta_{\max} = \frac{1}{\left\|\frac{d_x^f}{v}\right\|_\infty}.$$

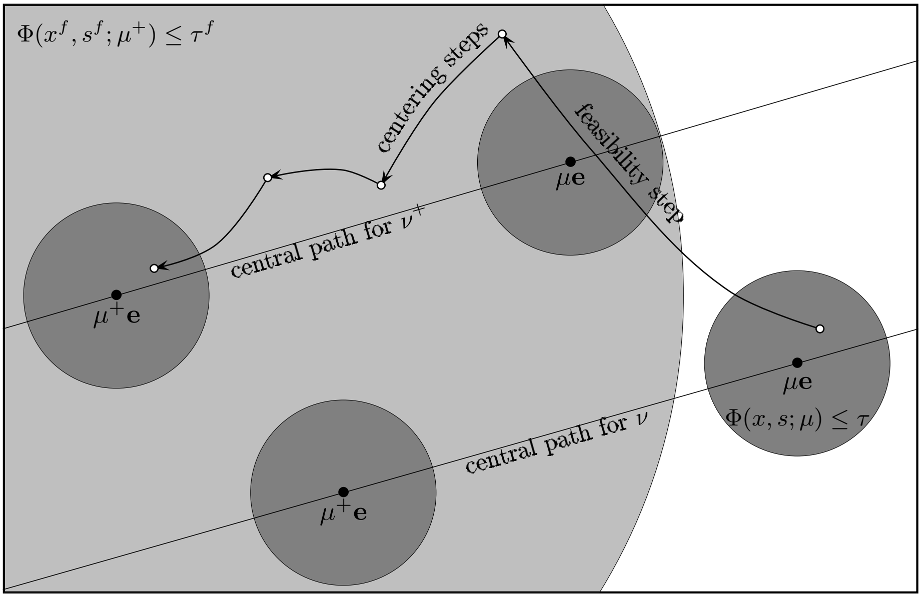


Figure 5.2: An illustration of an iteration of Algorithm 5.2.

Yet we turn to the requirement that $\Psi(v^f) \leq \tau^f$. Using (5.15), (5.16) and $xs = \mu v^2$, we write

$$\left(v^f\right)^2 = \frac{x^f s^f}{\mu^+} = \frac{\left(v + \theta d_x^f\right)\left(v - \theta d_x^f\right)}{1-\theta} = \frac{v^2 - \theta^2 (d_x^f)^2}{1-\theta}.$$

Hence, if $\theta < \theta_{\max}$ then we may write

$$v^f = \frac{\hat{v}}{\sqrt{1-\theta}}, \quad \text{where} \quad \hat{v} = \sqrt{v^2 - \theta^2 (d_x^f)^2}.$$

Lemma 5.4.1. *Let θ be such that $\frac{1}{\sqrt{1-\theta}} = O(1)$. Then $\Psi(\hat{v}) = O(n)$ implies $\Psi(v^f) = O(n)$.*

Proof. By Lemma 5.3.1 we have

$$\Psi(v^f) \leq n\psi\left(\frac{\varrho\left(\frac{\Psi(\hat{v})}{n}\right)}{\sqrt{1-\theta}}\right).$$

Let $\Psi(\hat{v}) = O(n)$. Then $\Psi(\hat{v}) \leq Cn$ for some positive constant C. Hence $\frac{\Psi(\hat{v})}{n} \leq C$. Recall that $\varrho(s) \geq 1$ for all $s \geq 0$ and $\varrho(s)$ is monotonically increasing. Also, $\psi(t)$ is monotonically increasing for $t \geq 1$. Hence we obtain

$\Psi(v^f) \leq n\psi\left(\varrho\,(C)/\sqrt{1-\theta}\right)$. Since $1/\sqrt{1-\theta} = O(1)$, the coefficient of n in the above upper bound for $\Psi(v^f)$ does not depend on n. Hence the lemma follows. □

Due to Lemma 5.4.1 it suffices for our goal to find θ such that $\Psi(\hat{v}) \leq \hat{\tau}$ where $\hat{\tau} = O(n)$. In the sequel we consider $\Psi(\hat{v})$ as a function of θ, denoted as $f_1(\theta)$. So we have

$$f_1(\theta) := \Psi(\hat{v}) = \Psi\left(\sqrt{v^2 - \theta^2 (d_x^f)^2}\right).$$

We proceed by deriving a tight upper bound for $f_1(\theta)$, thereby using similar arguments as in [10]. Since the kernel function $\psi(t)$ is eligible, $\Psi(v)$ is e-convex (cf. [10, Lemma 2.1]), whence we have

$$f_1(\theta) \leq f(\theta) := \tfrac{1}{2}\left[\Psi\left(v + \theta d_x^f\right) + \Psi\left(v - \theta d_x^f\right)\right].$$

The first and the second derivatives of $f(\theta)$ are as follows:

$$f'(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left[\psi'\left(v_i + \theta d_{x\,i}^f\right) - \psi'\left(v_i - \theta d_{x\,i}^f\right)\right] d_{x\,i}^f, \tag{5.17}$$

$$f''(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left[\psi''\left(v_i + \theta d_{x\,i}^f\right) + \psi''\left(v_i - \theta d_{x\,i}^f\right)\right] \left(d_{x\,i}^f\right)^2. \tag{5.18}$$

Since $\psi'''(t) < 0$, $\forall t > 0$, it follows that $\psi''(t)$ is monotonically decreasing. From this we deduce that

$$\psi''\left(v_i + \theta d_{x\,i}^f\right) + \psi''\left(v_i - \theta d_{x\,i}^f\right) \leq 2\psi''\left(v_i - \theta\left|d_{x\,i}^f\right|\right) \leq 2\psi''\left(v_{\min} - \theta\|d_x^f\|\right),$$

where $v_{\min} := \min(v)$ and θ small enough, i.e., such that $v_{\min} - \theta\|d_x^f\| > 0$. Substitution into (5.18) gives

$$f''(\theta) \leq \|d_x^f\|^2 \psi''\left(v_{\min} - \theta\|d_x^f\|\right).$$

By integrating both sides of this inequality with respect to θ, while using that $f'(0) = 0$, as follows from (5.17), we obtain

$$\begin{aligned} f'(\theta) &= f'(0) + \int_0^\theta f''(\xi)\, d\xi \leq \|d_x^f\|^2 \int_0^\theta \psi''\left(v_{\min} - \xi\|d_x^f\|\right) d\xi \\ &= -\|d_x^f\| \int_0^\theta \psi''\left(v_{\min} - \xi\|d_x^f\|\right) d\left(v_{\min} - \xi\|d_x^f\|\right) \\ &= \|d_x^f\|\left[\psi'\left(v_{\min}\right) - \psi'\left(v_{\min} - \theta\|d_x^f\|\right)\right]. \end{aligned}$$

Integrating once more, we get

$$\begin{aligned} f(\theta)-f(0) &= \int_0^\theta f'(\xi)\,d\xi \leq \|d_x^f\| \int_0^\theta \left[\psi'\left(v_{\min}\right)-\psi'\left(v_{\min}-\xi\|d_x^f\|\right)\right]\,d\xi \\ &= \psi'(v_{\min})\theta\|d_x^f\| + \psi\left(v_{\min}-\theta\|d_x^f\|\right)-\psi(v_{\min}) \\ &\leq \psi'(v_{\min})\theta\|d_x^f\| - \theta\|d_x^f\|\psi'(v_{\min}-\theta\|d_x^f\|) \\ &\leq \left[\psi'(v_{\min})-\psi'\left(v_{\min}-\theta\|d_x^f\|\right)\right]\theta\|d_x^f\|. \end{aligned} \tag{5.19}$$

where the last inequality holds because ψ is convex.[2]

The first derivative with respect to $v_{\min}$ of the right-hand side expression in this inequality is given by $\left(\psi''(v_{\min})-\psi''(v_{\min}-\theta\|d_x^f\|)\right)\theta\|d_x^f\|$. Since ψ'' is (strictly) decreasing, this derivative is negative. Hence it follows that the expression is decreasing in $v_{\min}$. Therefore, when θ and $\|d_x^f\|$ are fixed, the less $v_{\min}$ is, the larger the expression will be. Below we establish how small $v_{\min}$ can be when $\delta(v)$ is given.

For each coordinate v_i of v we have $\frac{1}{2}|\psi'(v_{\min})| \leq \frac{1}{2}\|\Psi(v)\| = \delta(v)$, which means that

$$-\delta(v) \leq -\tfrac{1}{2}\psi'(v_i) \leq \delta(v), \quad 1 \leq i \leq n.$$

Since the inverse function ρ of $-\frac{1}{2}\psi'$ is monotonically decreasing, this is equivalent to

$$\rho(\delta(v)) \leq v_i \leq \rho(-\delta(v)), \quad 1 \leq i \leq n. \tag{5.20}$$

Hence the smallest possible value of $v_{\min}$ is $\rho(\delta(v))$, and this value is attained in the (exceptional) case where $v_{\min}$ is the only coordinate of the vector v that differs from 1. So we may assume that $v_{\min} = \rho(\delta(v))$. This implies $-\frac{1}{2}\psi'(v_{\min}) = \delta(v)$ and hence $\psi'(v_{\min}) \leq 0$, whence $v_{\min} \leq 1$.

In the sequel we denote $\delta(v)$ simply as δ. Substitution into (5.19) gives that

$$f(\theta)-f(0) \leq \left[-2\delta-\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right)\right]\theta\|d_x^f\|.$$

Hence we certainly have $f(\theta) \leq \hat{\tau}$ if

$$f(0)+\left[-2\delta-\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right)\right]\theta\|d_x^f\| \leq \hat{\tau},$$

Since $f(0) = \Psi(v) \leq \tau$, this holds if θ is such that

$$\begin{aligned} \tau+\left[-2\delta-\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right)\right]\theta\|d_x^f\| \leq \hat{\tau} &\Leftrightarrow \\ \left[-2\delta-\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right)\right]\theta\|d_x^f\| \leq \hat{\tau}-\tau &\Leftrightarrow \\ -2\delta-\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right) \leq \tfrac{\hat{\tau}-\tau}{\theta\|d_x^f\|} &\Leftrightarrow \\ -\tfrac{1}{2}\psi'\left(\rho(\delta)-\theta\|d_x^f\|\right) \leq \delta+\tfrac{\hat{\tau}-\tau}{2\theta\|d_x^f\|}&. \end{aligned}$$

[2] We use that if f is convex and differentiable then

$$(b-a)f'(a) \leq f(b)-f(a) \leq (b-a)f'(b).$$

Since ρ is decreasing, the last inequality is equivalent to

$$\rho(\delta) - \theta\|d_x^f\| \geq \rho\left(\delta + \frac{\hat{\tau} - \tau}{2\theta\|d_x^f\|}\right). \tag{5.21}$$

Note that if θ approaches zero then the left-hand side expression converges to $\rho(\delta)$ and the right-hand side expression to zero. The left-hand side is decreasing in θ whereas the right-hand side is increasing. The largest possible θ makes both sides equal. In order to get a tight approximation for this value we first need to estimate $\|d_x^f\|$. The next lemma gives an upper bound for $\|d_x^f\|$. Its proof goes in a similar way as that of Lemma 3.3.3, given in [46].

Lemma 5.4.2. *One has*

$$\|d_x^f\| \leq \frac{\mathbf{e}^T(x+s)}{\zeta\rho(\delta)}. \tag{5.22}$$

Proof. It can be easily verified that the system (5.14), which defines the search directions $\Delta^f x$, $\Delta^f y$, and $\Delta^f s$, can be expressed in terms of the scaled search directions d_x^f and d_s^f as in (3.11) with $r_v = 0$, i.e.,

$$\bar{A}d_x^f = \nu r_b^0, \tag{5.23a}$$

$$\bar{A}^T \frac{\Delta^f y}{\mu} + d_s^f = \nu v s^{-1} r_c^0, \tag{5.23b}$$

$$d_x^f + d_s^f = 0, \tag{5.23c}$$

where

$$\bar{A} = AV^{-1}X, \quad V = \operatorname{diag}(v), \quad X = \operatorname{diag}(x).$$

From the above definition of $\bar{A}$ we deduce that $\bar{A} = \sqrt{\mu}\,AD$, where

$$D = \operatorname{diag}\left(\frac{xv^{-1}}{\sqrt{\mu}}\right) = \operatorname{diag}\left(\sqrt{\frac{x}{s}}\right) = \operatorname{diag}\left(\sqrt{\mu}\,vs^{-1}\right).$$

For the moment, let us define

$$r_b := \nu r_b^0, \quad r_c := \nu r_c^0. \tag{5.24}$$

With $\xi = -\frac{\Delta^f y}{\mu}$ we then have (by eliminating $d_s^f = -d_x^f$ from (5.23a)–(5.23c))

$$\sqrt{\mu}\,ADd_x^f = r_b, \tag{5.25}$$

$$\sqrt{\mu}\,DA^T\xi + d_x^f = -\frac{1}{\sqrt{\mu}}Dr_c, \tag{5.26}$$

By multiplying (5.26) both sides of from the left with $\sqrt{\mu}\,AD$ and using (5.25) it follows that

$$\mu AD^2A^T\xi + r_b = -AD^2r_c.$$

Therefore,

$$\xi = -\frac{1}{\mu}(AD^2A^T)^{-1}\left[AD^2r_c + r_b\right].$$

Substitution into (5.26) gives

$$\begin{aligned} d_x^f &= -\frac{1}{\sqrt{\mu}}Dr_c - \frac{1}{\sqrt{\mu}}\,DA^T(AD^2A^T)^{-1}\left[-AD^2r_c - r_b\right] \\ &= -\frac{1}{\sqrt{\mu}}\left[I - DA^T(AD^2A^T)^{-1}AD\right]Dr_c + \frac{1}{\sqrt{\mu}}\,DA^T(AD^2A^T)^{-1}r_b. \end{aligned}$$

To simplify notation we denote

$$P = DA^T(AD^2A^T)^{-1}AD.$$

Note that P is (the matrix of) the orthogonal projection to the row space of the matrix AD. We now may write

$$\sqrt{\mu}\,d_x^f = [I-P](-Dr_c) + DA^T(AD^2A^T)^{-1}r_b.$$

Let $(\bar{x},\bar{y},\bar{s})$ be such that $A\bar{x}=b$ and $A^T\bar{y}+\bar{s}=c$. Then we may write

$$\begin{aligned} r_b &= \nu r_b^0 = \nu(b - Ax^0) = \nu A(\bar{x}-x^0), \\ r_c &= \nu r_c^0 = \nu(c - A^Ty^0 - s^0) = \nu\left(A^T(\bar{y}-y^0) + \bar{s} - s^0\right). \end{aligned}$$

Thus we obtain

$$\sqrt{\mu}\,d_x^f = [I-P]\left(-\nu D\left(A^T(\bar{y}-y^0) + \bar{s} - s^0\right)\right) + \nu PD^{-1}(\bar{x}-x^0).$$

Since $I-P$ is the orthogonal projection to the null space of AD we have

$$[I-P]DA^T(\bar{y}-y^0) = 0.$$

Hence it follows that

$$\sqrt{\mu}\,d_x^f = [I-P]\left(-\nu D\left(\bar{s}-s^0\right)\right) + \nu PD^{-1}(\bar{x}-x^0).$$

To proceed we further simplify the notation by defining

$$u^x = \frac{\nu}{\sqrt{\mu}}D^{-1}(\bar{x}-x^0), \quad u^s = -\frac{\nu}{\sqrt{\mu}}D(\bar{s}-s^0). \tag{5.27}$$

Then we may write

$$d_x^f = [I-P]u^s + Pu^x.$$

Using the orthogonality of the two terms at the right-hand side, we may write

$$\|d_x^f\|^2 = \|[I-P]u^s\|^2 + \|Pu^x\|^2 \leq \|u^x\|^2 + \|u^s\|^2.$$

Due to (5.27), it follows that

$$\|d_x^f\|^2 \leq \frac{\nu^2}{\mu}\left(\left\|D^{-1}(\bar{x}-x^0)\right\|^2+\left\|D(\bar{s}-s^0)\right\|^2\right). \tag{5.28}$$

At this stage we use that the initial iterates are given by (3.1) and (3.2), so we have

$$x^0 = s^0 = \zeta\mathbf{e}, \quad y^0 = 0, \quad \mu^0 = \zeta^2, \tag{5.29}$$

where $\zeta > 0$ is such that

$$\|x^* + s^*\|_\infty \leq \zeta \tag{5.30}$$

for some optimal solutions x^* of (P) and (y^*, s^*) of (D).

We are still free to choose $\bar{x}$ and $\bar{s}$, subject to the constraints $A\bar{x} = b$ and $A^T\bar{y} + \bar{s} = c$. By choosing $\bar{x} = x^*$ and $\bar{s} = s^*$, the entries of the vectors $x^0 - \bar{x}$ and $s^0 - \bar{s}$ satisfy

$$0 \leq x^0 - \bar{x} \leq \zeta\mathbf{e}, \quad 0 \leq s^0 - \bar{s} \leq \zeta\mathbf{e}.$$

Thus it follows that

$$\begin{aligned}\|D^{-1}(\bar{x}-x^0)\|^2 + \|D(\bar{s}-s^0)\|^2 &\leq \zeta^2\left(\|D\mathbf{e}\|^2 + \left\|D^{-1}\mathbf{e}\right\|^2\right)\\ &= \zeta^2\mathbf{e}^T\left(\frac{x}{s}+\frac{s}{x}\right) = \zeta^2\mathbf{e}^T\left(\frac{x^2+s^2}{xs}\right)\\ &\leq \frac{\zeta^2\mathbf{e}^T\left(x^2+s^2\right)}{\min_i |x_i s_i|} \leq \frac{\zeta^2\left[\mathbf{e}^T(x+s)\right]^2}{\mu v_{\min}^2},\end{aligned}$$

where, as before $v_{\min} = \min(v)$. Substitution into (5.28) yields that

$$\|d_x^f\| \leq \frac{\mathbf{e}^T(x+s)}{v_{\min}\zeta},$$

where we used that $\frac{\nu^2\zeta^2}{\mu} = \nu$. Finally, since $v_{\min} \geq \rho(\delta)$, the lemma follows. □

The next lemma provides an upper bound for $\mathbf{e}^T(x+s)$.

Lemma 5.4.3. *One has*

$$\mathbf{e}^T(x+s) \leq n\zeta(1+\rho(-\delta)^2).$$

Proof. Let (x^*, y^*, s^*) be an optimal triple for (P) and (D) satisfying (5.30), and (x, y, s) a feasible triple for some perturbed pair (P_ν) and (D_ν). Setting $x^0 = s^0 = \zeta\mathbf{e}$, with ζ satisfying (5.30), (4.3), in the proof of Lemma 4.2.1, implies that

$$\left[(1-\nu)x^* - x + \nu\zeta\mathbf{e}\right]^T\left[(1-\nu)s^* - s + \nu\zeta\mathbf{e}\right] = 0.$$

Since $(x^*)^T s^* = 0$ we derive from this that

$$(1-\nu)(s^Tx^* + x^Ts^*) + \nu\zeta\mathbf{e}^T(x+s) = \nu(1-\nu)\zeta\mathbf{e}^T(x^*+s^*) + x^Ts + \nu^2\zeta^2\mathbf{e}^T\mathbf{e}.$$

Using

$$x^T s^* + s^T x^* \geq 0 \quad \text{and} \quad \zeta \mathbf{e}^T (x^* + s^*) \leq n\zeta^2,$$

one gets

$$\nu\zeta \mathbf{e}^T (x + s) \leq x^T s + n\nu\zeta^2. \tag{5.31}$$

On the other hand, one has $xs = \mu v^2$. Using $\mu = \nu\zeta^2$ and (5.20), we have

$$x^T s = \mu \mathbf{e}^T v^2 \leq n\nu\zeta^2 \rho(-\delta)^2.$$

Substitution in (5.31) implies that

$$\nu\zeta \mathbf{e}^T (x + s) \leq n\nu\zeta^2 \left(1 + \rho(-\delta)^2\right).$$

By dividing both sides of this inequality by $\nu\zeta$, the lemma follows. □

Substitution of the inequality in Lemma 5.4.3 into (5.22), we obtain

$$\|d_x^f\| \leq \frac{n(1 + \rho(-\delta)^2)}{\rho(\delta)}. \tag{5.32}$$

Yet we return to the condition (5.21) on θ:

$$\rho(\delta) \geq \theta \|d_x^f\| + \rho\left(\delta + \frac{\hat{\tau} - \tau}{2\theta \|d_x^f\|}\right).$$

The right-hand side is increasing in $\|d_x^f\|$. Therefore, due to (5.32), it suffices if

$$\rho(\delta) \geq \frac{\theta n(1 + \rho(-\delta)^2)}{\rho(\delta)} + \rho\left(\delta + \frac{\rho(\delta)\,(\hat{\tau} - \tau)}{2\theta n(1 + \rho(-\delta)^2)}\right). \tag{5.33}$$

Obviously this implies that $\theta n(1 + \rho(-\delta)^2) < \rho(\delta)^2$. Therefore, there exists $\alpha \in (0, 1)$ such that

$$\theta = \frac{\alpha \rho(\delta)^2}{n(1 + \rho(-\delta)^2)}. \tag{5.34}$$

It is clear that (5.33) can now be restated as

$$\rho(\delta) \geq \alpha\rho(\delta) + \rho\left(\delta + \frac{\hat{\tau} - \tau}{2\alpha\rho(\delta)}\right). \tag{5.35}$$

Our objective is to find the largest possible α satisfying this inequality. This requires a good understanding of the behavior of the function ρ. Figure 5.3 shows the graph of ρ (with $s \geq 0$) for several kernel functions.

In order to proceed we need bounds for $\delta = \delta(v)$ and $\rho(\delta)$. Recall that ρ is defined as the inverse function of $-\frac{1}{2}\psi'(t)$, and ϱ as the inverse function of $\psi(t)$ for $t \geq 1$. We also need the inverse function of $\psi(t)$ for $t \in (0, 1]$, which we denote

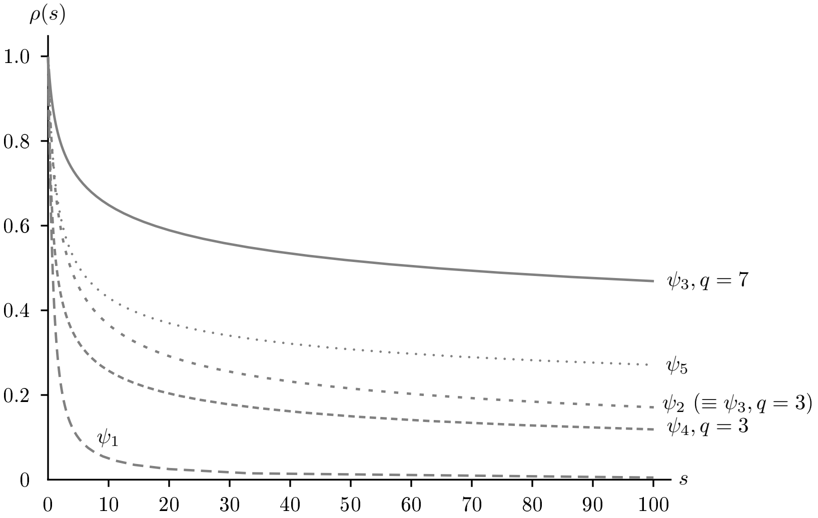


Figure 5.3: Graph of $\rho(s)$, $s \geq 0$, for several kernel functions.

as χ. To get tight estimates for these inverse functions we define the *barrier term* $\psi_b(t)$ of $\psi(t)$ by the relation

$$\psi(t) = \tfrac{t^2-1}{2} + \psi_b(t), \qquad t > 0. \tag{5.36}$$

It can be easily verified that for all kernel functions in Table 5.1 one has

$$\psi_b'(t) < 0, \qquad \psi_b''(t) > 0, \qquad \psi_b'''(t) < 0, \qquad t > 0.$$

Hence $\psi_b(t)$ is monotonically decreasing and $\psi_b'(t)$ is monotonically increasing. This implies that $\psi_b(t)$ and $-\psi_b'(t)$ have inverse functions and these function are monotonically decreasing. We denote these inverse functions as $\bar{\chi}$ and $\bar{\rho}$, respectively; in most cases they can be easily computed.

From now on we restrict ourselves to the case where $\psi = \psi_3$. Then we have

$$\psi_b(t) = \frac{t^{1-q} - 1}{q - 1} \quad \text{and} \quad \psi_b'(t) = -t^{-q}.$$

From this one easily derives that

$$\bar{\chi}(s) = \frac{1}{[1 + (q-1)s]^{\frac{1}{q-1}}}, \qquad s > \frac{-1}{q-1}, \tag{5.37}$$

$$\bar{\rho}(s) = \frac{1}{s^{\frac{1}{q}}}, \qquad s > 0. \tag{5.38}$$

The next two lemmas provide upper and lower bounds for χ and ρ. In the proofs we use that $\bar{\chi}$ and $\bar{\rho}$ are decreasing.

Lemma 5.4.4. *With χ denoting the inverse function of $\psi(t)$ for $t \in (0,1]$, one has*

$$\bar{\chi}\left(s+\tfrac{1}{2}\right) \leq \chi(s) \leq \bar{\chi}\left(s\right), \quad s \geq 0.$$

Proof. Let $t \in (0,1]$. Then one has

$$t = \chi(s) \quad \Leftrightarrow \quad \psi(t) = s \quad \Leftrightarrow \quad \psi_b(t) = s + \tfrac{1-t^2}{2} \quad \Leftrightarrow \quad \chi(s) = \bar{\chi}\left(s + \tfrac{1-\chi(s)^2}{2}\right).$$

Since $\chi(s) \in (0,1]$, this implies the inequalities in the lemma. □

Lemma 5.4.5. *With $\bar{\rho}$ denoting the inverse function of $-\psi_b'(t)$ for $t > 0$, one has*

$$\bar{\rho}(1+2s) \leq \rho(s) \leq \bar{\rho}(2s), \quad s \geq 0.$$

Moreover, if $\psi = \psi_3$ then

$$2s \leq \rho(-s) \leq 2s+1, \quad s \geq 0.$$

Proof. Since $\psi'(t) = t + \psi_b'(t)$, one has

$$\begin{aligned} t = \rho(s) \Leftrightarrow -\tfrac{1}{2}\psi'(t) = s \Leftrightarrow -\psi'(t) = 2s& \\ \Leftrightarrow -\psi_b'(t) = 2s + t \Leftrightarrow \rho(s) = \bar{\rho}(2s + \rho(s)).& \end{aligned}$$

If $s \geq 0$ then $\rho(s) = t \in (0,1]$, and hence $\bar{\rho}\,(2s) \geq \rho(s) \geq \bar{\rho}\,(2s+1)$, proving the first statement in the lemma. Now let $\psi = \psi_3$. Then we have, for $s \geq 0$,

$$t = \rho(-s) \;\Leftrightarrow\; -\tfrac{1}{2}\psi'(t) = -s \;\Leftrightarrow\; 2s = t - t^{-q} \;\Leftrightarrow\; t = 2s + t^{-q}, \quad t \geq 1.$$

Since $t \geq 1$ we have $t^{-q} \in (0,1]$. Hence $t = \rho(-s)$ implies $2s \leq \rho(-s) \leq 2s+1$. This proves the lemma. □

Recall that ϱ is the inverse function of $\psi(t)$ for $t \geq 1$. The following two results are less trivial than the preceding two lemmas.

Lemma 5.4.6 (Lemma 6.2 in [10])**.** *For $s \geq 0$, one has*

$$\sqrt{1+2s} \leq \varrho(s) \leq 1 + \sqrt{2s}.$$

Lemma 5.4.7. *One has, for each $v \in \mathbb{R}^n_{++}$,*

$$\tfrac{1}{2}\psi'\left(\varrho\left(\Psi(v)\right)\right) \leq \delta(v) \leq -\tfrac{1}{2}\psi'\left(\chi\left(\Psi(v)\right)\right).$$

Proof. The left-hand side inequality in the lemma is due to [10, Theorem 4.9]. The proof of the right-hand side inequality can be obtained by slightly changing the proof of [10, Theorem 4.9] and is therefore omitted. □

The above lemmas enable us to find an upper bound for $\delta = \delta(v)$ in terms of τ. Let $\Psi(v) \leq \tau$. Then, since χ and $-\frac{1}{2}\psi'$ are decreasing, $-\frac{1}{2}\psi'\chi$ is increasing. Hence, Lemma 5.4.7 implies that

$$\delta = \delta(v) \leq -\tfrac{1}{2}\psi'\left(\chi(\tau)\right). \tag{5.39}$$

By Lemma 5.4.4 we have $\chi(\tau) \geq \bar{\chi}(\tau + \frac{1}{2})$. Using once more that $-\frac{1}{2}\psi'$ is decreasing we obtain

$$2\delta \leq -\psi'\left(\bar{\chi}(\tau + \tfrac{1}{2})\right).$$

Since $-\psi'(t) = t^{-q} - t \leq t^{-q}$, and due to (5.37), it follows that

$$\begin{aligned} 2\delta &\leq \bar{\chi}(\tau + \tfrac{1}{2})^{-q} = \left[1 + (q-1)(\tau + \tfrac{1}{2})\right]^{\frac{q}{q-1}} \\ &= \left[1 + (q-1)(\tau + \tfrac{1}{2})\right]\left[1 + (q-1)(\tau + \tfrac{1}{2})\right]^{\frac{1}{q-1}} \leq \left[1 + (q-1)(\tau + \tfrac{1}{2})\right] e^{\tau + \frac{1}{2}}, \end{aligned}$$

where the last inequality is due to $(1 + ax)^{\frac{1}{x}} \leq e^a$ for $x > 0$ and $1 + ax > 0$. Hence, when taking $\tau \leq \frac{1}{2}$, we have

$$\delta \leq \tfrac{1}{2}qe. \tag{5.40}$$

Since ρ is decreasing, by applying ρ to both sides of (5.39), and using Lemma 5.4.4 and (5.37) we obtain

$$\rho(\delta) \geq \chi(\tau) \geq \bar{\chi}(\tau + \tfrac{1}{2}) = \frac{1}{\left[1 + (q-1)(\tau + \frac{1}{2})\right]^{\frac{1}{q-1}}}.$$

If $\tau \leq \frac{1}{2}$ this implies

$$\rho(\delta) \geq \frac{1}{q^{\frac{1}{q-1}}} = \frac{1}{e^{\frac{\log q}{q-1}}} \geq \frac{1}{e}. \tag{5.41}$$

Using that ρ is decreasing and also Lemma 5.4.5 and (5.38) we have

$$\begin{aligned} \rho\left(\delta + \frac{\hat{\tau} - \tau}{2\alpha\rho(\delta)}\right) &\leq \rho\left(\frac{\hat{\tau} - \tau}{2\alpha\rho(\delta)}\right) \leq \rho\left(\frac{\hat{\tau} - \tau}{2\rho(\delta)}\right) \\ &\leq \bar{\rho}\left(\frac{\hat{\tau} - \tau}{\rho(\delta)}\right) = \left(\frac{\rho(\delta)}{\hat{\tau} - \tau}\right)^{\frac{1}{q}} \leq \frac{1}{(\hat{\tau} - \tau)^{\frac{1}{q}}}. \end{aligned}$$

Also using $\rho \geq 1/e$ we conclude that (5.35) certainly will hold if

$$\frac{1 - \alpha}{e} \geq \frac{1}{(\hat{\tau} - \tau)^{\frac{1}{q}}}.$$

Now taking $\alpha = \frac{1}{2}$ it follows that (5.35) will be satisfied if

$$q \leq \frac{\log(\hat{\tau} - \tau)}{\log(2e)}. \tag{5.42}$$

Substitution of $\alpha = \frac{1}{2}$ into (5.34) yields

$$\theta = \frac{\rho(\delta)^2}{2n(1 + \rho(-\delta)^2)}.$$

Due to (5.41) we have $\rho(\delta) \geq 1/e$ and due to Lemma 5.4.5 and (5.40), $\rho(-\delta) \leq 2\delta + 1 \leq 1 + qe$. We therefore may conclude that (5.33) certainly holds if we take

$$\theta = \frac{1}{2e^2 n(1 + (1 + qe)^2)}. \tag{5.43}$$

This is the value that will be used in the sequel. As a consequence, the number of outer iterations is bounded above by

$$\left\lceil 2e^2 n \left(1 + (1 + qe)^2\right) \log \frac{\epsilon(\zeta \mathbf{e}, 0, \zeta \mathbf{e})}{\varepsilon} \right\rceil.$$

We finally have to estimate the number of iterations that are needed to enter the τ-neighborhood of the μ^+-center of (P_ν) and (D_ν). For that we need the parameters κ and γ for ψ_3. These parameters were obtained by Bai et al. [10] in the analysis of a large-update FIPM based on ψ_3. To make this thesis self-supporting, we calculate these parameters in the next subsection.

5.4.4 The parameters κ and γ for $\psi = \psi_3$

According to (5.10) the parameters κ and γ should be such that

$$s^{\gamma - 1} \frac{\left(\psi'\left(\varrho\left(s\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(s\right)\right)\right)\right)} \geq \kappa, \quad \forall s \geq \tau.$$

Note that $\varrho(s)$ is increasing in s and, by Lemma 5.4.6, $\sqrt{1+2s} \leq \varrho(s) \leq 1 + \sqrt{2s}$, for $s \geq 0$. For the moment, let

$$\xi := \psi'\left(\varrho\left(s\right)\right).$$

We proceed by deriving a lower bound for ξ. Since ψ' and ϱ are increasing, by using Lemma 5.4.6 we obtain

$$\begin{aligned} \xi = \psi'\left(\varrho\left(s\right)\right) \geq \psi'\left(\sqrt{1+2s}\right) = \sqrt{1+2s} - \frac{1}{\left(\sqrt{1+2s}\right)^q} \\ \geq \sqrt{1+2s} - \frac{1}{\sqrt{1+2s}} = \frac{2s}{\sqrt{1+2s}}. \end{aligned}$$

Using that $x/(1+x)$ is increasing in x and $s \geq \tau$, we may write

$$\xi \geq \sqrt{\frac{2s}{1+2s}}\,\sqrt{2s} \geq \check{\tau}\sqrt{2s} \geq \check{\tau}\sqrt{2\tau}, \qquad \text{where } \check{\tau} = \sqrt{\frac{2\tau}{1+2\tau}}. \tag{5.44}$$

Since ψ'' is decreasing, using Lemma 5.4.5 we may write

$$\begin{aligned}\psi''\left(\rho\left(\psi'\left(\varrho\left(s\right)\right)\right)\right) = \psi''\left(\rho\left(\xi\right)\right) \leq \psi''\left(\bar{\rho}\left(1+2\xi\right)\right)\\ = 1 + q\left[\frac{1}{\left(1+2\xi\right)^{\frac{1}{q}}}\right]^{-q-1} = 1 + q\left(1+2\xi\right)^{\frac{q+1}{q}}.\end{aligned}$$

Hence we have

$$\frac{\left(\psi'\left(\varrho\left(s\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(s\right)\right)\right)\right)} \geq \frac{\xi^2}{4\left[1+q\left(1+2\xi\right)^{\frac{q+1}{q}}\right]} = \frac{\xi^{\frac{q+1}{q}}}{4\left[1+q\left(1+2\xi\right)^{\frac{q+1}{q}}\right]}\,\xi^{\frac{q-1}{q}}$$

Since $x^a/\left[1+q(1+2x)^a\right]$ is increasing in x if $a > 0$, by using (5.44) it follows that

$$\frac{\left(\psi'\left(\varrho\left(s\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(s\right)\right)\right)\right)} \geq \frac{\left(\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}}{4\left[1+q\left(1+2\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\right]}\,\left(\check{\tau}\sqrt{2s}\right)^{\frac{q-1}{q}}.$$

This implies

$$s^{\frac{1-q}{2q}}\,\frac{\left(\psi'\left(\varrho\left(s\right)\right)\right)^2}{4\psi''\left(\rho\left(\psi'\left(\varrho\left(s\right)\right)\right)\right)} \geq \frac{\left(\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\left(\check{\tau}\sqrt{2}\right)^{\frac{q-1}{q}}}{4\left[1+q\left(1+2\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\right]}, \quad \forall s \geq \tau.$$

Thus we have shown that (5.10) holds for $\gamma = 1 - \frac{1-q}{2q} = \frac{q+1}{2q}$ and

$$\kappa = \frac{\left(\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\left(\check{\tau}\sqrt{2}\right)^{\frac{q-1}{q}}}{4\left[1+q\left(1+2\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\right]} = \frac{\tau^{\frac{q+1}{2q}}\,\check{\tau}^2}{2\left[1+q\left(1+2\check{\tau}\sqrt{2\tau}\right)^{\frac{q+1}{q}}\right]}.$$

Taking $\tau = \frac{1}{8}$ one gets $\check{\tau} = \frac{1}{\sqrt{5}}$. Also using $q > 1$ we obtain

$$\kappa = \frac{1}{2\cdot 5\cdot 4^{\frac{q+1}{2q}}\left[1+q\left(1+\frac{2}{\sqrt{5}}\frac{1}{2}\right)^{\frac{q+1}{q}}\right]} \geq \frac{1}{40\,q\left[1+\left(1+\frac{1}{\sqrt{5}}\right)^2\right]} \geq \frac{1}{124\,q}.$$

5.5 Complexity analysis

As we established in (5.13), the total number of iterations is at most

$$\frac{(\tau^f)^\gamma}{\theta\kappa\gamma}\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}.$$

We assume that $\hat{\tau}=O(n)$. Due to Lemma 5.4.1 we then also have $\tau^f=O(n)$, provided that $1/\sqrt{1-\theta}=O(1)$. Due to (5.43) the latter condition is satisfied. To simplify the presentation we use $\tau^f=n$ in the analysis below, but our results can easily be adapted to the case where $\hat{\tau}=O(n)$. Substituting $\gamma=\frac{q+1}{2q}$ and $\kappa\geq\frac{1}{124\,q}$ and θ, given by (5.43), the total number of iterations is bounded above by

$$248e^2qn(1+(1+qe)^2)n^{\frac{q+1}{2q}}\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}=O\left(q^3n\sqrt{n}\,n^{\frac{1}{2q}}\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}\right).$$

The expression $q^3n^{\frac{1}{2q}}$ is minimal if $q=\frac{\log n}{6}$ and then it is equal to $e^4(\log n)^3/512$. This value of q satisfies (5.42), since $\log(2e)\leq 6$. Hence we obtain the following iteration bound:

$$O\left(n\sqrt{n}\,(\log n)^3\log\frac{\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})}{\varepsilon}\right). \tag{5.45}$$

5.6 Detecting infeasibility or unboundedness

The algorithm, described in this chapter, will detect infeasibility or/and unboundedness of (P) and (D) if no optimal solutions exist. In that case Lemma 3.1.1 implies the existence of $\bar{\nu}>0$ such that the perturbed pair (P$_\nu$) and (D$_\nu$) satisfy the IPC if and only if $\nu\in(\bar{\nu},1]$. As long as $\nu^+=(1-\theta)\nu>\bar{\nu}$ the algorithm will run as it should, with θ given by (5.43). However, if ε is small enough, at some stage it will happen that $\nu>\bar{\nu}\geq\nu^+$. At this stage the new perturbed pair does not satisfy the IPC. This will reveal itself since at that time we necessarily have $\theta_{\max}<\tilde{\theta}$. If this happens we may conclude that there is no optimal pair (x^*,s^*) satisfying $\|x^*+s^*\|_\infty\leq\zeta$. In order to settle uncertainty about existence of optimal solutions for some larger values of ζ, or infeasibility and/or unboundedness, one may follow the procedure described in Subsection 3.5.

As mentioned in Section 5.1, Salahi et al. [101] presented a large-update IIPM based on ψ_3, with the convergence rate $O(n\sqrt{n}\log n)$. This convergence rate is the best known convergence rate for large-update IIPMs. In the next section, we briefly describe their algorithm.

5.7 An $O(n\sqrt{n}\log n)$ large-update IIPM

Salahi et al.'s algorithm [101] uses ψ_3 with $q = 1 + \log n$. Each iteration of this algorithm starts at a triple (x, y, s) which satisfies

$$\Phi(x, s; \mu_g) \le \eta(n, \lambda), \tag{5.46}$$

with μ_g given by (1.13) and

$$\eta(n, \lambda) := \frac{\left(\lambda^{\frac{\log n}{2}} - 1\right) n}{\log n}, \quad \text{for a} \quad \lambda \ge 2.$$

The Newton steps are obtained from the system

$$\begin{aligned} A\Delta x &= r_b, \\ A^T \Delta y + \Delta s &= r_c, \\ s\Delta x + x\Delta s &= -\mu_t v_t \nabla\Psi(v_t), \end{aligned}$$

with

$$v_t := \sqrt{\frac{xs}{\mu_t}}$$

where μ_t is the smallest root of the equation

$$\Phi(x, s, \mu) = \frac{(\lambda - 1)n}{2}.$$

It was shown (see [101, Lemma 2.5]) that if (5.46) holds then

$$\lambda \le \frac{\mu_g}{\mu_t} \le \lambda + \tfrac{2}{\log n}. \tag{5.47}$$

They derive a step size $\alpha \in (0, 1)$ for which the new iterates (x^+, y^+, s^+) and μ_g^+, with

$$\mu_g^+ := \frac{\left(x^+\right)^T s^+}{n},$$

satisfy (5.46), and

$$\Phi(x^+, s^+, \mu_t^+) = \frac{(\lambda - 1)n}{2},$$

where $\mu_t^+ := (1 - \theta)\mu_t$ for $\theta = O(\frac{1}{n\sqrt{n}\log n})$. (5.47) implies that μ_g is also reduced by a factor $1 - \theta$. The residual norms are bounded above by a constant multiplier of the mean value μ_g. Thus, the infeasibility and the duality gap are reduced by a factor $1 - \theta$. This imposes the convergence rate $O(n\sqrt{n}\log n)$ to the algorithm.

Unfortunately, the theoretical convergence rate of our algorithm is a factor $(\log n)^2$ worse than Salahi et al.'s algorithm. As the theoretical result is disappointing, in order to show that our algorithm is a large-update algorithm we rely on the numerical results which are presented in Chapter 6. Note that, theoretically speaking, both algorithms, i.e., Salahi et al.'s [101] and our algorithm, suffer from the deficiency that they are small-update methods as they use the barrier updating parameter $\theta = O(1/(n\sqrt{n}\log n))$ and $\theta = O(1/(n\sqrt{n}(\log n)^2))$, respectively.

6

Implementation: issues and results

So far, we have been dealing with the theoretical aspects of IIPMs for LO, i.e., global convergence and/or polynomiality. Although they are interesting by themselves, from a commercial point of view, the practical behavior of these methods may be even more interesting. There are some (numerical) issues that theory-oriented papers never discuss, despite the fact that they may be critical for a practically efficient implementation. Section 6.1 is devoted to a brief description of some of these issues. The most efficient implementation of IIPMs is the LIPSOL package which is based on Mehrotra's PC algorithm [74]. We compare the iteration numbers of our algorithm with those of this package. The LIPSOL package is described in Section 6.2.

6.1 Implementation of IIPMs: issues

As mentioned in Chapter 2, an implementation of the "big M" method to solve (P) and (D) was given by McShanne, Monma and Shanno [71]. Although this method was more efficient than the simplex method, it suffered from some numerical instabilities caused by huge coefficients. Lustig [62] designed his algorithm to overcome these issues. He ran his algorithm to solve the same set of NETLIB[1] problems as chosen by McShane et al. [71], i.e., those with no explicit upper bounded or free variables. Lustig's algorithm outperformed that of McShane et al. [71] in terms of iteration number and the simplex method in terms of CPU time. It is worth mentioning that Lustig used some sophisticated analytic approach to calculate the barrier parameter and the step sizes along the Newton directions while McShane et al. used a heuristic barrier parameter and a large fraction, e.g., 0.9995, of maximal step sizes.

Most of the computational efforts in IPMs for LO are devoted to the *Cholesky*

[1] http://www.netlib.org/lp/

factorization of the coefficient matrix in a linear system of the form

$$AD^2A^T\Delta y = \hat{b}, \tag{6.1}$$

where $\hat{b}$ is some right-hand side and D, in the LO case, has the form

$$D := \operatorname{diag}\left(\sqrt{\frac{x}{s}}\right).$$

McShane et al. [71] and Lustig [62] used, respectively, the linear algebra packages SMPAK and SPARSPAK [36] to solve the system (6.1). If there were sparse columns in the coefficient matrix, Choi, Monma and Shanno [17] applied the *Schur complement approach* to split sparse and dense columns. This led to better performance than the past implementations. The idea of using the Schur complement approach had been first suggested by Gill et al. [38]. In addition to the problems tested in [62, 71], Choi et al. considered NETLIB problems with bounded and/or free variables as well. Moreover, motivated by the fact that the heuristic values of the barrier parameter and the step sizes, applied by McShane et al., proved to be efficient, Choi et al. adapted Lustig's algorithm in such a way that it avoids the sophisticated method of Lustig in calculating these parameters and, instead, applies the heuristic values of McShane et al..

Variants of Lustig's algorithm were pursued by Lustig, Marsten and Shanno [63]. They considered the variant of Lustig's algorithm studied by McShane et al. [71] and Choi et al. [17], with minor modification with respect to the choice of the barrier parameter. Moreover, Lustig et al. [63] presented some further discussion on a variety of computational issues in the primal-dual implementation and the barrier methods in general, along with a new comprehensive implementation of the primal-dual algorithm for the entire NETLIB test set. Strengths and weaknesses of the Schur complement approach, suggested by Choi et al. [17], were also discussed. Moreover, Lustig et al. [63] showed that the limiting search directions obtained by Lustig [62] were nothing but the Newton directions for the system (1.5) which serve to reduce the infeasibility. They also discussed the role of the barrier parameter in more detail.

Other variants of Lustig's algorithm were also presented by Subramanian, Subramanian, Saltzman, Lustig and Shanno [69] and Tanabe [105].

The most efficient primal-dual IIPM, on which most of the existing IPM codes are based, e.g., LIPSOL of Zhang [120] or PCx package of Czyzyk et al. [18], is Mehrotra's PC approach [74]. This algorithm enjoys the feature of using a second-order approximation of the central path [74, 115]. It should be noted that the idea of using higher order approximation of the central path was proposed first by Megiddo [72] and further studied by Monteiro et al. [81]. Mehrotra [74] just combined these two existing ideas in a nice way in the algorithm of Lustig et al. [63]. He applied the new algorithm to solve a subset of the NETLIB problems which have no bounded or free variables. This reduced the number of iterations significantly, when compared to previous algorithms, e.g., [63, 82].

Another implementation of Mehrotra's algorithm was done by Lustig, Marsten and Shanno [64], incorporating the entire NETLIB test set. It was established that for large and more complex problems Mehrotra's method is more efficient. Numerical issues concerned with the Schur complement approach along with some other numerical problems were also discussed. Besides, some numerical remedies for these issues were presented.

After the release of a globally convergent variant of Lustig's algorithm by Kojima et al. [54], Lustig, Marsten and Shanno [65] designed a practically globally convergent variant of Mehrotra's PC method.

Another successful implementation of Mehrotra's PC algorithm was given by Lustig, Marsten and Shanno [64].

The most successful implementation of Mehrotra's PC algorithm is the LIPSOL package, which was developed by Zhang [120]. Because LIPSOL is currently the leading IPM-based package for solving LO problems, we devote the next section to a detailed description of Mehrotra's method.

6.2 Mehrotra's PC approach: LIPSOL package

In this section we deal with a variant of Mehrotra's PC method which runs in LIPSOL package. In addition to its being on top of all IPM implementations, our motivation to describe this package in detail is that in our experiments, presented in Section 6.3, we use some heuristics which are inspired by those of this package.

6.2.1 About LIPSOL

LIPSOL is a free software[2] based on MATLAB. Programming in MATLAB is very easy and it has a high level technical computing environment for computations. Matrices and formulas can be expressed in a way very close to what we write mathematically. Besides, because MATLAB provides external interface facilities to enable interaction with programs in Fortran and C languages in the form of *mex-files*, some routine tasks like matrix and vector operations are done using MATLAB functions while computationally intensive tasks like Cholesky factorization are done using Fortran codes. As a result, one enjoys the facility of simple programming in MATLAB and the computational speed of Fortran. In addition, LIPSOL uses some new tricks to overcome numerical instabilities in the Cholesky factorization of the coefficient matrix of the linear system (6.1). Considering the fact that it more often happens that the coefficient matrix of the linear system (6.1) is close to positive semidefinite not definite, the regular Cholesky factorization may be unstable. To overcome this issue, Zhang suggests the use of so-called *Cholesky-infinity factorization*. In this procedure, once a diagonal pivot in the Cholesky factorization is found to be zero, the corresponding diagonal element of

[2] `http://www.caam.rice.edu/~zhang/lipsol/`

$\mathbb{L}$, with $\mathbb{L}$ the Cholesky factor, is set to infinity. He had to write an external Fortran code for this new procedure because access to MATLAB's built-in function `chol` was impossible.

There is another version of LIPSOL which is used as a subroutine of the MATLAB command `linprog` to solve large-scale problems. If the problem is not tagged as 'Large-Scale', `linprog` uses the simplex method as a default algorithm, and otherwise the function `lipsol`. Recall that, currently, a variety of linear algebraic techniques to solve a linear system, e.g., column approximate minimum degree permutation (the function `colamd`), Cholesky factorization, Cholesky-infinity factorization, etc., are all available in MATLAB. Due to this, the function `lipsol` does not use mex-files. Currently, the Cholesky-infinity factor can be easily computed by using the MATLAB function `cholinc`.

Computational experiments, e.g., [64], reveal that the larger the dimension of a problem is, the more IPMs outperform the simplex method.

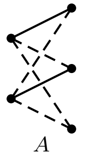
Some preprocessing is performed before the main algorithm starts to run: checking obvious infeasibility, deleting fixed variables and zero rows and columns from the matrix A, easy handling of free and bounded variables, solving singleton constraints, if any required, along with *scaling* the problem and making the matrix A *structurally full rank*[3]. Some heuristic test is carried out to check if the matrix A has some sparse columns and then by separating the sparse columns, if any, from the dense ones, the Schur complement approach is applied to solve the *normal equations* (6.1). See Subsection 6.2.2 for more detail.

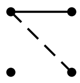
If the solution of the normal equations (6.1) is not satisfactory, i.e., the residual norm $\|AD^2A^T\Delta y - \hat{b}\|$ is too large, Y. Zhang ignores the solution and uses, instead,

[3] The structural rank of a matrix A in $\mathbb{R}^{m\times n}$ is the size of a maximum matching of the bipartite graph of A. It is actually an upper bound for the numerical rank of A. So, if $m = n$, A is structurally full rank if its bipartite graph has a perfect matching. If $m \neq n$, Davis [21] calls matrix A structurally full row (column) rank if all nodes corresponding to the rows (columns) are matched in a maximum matching of its bipartite graph (note that there is no perfect matching in this case anymore). Let us give an example. Given the matrices A and B by

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 6 & 8 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},$$

their bipartite graphs are as follows.


A


B

In both graphs, the set of solid edges is a maximum matching. For matrix A, the set of the nodes corresponding the rows are all matched in the maximum matching. Thus A is structurally full row rank but not full column rank. Based on a similar argument, the matrix B turns out to be neither full row nor full column rank. We say that the matrix B is *structurally rank deficient.* Note that for both matrices the numerical rank equals 1.

the solution of *the augmented system*, given by

$$\begin{pmatrix} -D^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_b \\ r_c - \frac{r_{xs}}{x} \end{pmatrix}, \tag{6.2}$$

with r_{xs} defined as in Subsection 6.2.3. Δs is given by

$$\Delta s = \tfrac{r_{xs}}{x} - D^{-2}\Delta x.$$

Remark 6.2.1. *It can be verified that the augmented system* (6.2) *is equivalent to the following system:*

$$\begin{pmatrix} 0 & AD^2A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} AD^2 r_b + r_c - \frac{r_{xs}}{x} \\ r_c - \frac{r_{xs}}{x} \end{pmatrix}.$$

This means that the normal equations (6.1) *can be obtained form the augmented system* (6.2) *by setting*

$$\hat{b} = AD^2 r_b + r_c - \tfrac{r_{xs}}{x}.$$

In order to solve the system (6.2), the author uses block $\mathbb{L}D\mathbb{L}^T$ factorization of the coefficient matrix. This is done by using the MATLAB function `ldl`. For advantages and disadvantages of using the augmented system instead of normal equations and an extensive discussion on implementation of IPMs and related issues we refer to [115].

Before describing the variant of Mehrotra's algorithm in LIPSOL, we first briefly explain the Schur complement approach.

6.2.2 The Schur complement approach

Consider the coefficient matrix of the linear system (6.1). Let A_s and A_d denote, respectively, sparse and dense columns of A, and D_s and D_d the corresponding subdiagonals of the scaling matrix D. By splitting the columns of A in sparse and dense columns, we write

$$A = [A_s\ A_d].$$

Then we may write

$$AD^2A^T = A_s D_s^2 A_s^T + A_d D_d^2 A_d^T.$$

Denoting $P = A_s D_s^2 A_s^T$ and $U = A_d D_d$ the latter can be rewritten as

$$AD^2A^T = P + UU^T.$$

If P is nonsingular then, by the Sherman-Morrison-Woodbury formula, we have

$$\begin{aligned}(P+UU^T)^{-1} &= P^{-1} - P^{-1}U(I+U^TP^{-1}U)^{-1}U^TP^{-1} \\ &= P^{-1}[I - U(I+U^TP^{-1}U)^{-1}U^TP^{-1}].\end{aligned}$$

Letting $P = \mathbb{L}\mathbb{L}^T$, with $\mathbb{L}$ a lower triangular matrix, Δy in (6.1) is obtained from the following system:

$$\mathbb{L}\mathbb{L}^T \Delta y = \hat{b} - U(I + W^T W)^{-1} W^T \mathbb{L}^{-1} \hat{b}, \quad \text{with} \quad W := \mathbb{L}^{-1} U.$$

Note that P is a large sparse matrix whose Cholesky decomposition is not so computationally expensive. The process of calculating Δy can be organized as follows:

- solve $\mathbb{L}W = U$;
- solve the system $\mathbb{L}\bar{b} = \hat{b}$;
- solve $(I + W^T W)\xi = W^T \bar{b}$;
- solve $\mathbb{L}\mathbb{L}^T \Delta y = \hat{b} - U\xi$.

Note that the matrix U has usually low rank, whence the matrix $I + W^T W$ is small and requires a small number of back-substitutions once P is factorized. However, the sparse portion of the coefficient matrix in (6.1) may be severely ill-conditioned or even singular, leading to some large residual norm $\|AD^2 A^T \Delta y - \hat{b}\|$. This makes the Schur complement approach useless in that case.

6.2.3 Newton search directions

Mehrotra considered the system (1.4). Without assuming feasibility of the current iterates, this system can be rewritten as follows:

$$\begin{aligned} A\Delta x &= r_b, \\ A^T \Delta y + \Delta s &= r_c, \\ s\Delta x + x\Delta s &= r_{xs}, \end{aligned} \tag{6.3}$$

with

$$r_{xs} := \mu \mathbf{e} - xs - \Delta x \Delta s.$$

Mehrotra defines the directions $(\Delta x, \Delta y, \Delta s)$ as follows:

$$(\Delta x, \Delta y, \Delta s) := (\Delta^a x, \Delta^a y, \Delta^a s) + (\Delta^{cc} x, \Delta^{cc} y, \Delta^{cc} s),$$

where $(\Delta^a x, \Delta^a y, \Delta^a s)$ are affine-scaling directions obtained from the system (6.3) after setting $r_{xs} := -xs$ and $(\Delta^{cc} x, \Delta^{cc} y, \Delta^{cc} s)$ the *combined centering-corrector* steps obtained from the system (6.3) after setting $r_b = r_c = 0$ and $r_{xs} := \mu \mathbf{e} - \Delta^a x \Delta^a s$.

The predictor step is applied to calculate the barrier parameter as explained in Subsection 6.2.4. Once the barrier parameter is obtained, Mehrotra proceeds with computation of the combined centering-corrector steps. The coefficient matrix of the combined centering-corrector directions and the affine-scaling directions are the same and thus only one matrix factorization is required.

The corrector step is motivated as follows. It can be easily verified that the affine-scaling step $(\Delta^a x, \Delta^a y, \Delta^a s)$ is obtained after linearizing of the system (1.4) with $\mu = 0$. This means that if full affine-scaling step is taken then one should have

$$(x + \Delta^a x)(s + \Delta^a s) = 0. \tag{6.4}$$

On the other hand, one has

$$(x + \Delta^a x)(s + \Delta^a s) = xs + x\Delta^a s + s\Delta^a x + \Delta^a x \Delta^a s = \Delta^a x \Delta^a s,$$

meaning that (6.4) may not hold in general. The corrector step tries to compensate for this deviation from the linearity, modifying the search directions so that the pairwise products come closer to zero.

6.2.4 Choice of the barrier parameter

After computation of the affine-scaling directions $(\Delta^a x, \Delta^a y, \Delta^a s)$, Mehrotra calculates $\bar{\mu}$ as follows:

$$\bar{\mu} := \frac{(x + \alpha^p_{\max} \Delta^a x)^T (s + \alpha^d_{\max} \Delta^a s)}{n}, \tag{6.5}$$

where

$$\begin{aligned} \alpha^p_{\max} &:= \min\left\{1, \max\left\{\alpha : \ x + \alpha \Delta^a x \geq 0\right\}\right\}, \\ \alpha^d_{\max} &:= \min\left\{1, \max\left\{\alpha : \ s + \alpha \Delta^a s \geq 0\right\}\right\}. \end{aligned} \tag{6.6}$$

The quantity $\bar{\mu}$ serves to estimate the efficiency of the predictor step as follows. If $\bar{\mu} \ll \mu$ then it means that the predictor step makes a large amount of reduction on the duality gap. If $\bar{\mu}$ is smaller than but close to μ, then it means that the predictor step does not make significant reduction on the duality gap, and thus a small reduction on μ is allowed. Mehrotra suggests the following heuristic which has been efficient in computational experiments [74]:

$$\mu^+ = \left(\frac{\bar{\mu}}{\mu}\right)^2 \mu.$$

Setting the new value of μ to μ^+, the combined centering-corrector Newton step $(\Delta^{cc} x, \Delta^{cc} y, \Delta^{cc} s)$ is computed.

The Newton directions $(\Delta x, \Delta y, \Delta s)$ are given by

$$(\Delta x, \Delta y, \Delta s) := (\Delta^a x, \Delta^a y, \Delta^a s) + (\Delta^{cc} x, \Delta^{cc} y, \Delta^{cc} s).$$

After that Zhang [120] (LIPSOL) proposes to obtain the step sizes α_p and α_d in, respectively, the primal and the dual spaces such that the new iterates (x^+, y^+, s^+), defined as

$$x^+ := x + \alpha_p \Delta x, \quad y^+ := y + \alpha_d \Delta y \quad \text{and} \quad s^+ := s + \alpha_d \Delta s,$$

satisfy

$$x^+s^+ \geq \gamma \tfrac{x^{+T}s^+}{n} \quad \text{with} \quad \gamma = 10^{-5}.$$

6.2.5 Stopping criteria

The algorithm stops if the following stopping criteria holds:

$$E(x,y,s) := \frac{\|r_b\|}{\max(1,\|b\|)} + \frac{\|r_c\|}{\max(1,\|c\|)} + \frac{|c^Tx - b^Ty|}{\max(1,|c^Tx|,|b^Ty|)} \leq \varepsilon, \tag{6.7}$$

for a predetermined $\varepsilon > 0$. See Algorithm 6.1.

Algorithm 6.1 The PC algorithm of Mehrotra [74]

```
Input:
   accuracy parameter: $\varepsilon > 0$;
begin
   initial points: $x^0 > 0$, $y^0$ and $s^0 > 0$  $\mu = \mu^0 := (x^0)^T s^0/n$;
   while $E(x,y,s) > \varepsilon$
      calculate $(\Delta^a x, \Delta^a y, \Delta^a s)$ and $\alpha^p_{\max}$ and $\alpha^d_{\max}$;
      $\mu$-update: $\mu := (\bar{\mu}/\mu)^2 \mu$ with $\bar{\mu}$ given by (6.5);
      $x : = x + \alpha_p(\Delta^a x + \Delta^{cc} x)$;
      $y : = y + \alpha_d(\Delta^a y + \Delta^{cc} y)$;
      $s : = s + \alpha_d(\Delta^a s + \Delta^{cc} s)$;
   endwhile
end
```

Mehrotra did not consider convergence or polynomiality of his algorithm. However, Y. Zhang and D. Zhang [121] proposed some variant of this algorithm that is polynomial-time. Their proof uses the potential function suggested by Mehrotra [74] and the recipe of the polynomiality proof given by Zhang [119].

6.3 Implementation of our large-update IIPM

In this section, we present the numerical results of the algorithm described in Chapter 5. Theoretically, the barrier parameter μ is updated by a factor $(1-\theta)$ with θ given by (5.43), and the iterates are kept very close to the μ-centers, namely the τ-neighborhood of the μ-centers, with $\tau = \frac{1}{8}$. In practice, it is not efficient to do so and not necessary either. We present a variant of the algorithm which uses a predictor-corrector step in the feasibility step. Moreover, for the parameter τ, defined in Section 5.4.1, we allow some larger value than $\frac{1}{8}$, e.g., $\tau = O(n)$. We set $\tau = \hat{\tau} = O(n)$ with $\hat{\tau}$ defined as in Section 5.4.1. As a consequence, the algorithm does not need centering steps. We choose $\hat{\tau}$ according to the following heuristics: if $n \leq 500$, then $\hat{\tau} = 100n$, for $500 \leq n \leq 5000$, we choose $\hat{\tau} = 10n$ and for $n \geq 5000$, we set $\hat{\tau} = 3n$. We compare the performance of the algorithm with the well-known LIPSOL package [120].

6.3.1 Starting point

A critical issue when implementing a primal-dual method is to find a suitable starting point. It seems sensible to look for a starting point which is well-centered and as close to a feasible primal-dual point as possible. The one suggested by theory, i.e., given by (2.39), being nicely centered, may be quite far from the feasibility region. Moreover, to find a suitable ζ is another issue.

In our implementation, we use a starting point which is proposed by Lustig et al. [64] and inspired by the starting point used by Mehrtora [74]. It uses the least squares solution of the system of constraints in (P), namely,

$$\tilde{x} = A^T {(AA^T)}^{-1} b.$$

As in [64], we define

$$\xi_1 = \max \left(- \min_{1 \leq j \leq n} \tilde{x}_j, 100, \frac{\|b\|_1}{100} \right) \quad \text{and} \quad \xi_2 = 1 + \|c\|_1 .$$

Then for $j = 1, \cdots, n$, assign

$$\bar{x}_j = \max \left(\tilde{x}_j, \xi_1 \right).$$

Now putting, for $j = 1, \cdots, n$,

$$\bar{s}_j = \begin{cases} c_j + \xi_2 & \text{if} \quad c_j > \xi_2 \\ -c_j & \text{if} \quad c_j < -\xi_2 \\ c_j + \xi_2 & \text{if} \quad 0 \leq c_j < \xi_2 \\ \xi_2 & \text{if} \quad -\xi_2 \leq c_j \leq 0. \end{cases}$$

we set

$$x^0 = \bar{x}, \qquad y^0 = 0, \qquad s^0 = \bar{s} \quad \text{and} \quad \mu^0 := \frac{{(x^0)}^T s^0}{n}.$$

Since we are interested in a point which is in the τ-neighborhood of the μ^0-center, as long as $\Phi(x^0, s^0, \mu^0) > \tau$, we keep increasing μ^0 by a constant factor, say 1.1. In that way, we obtain a barrier parameter μ^0 for which $\Phi(x^0, s^0, \mu^0) \leq \tau$.

6.3.2 Feasibility step size

As in other efficient numerical experiments, e.g., [64, 120], regardless of the theoretical result, we apply different step sizes along the primal step Δx and the dual step $(\Delta y, \Delta s)$. This implies that the feasibility improves much faster than when identical step sizes are used. Letting (x, y, s) be the current iterates and $(\Delta x, \Delta y, \Delta s)$ the Newton step, we obtain the maximum step sizes $\theta^p_{\max}$ and $\theta^d_{\max}$ in, respectively, the primal and the dual spaces as follows:

$$\theta^p_{\max} = \min_{\Delta x_i < 0} \left\{ 1, -\frac{x_i}{\Delta x_i} \right\}, \quad \theta^d_{\max} = \min_{\Delta s_i < 0} \left\{ 1, -\frac{s_i}{\Delta s_i} \right\}.$$

The goal is to keep the iterates close to the μ-center, i.e., in its $\hat{\tau}$-neighborhood where $\hat{\tau}$ is defined in Subsection 5.4.3. Thus, letting $\bar{\theta}$ be such that

$$\Phi(x + \bar{\theta}\theta^p_{\max}\Delta x, s + \bar{\theta}\theta^d_{\max}\Delta s, \mu) \leq \hat{\tau},$$

the primal and the dual step sizes θ_p and θ_d are defined as follows:

$$\theta_p = \bar{\theta}\theta^p_{\max} \quad \text{and} \quad \theta_d = \bar{\theta}\theta^d_{\max}.$$

6.3.3 Stopping criteria

As in LIPSOL, our algorithm terminates if the condition (6.7) or the following condition is met:

$$|x^T s - {x^+}^T s^+| < \varepsilon,$$

where $\varepsilon = 10^{-6}$. The condition (6.7) measures the total relative errors in the optimality conditions (1.1) whilst the latter criterion terminates the program if only a tiny improvement is obtained on the optimality. In fact, it prevents the program from stalling. We include this criterion following Lustig [62].

6.3.4 Solving the linear system

Unlike LIPSOL which uses the Schur complement approach to solve the linear system, we simply apply the backslash command of MATLAB ('\') to solve the normal equations (6.1). Denoting $M := AD^2A^T$ in (6.1), whenever the multiplier matrix M is ill-conditioned, we could obtain some more accurate solution by perturbing M as

$$M = M + 10^{-9}I,$$

where I is the identity matrix with size of M.

6.3.5 An iteration of the algorithm

Motivated by the numerical results, and considering the fact that Mehrotra's PC method has become the most efficient in practice and used in most IPM-based software packages, e.g., [3, 18, 112, 120], we present the numerical results of the variant of our algorithm which uses Mehrotra's PC direction at the feasibility step.

At the feasibility step, we apply the system

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

to obtain the affine-scaling directions $(\Delta^a x, \Delta^a y, \Delta^a s)$. Then, the maximum step sizes $\theta^p_{\max}$ and $\theta^d_{\max}$ in, respectively, primal and dual spaces are calculated as described in Subsection 6.3.2. Then defining

$$\mu^a = \frac{(x + \theta^p_{\max}\Delta^a x)^T (s + \theta^d_{\max}\Delta^a s)}{n},$$

we let

$$\sigma = \left(\bar{\sigma}\frac{\mu^a}{\mu}\right)^3, \quad \bar{\sigma} \in (0,1).$$

We use $\bar{\sigma} = 0.3$ as the default value of $\bar{\sigma}$. If $\sigma < 1$, we calculate the new barrier update parameter μ as follows:

$$\mu_{\text{new}} = \sigma\mu^a.$$

Then, if necessary, by increasing μ_{new} by a constant factor, say 1.1, we derive some μ_{new} for which

$$\Phi(x, s, \mu_{\text{new}}) \leq \hat{\tau}.$$

The ideal case occurs when $\mu_{\text{new}} < \mu$. Because then by setting $\mu = \mu_{\text{new}}$, the corrector step $(\Delta^c x, \Delta^c y, \Delta^c s)$, obtained from

$$\begin{aligned} A\Delta^c x &= 0, \\ A^T\Delta^c y + \Delta^c s &= 0, \\ s\Delta^c x + x\Delta^c s &= \mu\mathbf{e} - \Delta^a x\Delta^a s, \end{aligned} \tag{6.8}$$

yields an improvement of the duality gap. If $\mu_{\text{new}} \geq \mu$, then the use of the system (6.8) with $\mu = \mu_{\text{new}}$ gives rise to an increase or no improvement of the duality gap. Hence the use of $\mu = \mu_{\text{new}}$ is no longer sensible in this case. Recall that if $\sigma \geq 1$ then it means that the duality gap was increased after the affine-scaling step $(\Delta^a x, \Delta^a y, \Delta^a s)$. Thus a μ-update makes no sense in this case either.

If $\sigma < 1$ and $\mu_{\text{new}} < \mu$, we use the system (6.8) with $\mu = 0$ as a corrector step.

The feasibility step $(\Delta^f x, \Delta^f y, \Delta^f s)$ is obtained as follows:

$$\Delta^f x = \Delta^a x + \Delta^c x, \quad \Delta^f y = \Delta^a y + \Delta^c y, \quad \Delta^f s = \Delta^a s + \Delta^c s.$$

Next, we calculate the primal and the dual step sizes θ_p and θ_d, as described in Subsection 6.3.2, and then obtain the new iterates (x^f, y^f, s^f) as follows:

$$x^f = x + \theta_p \Delta^f x, \quad y^f = y + \theta_d \Delta^f y \quad \text{and} \quad s^f = s + \theta_d \Delta^f s.$$

6.3.6 Results

In this section, we present our numerical results. Motivated by the theoretical results, which say that the kernel function ψ_3 gives the best known theoretical iteration bound for large-update IIPMs, we compare the performance of the algorithm described in the previous subsection based on both the logarithmic barrier function and the ψ_3-based barrier function. As the theory suggests, we use $q = \frac{\log n}{6}$ in ψ_3.

Our test was done on a standard PC with Intel® Core™ 2 Duo CPU and 3.25 GB of RAM. The code was implemented by version 7.11.0 (R2010b) of MATLAB® on a Windows XP Professional operating system. The problems chosen for our test are from the NETLIB set. To simplify the study, we chose the problems which have the following format:

$$\min \left\{ c^T x : \quad Ax = b, \ x \geq 0 \right\};$$

i.e., there is no nonzero lower bound or finite upper bound on the decision variables. These problems are listed in Tables 6.3 and 6.4.

We perform the following preprocessing before the main algorithm starts to run which are in common with LIPSOL: checking obvious infeasibility, deleting fixed variables and zero rows and columns from the matrix A and solving singleton constraints, if any required. As mentioned in Section 6.2, in addition to these actions, in LIPSOL, Zhang scales the problem and makes the matrix A structurally full rank. We disabled both of these phases when running LIPSOL.

Numerical results are presented in Tables 6.1 and 6.2. In the second and the fourth columns, we listed the total number of iterations of the algorithm based on, respectively, ψ_1, the kernel function of the logarithmic barrier function, and ψ_3. The third and fifth columns contain the quantity $E(x, y, s)$. The iteration numbers of the LIPSOL package are given in the sixth column of these tables, and the seventh column lists the quantity $E(x, y, s)$ of the LIPSOL package. In each row, the dark gray cell denotes the smallest of the iteration numbers of the three algorithms, and the bold number denotes the smallest of the iteration numbers of the ψ_1-based and the ψ_3-based algorithms.

As it can be noticed from the last row of the table, the overall performance of the algorithm based on ψ_1 is much better than that the variant based on ψ_3. However, in some of the problems, the ψ_3-based algorithm outperforms the ψ_1-based algorithm. This happens for the problems AGG, BANDM, DEGEN2, DEGEN3, SCSD1, SCSD6, SCSD8 and SHARE2B. Obviously, LIPSOL is still the champion; though, our ψ_1-based algorithm saves one iteration compared with LIPSOL for the problems AGG2 and AGG3, and two iterations for STOCFOR1.

Problem	ψ_1		ψ_3		LIPSOL	
	it.	$E(x,y,s)$	it.	$E(x,y,s)$	it.	$E(x,y,s)$
25FV47	**26**	1.8E-007	32	1.1E-007	25	2.8E-007
ADLITTLE	12	6.8E-008	12	1.3E-007	11	2.4E-011
AFIRO	8	1.0E-007	8	8.6E-008	7	3.7E-009
AGG	17	8.8E-007	19	2.7E-007	18	1.1E-008
AGG2	17	9.5E-007	18	2.7E-007	18	2.6E-010
AGG3	18	3.0E-007	18	6.0E-007	16	6.2E-008
BANDM	20	2.6E-007	**18**	8.5E-007	16	3.6E-007
BEACONFD	11	1.1E-007	11	5.9E-007	11	1.2E-010
BLEND	13	6.2E-007	13	1.7E-008	12	5.7E-011
BNL1	**32**	5.0E-007	34	2.2E-007	25	5.3E-008
BNL2	**33**	4.1E-007	35	5.6E-007	31	1.3E-007
BRANDY	**19**	2.5E-007	20	4.3E-007	18	2.0E-008
D2Q06C	28	5.6E-001	45	1.3E-007	28	4.8E-007
DEGEN2	25	1.3E-004	**16**	2.8E-005	13	4.2E-007
DEGEN3	23	1.4E-004	**21**	5.3E-004	19	1.4E-007
E226	22	7.4E-007	22	8.3E-008	20	8.9E-007
FFFFF800	26	1.0E-006	27	3.5E-006	26	3.0E-007
ISRAEL	**22**	2.0E-007	23	2.2E-007	20	2.2E-007
LOTFI	**16**	3.7E-007	18	6.8E-006	15	4.6E-008
MAROS-R7	19	8.0E-007	19	2.4E-008	14	1.0E-009
SC105	10	1.7E-008	10	1.5E-008	9	4.2E-008
SC205	11	2.5E-007	12	3.6E-008	11	6.5E-009
SC50A	9	1.4E-007	9	9.4E-008	9	2.8E-009

Table 6.1: Numerical results ($q = \frac{\log n}{6}$ in ψ_3)

Problem	ψ_1		ψ_3		LIPSOL	
	it.	$E(x,y,s)$	it.	$E(x,y,s)$	it.	$E(x,y,s)$
SC50B	7	5.4E-007	7	4.0E-007	7	1.6E-007
SCAGR7	13	3.1E-007	13	5.8E-007	11	3.5E-007
SCFXM1	**18**	4.2E-007	23	3.7E-007	16	3.7E-007
SCFXM2	**21**	1.4E-006	22	1.3E-007	19	1.6E-008
SCFXM3	**23**	1.3E-007	25	5.9E-008	20	3.0E-010
SCSD1	13	3.9E-007	**12**	4.3E-008	10	3.3E-011
SCSD6	15	3.8E-007	**13**	3.3E-008	11	7.8E-008
SCSD8	13	6.4E-008	**12**	2.4E-007	11	4.0E-011
SCTAP1	**18**	5.5E-007	20	1.1E-007	16	1.2E-008
SCTAP2	19	1.3E-007	19	2.2E-008	18	3.5E-009
SCTAP3	19	6.4E-007	19	1.3E-008	17	2.4E-008
SHARE1B	**23**	6.0E-007	26	1.0E-008	21	1.9E-010
SHARE2B	12	1.3E-008	11	8.5E-007	11	1.7E-007
SHIP04L	**15**	7.8E-007	17	1.6E-007	12	5.6E-011
SHIP04S	**15**	3.2E-007	16	5.3E-008	12	3.6E-007
SHIP12L	**19**	9.1E-007	27	5.4E-007	15	7.7E-009
SHIP12S	**17**	1.3E-007	19	3.4E-008	15	3.6E-007
STOCFOR1	14	4.9E-007	23	9.2E-007	16	1.1E-007
STOCFOR2	**25**	1.8E-007	33	2.0E-008	21	2.3E-008
TRUSS	**18**	3.9E-007	20	7.3E-007	17	8.4E-007
WOOD1P	**17**	8.3E-007	18	9.7E-006	14	7.0E-010
WOODW	25	3.1E-007	25	9.9E-007	23	5.1E-010
Total	816		880		725	

Table 6.2: Numerical results ($q = \frac{\log n}{6}$ in ψ_3)

Problem	rows	columns	nonzeros	Optimal objective
25FV47	821	1876	10705	5.50185E+003
ADLITTLE	56	138	424	2.25495E+005
AFIRO	27	51	102	–4.64753E+002
AGG	488	615	2862	–3.59918E+007
AGG2	516	758	4740	–2.02393E+007
AGG3	516	758	4756	1.03121E+007
BANDM	305	472	2494	–1.58628E+002
BEACONFD	173	295	3408	3.35925E+004
BLEND	74	114	522	–3.08121E+001
BNL1	643	1586	5532	1.97763E+003
BNL2	2324	4486	14996	1.81124E+003
BRANDY	220	303	2202	1.51851E+003
D2Q06C	2171	5831	33081	1.22784E+005
DEGEN2	444	757	4201	–1.43518E+003
DEGEN3	1503	2604	25432	–9.87294E+002
E226	223	472	2768	–1.87519E+001
FFFFF800	524	1028	6401	5.55680E+005
ISRAEL	174	316	2443	–8.96645E+005
LOTFI	153	366	1136	–2.52647E+001
MAROS-R7	3136	9408	144848	1.49719E+006
SC105	105	163	340	–5.22021E+001
SC205	205	317	665	–5.22021E+001
SC50A	50	78	160	–6.45751E+001

Table 6.3: Netlib problems

Problem	rows	columns	nonzeros	Optimal objective
SC50B	50	78	148	–7.00000E+001
SCAGR7	129	185	465	–2.33139E+006
SCFXM1	330	600	2732	1.84168E+004
SCFXM2	660	1200	5469	3.66603E+004
SCFXM3	990	1800	8206	5.49013E+004
SCSD1	77	760	2388	8.66667E+000
SCSD6	147	1350	4316	5.05000E+001
SCSD8	397	2750	8584	9.05000E+002
SCTAP1	300	660	1872	1.41225E+003
SCTAP2	1090	2500	7334	1.72481E+003
SCTAP3	1480	3340	9734	1.42400E+003
SHARE1B	117	253	1179	–7.65893E+004
SHARE2B	96	162	777	–4.15732E+002
SHIP04L	402	2166	6380	1.79332E+006
SHIP04S	402	1506	4400	1.79871E+006
SHIP12L	1151	5533	16276	1.47019E+006
SHIP12S	1151	2869	8284	1.48924E+006
STOCFOR1	117	165	501	–4.11320E+004
STOCFOR2	2157	3045	9357	–3.90244E+004
TRUSS	1000	8806	27836	4.58816E+005
WOOD1P	244	2595	70216	1.44290E+000
WOODW	1098	8418	37487	1.30448E+000

Table 6.4: Netlib problems

7

Conclusions

7.1 Concluding remarks

In this thesis, we analyze large-update infeasible interior-point methods (IIPMs) for LO. Our work is motivated by [97] in which Roos presents a full-Newton IIPM for LO. Since the analysis of our large-update IIPMs requires properties of barrier functions based on kernel functions that are used in large-update feasible interior-point methods (FIPMs), we present primal-dual large-update FIPMs for LO based on kernel functions, as well.

In Roos' algorithm, the iterates move within small neighborhoods of the μ-centers of the perturbed problem pairs. As in many IIPMs, the algorithm reduces the infeasibility and the duality gap at the same rate. His algorithm has the advantage that it uses full Newton steps and hence no calculation of step size is needed. Moreover, its theoretical iteration bound is $O(n\log(\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})/\varepsilon))$ which coincides with the best-known iteration bound for IIPMs. Nevertheless, it has the deficiency that it is too slow in practice.

We attempt to design a large-update version of Roos' algorithm which allows larger reductions of $\epsilon(x,y,s)$ at an iteration. This requires that the parameter θ is larger than $O(1/n)$, even $\theta = O(1)$. Unfortunately, the result of the theoretical analysis in Chapter 5 implies that θ is $O(1/(n(\log n)^2))$ which yields $O(n\sqrt{n}(\log n)^3\log(\epsilon(\zeta\mathbf{e},0,\zeta\mathbf{e})/\varepsilon)$ iteration bound for a variant. Since the theoretical complexity of the algorithm is disappointing, we rely on the numerical results to establish that our algorithm is really a large-update method. A practically efficient version of the algorithm is presented and its numerical results are compared with the well-known LIPSOL package. Fortunately, the numerical results seem promising as our algorithm has iteration numbers close to those of LIPSOL and, in a few cases, outperforms LIPSOL. This makes clear that IIPMs suffer from the same irony as FIPMs, i.e., regardless of their nice practical performance, the theoretical complexity of large-update methods is worse. Recall that the best

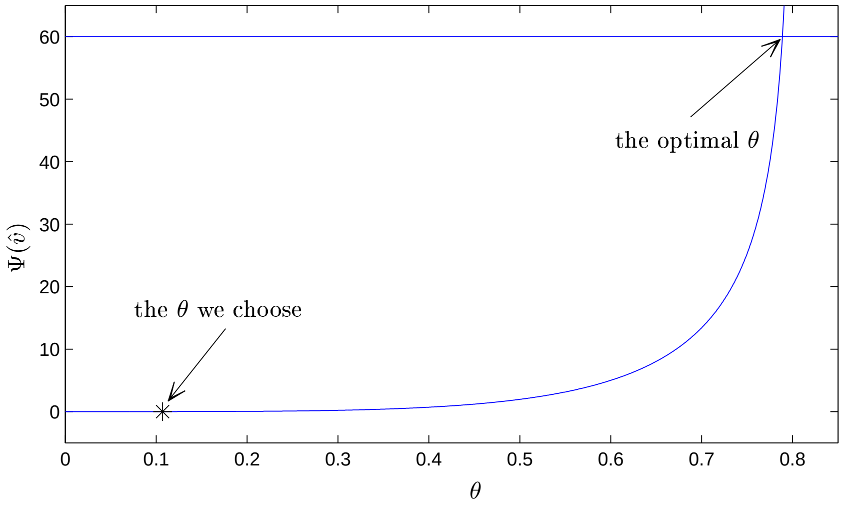


Figure 7.1: Typical behavior of $\Psi(\hat{v})$ as a function of θ

known iteration bound for large-update IIPMs is $O(n\sqrt{n}\log n \log(\epsilon(\zeta\mathbf{e}, 0, \zeta\mathbf{e})/\varepsilon)$ which is due to Salahi et al. [101].

As in other successful implementations like e.g., [64, 120], different step sizes in the primal and the dual spaces are used in our implementation. This gives rise to a faster achievement in feasibility than when identical step sizes are used. Moreover, inspired by the LIPSOL package, we use a predictor-corrector step in the feasibility step of the algorithm.

7.2 Further research

In this section, we mention a few directions for future research that are related to the subject of this thesis.

- As mentioned before, our algorithm has a factor $(\log n)^2$ worse iteration bound than the best known iteration bound for large-update IIPMs. One may consider how to modify the analysis such that the iteration bound of our algorithm is improved by a factor $(\log n)^2$.

- As mentioned in Section 7.1, according to the analysis of our algorithm presented in Chapter 5, the barrier-updating parameter θ is $O(1/(n(\log n)^2))$. This yields the loose iteration bound given by (5.45). This slender value of θ is obtained because of some difficulties in the analysis of the algorithm which uses the largest value of θ, satisfying (5.21), to assure that $\Psi(\hat{v}) = O(n)$.

This value of θ is much smaller than the best value we may choose. A typical graph of $\Psi(\hat{v})$, as a function of θ, is as depicted in Figure 7.1. Assuming $n = 60$, the largest value of θ satisfying $\Psi(\hat{v}) = n$ is 0.788840 while the value of θ suggested by theory is 0.107140. A future research may focuss on some new analysis of the algorithm which yields some larger value of θ.

- Roos' full-Newton step IIPM was extended to Semidefinite Optimization (SDO) by Mansouri and Roos [67], to Symmetric Optimization (SO) by Gu et al. [47] and to LCP by Mansouri et al. [68]. An extension of large-update FIPMs based on kernel functions to SDO was presented by El Ghami [37]. One may consider how our algorithm behaves in theory and practice when it is extended to the cases of SDO, SO and LCP.

Bibliography

[1] I. Adler, G. C. Resende, G. Veiga, and N. Karmarkar. An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44(3, (Ser. A)):297–335, 1989.

[2] K. Amini and A. Haseli. A new proximity function generating the best known iteration bounds for both large-update and small-update interior-point methods. *ANZIAM J.*, 49(2):259–270, 2007.

[3] Erling D. Andersen and Knud D. Andersen. The Mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High performance optimization*, volume 33 of *Appl. Optim.*, pages 197–232. Kluwer Acad. Publ., Dordrecht, 2000.

[4] K. M. Anstreicher. A monotonic projective algorithm for fractional linear programming. *Algorithmica*, 1(4):483–498, 1986.

[5] K. M. Anstreicher. A standard form variant, and safeguarded linesearch, for the modified Karmarkar algorithm. *Mathematical Programming*, 47(3, (Ser. A)):337–351, 1990.

[6] K. M. Anstreicher. Potential reduction algorithms. In *Interior point methods of mathematical programming*, volume 5 of *Appl. Optim.*, pages 125–158. Kluwer Acad. Publ., Dordrecht, 1996.

[7] K. M. Anstreicher and R. A. Bosch. Long steps in an $O(n^3L)$ algorithm for linear programming. *Mathematical Programming*, 54(3):251–265, 1992.

[8] A. Asadi, G. Gu, and C. Roos. Convergence of the homotopy path for a full-newton step infeasible interior-point method. *Operations Research Letters*, 38(2):147 – 151, 2010.

[9] A. Asadi and C. Roos. A class of large-update infeasible interior-point algorithms for linear optimization. Submitted to *Optimization Methods & Software*, 2010.

[10] Y. Q. Bai, M. El Ghami, and C. Roos. A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization. *SIAM J. Optim.*, 15(1):101–128 (electronic), 2004.

[11] Y. Q. Bai, M. El Ghami, and C. Roos. A primal-dual interior-point algorithm for linear optimization based on a new proximity function. *Optimization Methods & Software*, 17(6):985–1008, 2002.

[12] Y. Q. Bai, J. Guo, and C. Roos. A new kernel function yielding the best known iteration bounds for primal-dual interior-point algorithms. *Acta Mathematica Sinica*, 25:2169–2178, December 2009.

[13] E. R. Barnes. A variation on Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36(2):174–182, 1986.

[14] S. C. Billups and M. C. Ferris. Convergence of an infeasible interior-point algorithm from arbitrary positive starting points. *SIAM J. Optim.*, 6(2):316–325, 1996.

[15] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: a survey. *Oper. Res.*, 29(6):1039–1091, 1981.

[16] A. Charnes, W. W. Cooper, and A. Henderson. *An introduction to linear programming.* John Wiley & Sons Inc., New York, 1953.

[17] I. C. Choi, C. L. Monma, and D. F. Shanno. Further development of a primal–dual interior point method. *ORSA Journal on Computing*, 2:304–311, 1990.

[18] J. Czyzyk, S. Mehrotra, M. Wagner, and S. Wright. PCx: an interior-point code for linear programming. *Optimization Methods & Software*, 11/12(1-4):397–430, 1999.

[19] G. B. Dantzig. Reminiscences about the origins of linear programming. *Operations Research Letters*, 1(2):43–48, 1981/82.

[20] G. B. Dantzig. *Linear programming and extensions.* Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, corrected edition, 1998.

[21] T. A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[22] G. de Ghellinck and J.-Ph. Vial. A polynomial Newton method for linear programming. *Algorithmica*, 1(4):425–453, 1986.

[23] D. den Hertog and C. Roos. A survey of search directions in interior point methods for linear programming. *Mathematical Programming*, 52(3):481–509, 1991.

[24] D. den Hertog, C. Roos, and T. Terlaky. A potential-reduction variant of Renegar's short-step path-following method for linear programming.

[25] D. den Hertog, C. Roos, and J.-Ph. Vial. A complexity reduction for the long-step path-following algorithm for linear programming. *SIAM J. Optim.*, 2(1):71–87, 1992.

[26] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Dokl. Akad. Nauk SSSR*, 174:747–748, 1967.

[27] M. L. Dowling. An affine scaling algorithm for linear programming problems with inequality constraints. *Math. Methods Oper. Res.*, 43(3):301–318, 1996.

[28] H. A. Eiselt and C.-L. Sandblom. *Linear programming and its applications.* Springer, Berlin, 2007.

[29] M. V. Èl′vov. Analysis of the convergence of a class of barrier-projective methods for solving linear programming problems. *Zh. Vychisl. Mat. Mat. Fiz.*, 38(9):1525–1533, 1998.

[30] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: Sequential unconstrained minimization techniques.* John Wiley and Sons, Inc., New York-London-Sydney, 1968.

[31] C. Fraley and J.-Ph. Vial. Numerical study of projective methods for linear programming. In *Optimization (Varetz, 1988)*, volume 1405 of *Lecture Notes in Math.*, pages 25–38. Springer, Berlin, 1989.

[32] R. M. Freund. Polynomial–time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical Programming*, 51:203–222, 1991.

[33] D. Gale, H. W. Kuhn, and A. W. Tucker. Linear programming and the theory of games. In *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, pages 317–329. John Wiley & Sons Inc., New York, N. Y., 1951.

[34] S. I. Gass and A. A. Assad. *An annotated timeline of operations research: an informal history.* International Series in Operations Research & Management Science, 75. Kluwer Academic Publishers, Boston, MA, 2005.

[35] D. M. Gay. A variant of Karmarkar's linear programming algorithm for problems in standard form. *Mathematical Programming*, 37(1):81–90, 1987.

[36] A. George and J. W. H. Liu. *Computer solution of large sparse positive definite systems.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1981. Prentice-Hall Series in Computational Mathematics.

[37] M. El Ghami. *New primal-dual interior-point method based on kernel functions.* PhD thesis, Delft Univ. of Technology, The Netherlands, 2005.

[38] P. E. Gill, W. Murray, and M. A. Saunders. A single – phase dual barrier method for linear programming. Technical Report SOL 88–10, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, CA 94305, USA, August 1988.

[39] Ph. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected newton barrier methods for linear programming and an equivalence to karmarkar's projective method. *Mathematical Programming*, 36(2):183–209, 1986.

[40] D. Goldfarb and M. J. Todd. Modifications and implementation of the ellipsoid algorithm for linear programming. *Mathematical Programming*, 23(1):1–19, 1982.

[41] A. J. Goldman and A. W. Tucker. Theory of linear programming. In *Linear inequalities and related systems*, Annals of Mathematics Studies, no. 38, pages 53–97. Princeton University Press, Princeton, N.J., 1956.

[42] C. C. Gonzaga. An algorithm for solving linear programming programs in $O(n^3L)$ operations. In *on Progress in Mathematical Programming: Interior-Point and Related Methods*, pages 1–28, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[43] C. C. Gonzaga. Interior point algorithms for linear programming with inequality constraints. *Mathematical Programming*, 52(2, Ser. B):209–225, 1991.

[44] C. C. Gonzaga. Large–step path–following methods for linear programming, Part I : Barrier function method. *SIAM J. Optim.*, 1:268–279, 1991.

[45] C. C. Gonzaga. Path-following methods for linear programming. *SIAM Rev.*, 34(2):167–224, 1992.

[46] G. Gu, H. Mansouri, M. Zangiabadi, Y. Q. Bai, and C. Roos. Improved full-newton step $O(n)$ infeasible interior-point method for linear optimization. *Journal of Optimization Theory and Applications*, 145(2), 2010.

[47] G. Gu, M. Zangiabadi, and C. Roos. Full Nesterov-Todd step infeasible interior-point method for symmetric optimization. To appear in *European Journal of Operational Research.*

[48] Farkas J. Über die Theorie der Einfachen Ungleichungen. *Journal für die Reine und Angewandte Mathematik*, 124:1–27, 1902.

[49] Fourier J. Solution d' une question particuliére du calcul des inégalités. *Nouveau Bulletin des sciences par la Société philomathique de Paris*, pages 99–100, 1826.

[50] B. Jansen, C. Roos, T. Terlaky, and J.-P. Vial. Primal-dual algorithms for linear programming based on the logarithmic barrier method. *J. Optim. Theory Appl.*, 83(1):1–26, 1994.

[51] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[52] L. G. Khachiyan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244(5):1093–1096, 1979. Translated into English in *Soviet Mathematics Doklady* 20, 191Ű-194.

[53] V. Klee and G. J. Minty. How good is the simplex algorithm? In *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pages 159–175. Academic Press, New York, 1972.

[54] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Mathematical Programming*, 61(3, Ser. A):263–280, 1993.

[55] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44(1, (Ser. A)):1–26, 1989.

[56] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear programming. In *Progress in mathematical programming (Pacific Grove, CA, 1987)*, pages 29–47. Springer, New York, 1989.

[57] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50(3, (Ser. A)):331–342, 1991.

[58] M. Kojima, T. Noma, and A. Yoshise. Global convergence in infeasible-interior-point algorithms. *Mathematical Programming*, 65(1, Ser. A):43–72, 1994.

[59] J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, editors. *History of mathematical programming: a collection of personal reminiscences.* North-Holland Publishing Co., Amsterdam, 1991.

[60] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming.* International Series in Operations Research & Management Science, 116. Springer, New York, third edition, 2008.

[61] I. J. Lustig. A practical approach to Karmarkar's algorithm. Technical Report SOL 85–5, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, CA 94305, USA, 1985.

[62] I. J. Lustig. Feasibility issues in a primal-dual interior-point method for linear programming. *Mathematical Programming*, 49(2, (Ser. A)):145–162, 1990/91.

[63] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Computational experience with a primal-dual interior point method for linear programming. *Linear Algebra Appl.*, 152:191–222, 1991.

[64] I. J. Lustig, R. E. Marsten, and D. F. Shanno. On implementing Mehrotra's predictor-corrector interior-point method for linear programming. *SIAM J. Optim.*, 2(3):435–449, 1992.

[65] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Computational experience with a globally convergent primal-dual predictor-corrector algorithm for linear programming. *Mathematical Programming*, 66(1, Ser. A):123–135, 1994.

[66] H. Mansouri and C. Roos. Simplified $O(nL)$ infeasible interior-point algorithm for linear optimization using full-Newton steps. *Optimization Methods & Software*, 22(3):519–530, 2007.

[67] H. Mansouri and C. Roos. A new full-Newton step $O(n)$ infeasible interior-point algorithm for semidefinite optimization. *Numer. Algorithms*, 52(2):225–255, 2009.

[68] H. Mansouri, M. Zangiabadi, and M. Pirhaji. A full-Newton step $O(n)$ infeasible-interior-point algorithm for linear complementarity problems. *Nonlinear Anal. Real World Appl.*, 12(1):545–561, 2011.

[69] R. E. Marsten, R. Subramanian, M. Saltzman, I. J. Lustig, and D. F. Shanno. Interior Point Methods for Linear Programming: Just Call Newton, Lagrange, and Fiacco and McCormick! *INTERFACES*, 20(4):105–116, 1990.

[70] J. F. McCloskey. The beginnings of operations research: 1934–1941. *Oper. Res.*, 35(1):143–152, 1987.

[71] K.A. McShane, C.L. Monma, and D.F. Shanno. An implementation of a primal-dual interior point method for linear programming. *ORSA Journal on Computing*, 1:70–83, 1989.

[72] N. Megiddo. Pathways to the optimal set in linear programming. In *Progress in mathematical programming (Pacific Grove, CA, 1987)*, pages 131–158. Springer, New York, 1989.

[73] N. Megiddo and M. Shub. Boundary behavior of interior point algorithms in linear programming. *Math. Oper. Res.*, 14(1):97–146, 1989.

[74] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. Optim.*, 2(4):575–601, 1992.

[75] S. Mizuno. Polynomiality of infeasible-interior-point algorithms for linear programming. *Mathematical Programming*, 67(1, Ser. A):109–119, 1994.

[76] S. Mizuno, M. Kojima, and M. J. Todd. Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming. *SIAM J. Optim.*, 5(1):52–67, 1995.

[77] S. Mizuno, M. J. Todd, and Y. Ye. On adaptive-step primal-dual interior-point algorithms for linear programming. *Math. Oper. Res.*, 18(4):964–981, 1993.

[78] S. Mizuno, M. J. Todd, and Y. Ye. A surface of analytic centers and primal-dual infeasible-interior-point algorithms for linear programming. *Math. Oper. Res.*, 20(1):135–162, 1995.

[79] C. L. Monma and A. J. Morton. Computational experience with a dual affine variant of Karmarkar's method for linear programming. *Operations Research Letters*, 6(6):261 – 267, 1987.

[80] R. D. C. Monteiro and I. Adler. Interior path–following primal-dual algorithms. Part I. Linear programming. *Mathematical Programming*, 44(1, (Ser. A)):27–41, 1989.

[81] 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 extension. *Math. Oper. Res.*, 15(2):191–214, 1990.

[82] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. *Mathematical Programming*, 14(1):41–72, 1978.

[83] J. V. Neumann. On a maximization problem (manuscript). Institute for Advanced Study, Princeton University, Princeton, NJ 08544, USA, 1947.

[84] M. Padberg. A different convergence proof of the projective method for linear programming. *Operations Research Letters*, 4(6):253–257, 1986.

[85] J. Peng, C. Roos, and T. Terlaky. New complexity analysis of the primal-dual Newton method for linear optimization. *Ann. Oper Res.*, 99:23–39 (2001), 2000.

[86] J. Peng, C. Roos, and T. Terlaky. A new and efficient large-update interior-point method for linear optimization. *Vychisl. Tekhnol.*, 6(4):61–80, 2001.

[87] J. Peng, C. Roos, and T. Terlaky. Self-regular functions and new search directions for linear and semidefinite optimization. *Mathematical Programming*, 93:129–171, 2002.

[88] J. Peng and T. Roos, C.and Terlaky. *Self-regularity: a new paradigm for primal-dual interior-point algorithms.* Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2002.

[89] M. Peyghami. *Two New Proximity Functions for Feasible Interior-Point Methods and a Show of an Ill-behaved Central Path.* PhD thesis, Sharif University of Technology, Iran, 2005.

[90] F. Potra. An infeasible interior-point predictor-corrector algorithm for linear programming. Reports on Computational Mathematics 26, Dept. of Mathematics, The University of Iowa, Iowa City, IA 52242, USA, June 1992.

[91] F. Potra. An infeasible-interior-point predictor-corrector algorithm for linear programming. *SIAM J. Optim.*, 6(1):19–32, 1996.

[92] J. Renegar. A polynomial-time algorithm, based on Newton's method for linear programming. *Mathematical Programming*, 40(1):59–93, 1988.

[93] J. Renegar. *A mathematical view of interior-point methods in convex optimization.* MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[94] G. Rinaldi. A projective method for linear programming with box-type constraints. *Algorithmica*, 1(4):517–527, 1986.

[95] C. Roos. New trajectory–following polynomial–time algorithm for linear programming problems. *Journal of Optimization Theory and Applications*, 63:433–458, 1989.

[96] C. Roos. An $O(n^3L)$ approximate center method for linear programming. In S. Dolecki, editor, *Optimization : Proceedings of the 5th French–German Conference in Castel–Novel, Varetz, France, October 1988*, volume 1405 of *Lecture Notes in Mathematics*, pages 147–158. Springer Verlag, Berlin, Germany, 1989.

[97] C. Roos. A full-Newton step $O(n)$ infeasible interior-point algorithm for linear optimization. *SIAM J. Optim.*, 16(4):1110–1136 (electronic), 2006.

[98] C. Roos, T. Terlaky, and J.-Ph. Vial. *Interior point methods for linear optimization.* Springer, New York, 2006. Second edition of *Theory and algorithms for linear optimization* [Wiley, Chichester, 1997].

[99] C. Roos and J.-Ph. Vial. Long steps with the logarithmic penalty barrier function in linear programming. In J. Gabszevwicz, J. F. Richard, and L. Wolsey, editors, *Economic Decision–Making : Games, Economics and Optimization, dedicated to J. H. Dreze*, pages 433–441. Elsevier Science Publisher B.V., Amsterdam, The Netherlands, 1989.

[100] C. Roos and J.-Ph. Vial. A polynomial method of approximate centers for linear programming. *Mathematical Programming*, 54(3):295–305, 1992.

[101] M. Salahi, M. R. Peyghami, and T. Terlaky. New complexity analysis of IIPMs for linear optimization based on a specific self-regular function. *European J. Oper. Res.*, 186(2):466–485, 2008.

[102] 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, editors, *System Modelling and Optimization : Proceedings of the 12th IFIP-Conference held in Budapest, Hungary, September 1985*, volume 84 of *Lecture Notes in Control and Information Sciences*, pages 866–876. Springer Verlag, Berlin, West–Germany, 1986.

[103] H. Taha. *Operations research.* Pearson Education, Inc., Upper Saddle River, New Jersey 07458, eighth edition, 2007.

[104] K. Tanabe. Centered Newton method for mathematical programming. In *System modelling and optimization (Tokyo, 1987)*, volume 113 of *Lecture Notes in Control and Inform. Sci.*, pages 197–206. Springer, Berlin, 1988.

[105] K. Tanabe. Centered newton method for mathematical programming. In *System modelling and optimization (Tokyo, 1987)*, volume 113 of *Lecture Notes in Control and Inform. Sci.*, pages 197–206. Springer, Berlin, 1988.

[106] M. J. Todd. Potential-reduction methods in mathematical programming. *Mathematical Programming*, 76(1, Ser. B):3–45, 1997. Interior point methods in theory and practice (Iowa City, IA, 1994).

[107] M. J. Todd and Y. F. Wang. On combined phase *I*–phase *II* projective methods for linear programming. *Algorithmica*, 9(1):64–83, 1993.

[108] M. J. Todd and Y. Ye. A centered projective algorithm for linear programming. *Math. Oper. Res.*, 15(3):508–529, 1990.

[109] J. A. Tomlin. An experimental approach to Karmarkar's projective method for linear programming. *Math. Programming Stud.*, (31):175–191, 1987.

[110] T. Tsuchiya. Theoretical analysis of an affine scaling algorithm: the interior point method and a duality theorem. *Proc. Inst. Statist. Math.*, 42(2):277–296, 1994.

[111] P. M. Vaidya. An algorithm for linear programming which requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic operations. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 29–38, New York, NY, USA, 1987. ACM.

[112] R. J. Vanderbei. LOQO: an interior point code for quadratic programming. *Optimization Methods & Software*, 11/12(1-4):451–484, 1999.

[113] R. J. Vanderbei and J. C. Lagarias. I. I. Dikin's convergence result for the affine-scaling algorithm. In *Mathematical developments arising from linear programming (Brunswick, ME, 1988)*, volume 114 of *Contemp. Math.*, pages 109–119. Amer. Math. Soc., Providence, RI, 1990.

[114] R. J. Vanderbei, M. S. Meketon, and B. A. Freedman. A modification of karmarkar's linear programming algorithm. *Algorithmica*, 1(4):395–407, 1986.

[115] S. Wright. *Primal-dual interior-point methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[116] Y. Ye. A class of projective transformations for linear programming. *SIAM J. Comput.*, 19(3):457–466, 1990.

[117] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50(2, (Ser. A)):239–258, 1991.

[118] Y. Ye. *Interior point algorithms.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1997.

[119] Y. Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM J. Optim.*, 4(1):208–227, 1994.

[120] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. *Optimization Methods & Software*, 10(1):1–31, 1998.

[121] Y. Zhang and D. Zhang. On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. *Mathematical Programming*, 68(3, Ser. A):303–318, 1995.

List of notations and abbreviations

Sets

$\mathbb{R}$	field of real numbers.
$\mathbb{R}_+$	set of nonnegative real number.
$\mathbb{R}_{++}$	set of positive real numbers.
$\mathbb{R}^n$	set of real n-vectors ($n \times 1$ matrices).
$\mathbb{R}^{m\times n}$	set of real $m \times n$ matrices.

Vectors and matrices

$\mathbf{e}$	all-one vector.
I	identity matrix.

Abbreviations

IPC	interior-point condition.
IPM(s)	interior-point method(s).
FIPM(s)	feasible interior-point method(s).
IIPM(s)	infeasible interior-point method(s).
PFM(s)	path-following method(s).
PRM(s)	potential-reduction method(s).
PC	predictor-corrector.
LO	linear optimization.
QP	quadratic problem.
LCP	linear complementarity problem.
HLCP	horizontal linear complementarity problem.

Summary

Large-update Infeasible Interior-Point Algorithms for Linear Optimization

Recently, Roos [97] proposed a full-Newton step infeasible interior-point method (IIPM) for linear optimization (LO). Shortly afterwards, Mansouri and Roos [66] presented a variant of this algorithm and Gu et al. [46] a version with a simplified analysis.

Roos' algorithm is a path-following method. It uses the so-called *homotopy path* as a guideline to an optimal solution. The algorithm has the advantage that it uses only full Newton steps (the step size is always 1, hence requires no computation), and its convergence rate is $O(n)$, which coincides with the best known convergence rate for IIPMs. Apart from these nice features, the algorithm has the deficiency that it is a small-update method and hence it is too slow for practical purposes.

In this thesis we design a large-update version of Roos' algorithm. We present a practically efficient implementation of (a variant of) the algorithm and compare its performance with that of the well-known LIPSOL package [120]. The numerical results are promising as the iteration numbers of our algorithm are close to those of LIPSOL; in a few cases they outperform LIPSOL.

Not surprisingly, as in large-update feasible interior-point methods (FIPMs), there is a gap between the practical and the theoretical behavior of our large-update IIPM. To be more precise, its theoretical convergence rate is $O(n\sqrt{n}(\log n)^3)$ which is worse than the convergence rate of its full-Newton step variant. This phenomenon is well-known in the field of IPMs, and has been called *the irony of IPMs*: small-update methods have the best complexity results and are slow in practice, whereas large-update methods have worse complexity results and excellent performance in practice. For example, large-update FIPMs are by a factor $O(\log n)$ worse than that of the full-Newton step FIPMs, i.e., $O(\sqrt{n}\log n)$ versus $O(\sqrt{n})$ [37, 98].

The thesis also contains a survey of IIPMs that have been presented by several authors in last two decades. It covers a wide range of methods, starting from Lustig's algorithm [62], to the infeasible potential-reduction methods of Mizuno, Kojima and Todd [76]. We focus on convergence properties and polynomiality of the IIPMs presented in our survey.

Samenvatting

Onlangs publiceerde Roos [97] een volle-Newton stap ‘infeasible’ inwendige-punt methode (IIPM) voor lineaire optimalisatie (LO). Iets later stelden Mansouri en Roos [66] een variant voor van dit algoritme, en kort daarna publiceerden Gu et al. [46] een versie met een envoudigere analyse.

Het algoritme van Roos is een padvolgende methode. Het gebruikt het zogenaamde *homotopie pad* als een gidslijn naar een optimale oplossing. Het algoritme heeft als voordeel dat het alleen volle Newton stappen gebruikt (er is dus geen berekening nodig van de stapgrootte, deze is altijd 1), en de convergentiesnelheid is $O(n)$, de best bekende convergentiesnelheid voor IIPMn. Naast deze goede eigenschappen heeft Roos’ algoritme het nadeel dat het een zogenaamde ‘small-update’ methode is, waardoor de methode te traag is voor practische doeleinden.

In dir proefschrift ontwerpen we een ‘large-update’ versie van genoemd algoritme van Roos. We presenteren een in de praktijk efficiënte implementaie van (een variant van) het algoritme en vergelijken de performance met die van het bekende pakket LIPSOL [120]. De numerieke resultaten zijn veelbelovend omdat de benodigde aantallen iteraties voor ons algoritme dicht bij die van LIPSOL liggen, en in enkele gevallen zelfs beter zijn.

Niet verrassend is dat er, evenals bij ‘large-update feasible’ inwendige-punt methoden (FIPMn), een discrepantie is tussen het practische en het theoretische gedrag van onze ‘large-update’ IIPM. De theoretische convergentiesnelheid is namelijk $O(n\sqrt{n}(\log n)^3)$, hetwelk slechter is dan de convergentiesnelheid van de volle-Newton stap variant. Dit verschijnsel is welbekend in het gebied van IPMn, en staat bekend als de *ironie van IPMn*: ‘small-update’ methoden hebben de beste complexiteit en zijn traag in de praktijk, terwijl ‘large-update’ methoden slechtere complexiteit hebben en in de praktijk veel sneller zijn. Bijvoorbeeld, de convergentiesnelheid van ‘large-update’ FIPMn is een factor $O(\log n)$ slechter dan die van volle-Newton stap FIPMn, namelijk, $O(\sqrt{n}\log n)$ versus $O(\sqrt{n})$ [37, 98].

Het proefschrift bevat ook en overzicht van IIPMn die gedurende de laatste twee decades zijn voorgesteld door diverse auteurs. Het beschrijft een groot aantal IIPMn, van Lustig’s algorithm [62], tot de ‘infeasible’ potentiaal-reductie methoden Mizuno, Kojima and Todd [76]. De nadruk in dit overzicht ligt op convergentie-eigenschappen en polynomialiteit van de besproken IIPMn.

Curriculum Vitae

Alireza Asadi was born in Marand, Eastern Azerbaijan, Iran on September 19, 1978.

He finished his secondary school in 1995 at Bahman Shamlou high school, Marand, Iran. He started his studies in Tabriz University, Tabriz, Iran and got his bachelor degree in Applied Mathematics in 2002. He got his master of science degree in Applied Mathematics, more specifically Numerical Analysis, at Tabriz University, Tabriz, Iran, in 2005.

In 2006, he started his PhD program at the Optimization Group, Department of Software Technology, faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology[1], Delft, The Netherlands, under the supervision of Prof. Kees Roos[2]. During his PhD, he was involved in a project entitled as *More Efficient Interior Point Methods for Conic Optimization.* His research was financially supported by the Netherlands Organization for Scientific Research (NWO Grant 613.000.441). Moreover, he completed the PhD course program of the Dutch Network on Operations Research (LNMB) and got an LNMB diploma.

[1] Technische Universiteit Delft (TU Delft)

[2] `http://www.st.ewi.tudelft.nl/~roos/`