Roland Török

Interior-point algorithms for solving linear complementarity problems and their applications

Department of Operations Research and Actuarial Sciences

Supervisor:

Petra Renáta Rigó Ph.D.
Associate Professor

Corvinus University of Budapest

Doctoral School of Economics, Business and Informatics

Interior-point algorithms for solving linear complementarity problems and their applications

Ph.D. Thesis

Roland Török

Budapest, 2025

Contents

Acknowledgements

First of all, I am very grateful to my supervisor, Petra Renáta Rigó, who helped me a lot during my PhD studies not only with her professional expertise, but also with motivation, encouragement and emotional support as well. Furthermore, I would like to express my gratitude towards the researchers of the Corvinus Centre for Operations Research, especially my previous supervisor Tibor Illés, who not only supported me during my MSc diploma thesis, but he also continued to guide me in my research. I am also very thankful to my fellow researcher and friend, Anita Varga, whose professional insights and emotional support really helped me during my PhD studies.

On the other hand, I feel deeply indebted to my family, who supported me emotionally throughout this period of my life. I am especially thankful to my mother and her partner, Sándor, for their invaluable help. I also greatly appreciate the encouragement of my brother, Richárd, and his wife during this challenging time. I am particularly grateful to my girlfriend, Nóri, who not only supported me emotionally but also gave me all the help I needed to get through the most difficult moments. Furthermore, I am thankful to her mother, Anita, for her kind emotional support as well. Finally, I feel blessed to have our dog, Zsüli, who never failed to make me smile.

Moreover, I am deeply grateful to my friends, Tamás, Andrea, and Zsolt, for their unwavering emotional support and encouragement during this phase of my life. I would also like to highlight my best friend, Tivadar, who was always there whenever I needed help.

Finally, I would like to express my sincere appreciation to all the researchers at the Doctoral School of Economics, Business, and Informatics for their support throughout my PhD. I am also thankful to the Doctoral School for its financial assistance, particularly through the Corvinus Doctoral Scholarship.

Abbreviations and notations

Linear optimization - LO

Quadratic programming - QP

Linear complementarity problem - LCP

Positive semidefinite - PSD

Second-order cone optimization - SOCO

Symmetric cone optimization - SCO

Cartesian symmetric cone linear complementarity problem - SCLCP

Cartesian symmetric cone horizontal linear complementarity problem - SCHLCP

Interior-point algorithm - IPA

Primal-dual - PD

Predictor-corrector - PC

Lowercase Latin letters - Scalars and indices.

Bold lowercase Latin letters - vectors.

Uppercase Latin letters - matrices.

$\mathbb{R}^n$ - the set of n-dimensional vectors.

$\mathbb{R}^n_+$ - the set of n-dimensional vectors with strictly positive coordinates.

$\mathbb{R}^n_\oplus$ - the set of n-dimensional vectors with nonnegative coordinates.

$\mathbf{e}$ - the all-ones vector with the corresponding size.

$\mathbf{xs}$ - the componentwise product of vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s} \in \mathbb{R}^n$.

$\frac{\mathbf{x}}{\mathbf{s}}$ - the componentwise division of vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s} \in \mathbb{R}^n$, where $\mathbf{s}_i \neq 0, \forall 0 \leq i \leq n$.

$\sqrt{\mathbf{xs}}$ - the componentwise square root of vector $\mathbf{xs}$, where $\mathbf{x}, \mathbf{s} \in \mathbb{R}^n$.

$\mathbf{x}^-$ - $\min\{\mathbf{x}, \mathbf{0}\}$, where the minimum is taken componentwise and $\mathbf{x} \in \mathbb{R}^n$.

$\mathbf{x}^T\mathbf{s}$ - the scalar product of vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s} \in \mathbb{R}^n$.

$\Delta\mathbf{x}, \Delta\mathbf{s}$ - Newton directions.

$\mathrm{diag}(\mathbf{x})$ - diagonal matrix formed by the elements of the vector $\mathbf{x}$.

$\|.\|$ - Euclidean norm.

$\|.\|_\infty$ - infinity norm.

Chapter 1

Introduction

1.1 Historical background

The development of linear optimization (LO) began during World War II, where the optimization of resource allocation in military logistics was needed. Dantzig introduced the simplex algorithm in 1947 for solving LO problems [20]. Although the simplex algorithm has worked very well in practice, theoretically in the worst case the simplex algorithm requires exponential number of iterations. In 1972, Klee and Minty [67] constructed an example for which the simplex method takes a number of steps exponential in the problem size. In the meantime, the development of interior-point algorithms (IPAs) started. First, Frisch [41] showed a new approach in 1957, called multiplex method that used logarithmic function for determining the direction in each step. In 1967 Dikin [34] showed an iterative method for solving LO and quadratic programming (QP) problems as well. The ellipsoid method was proposed first by Yudin and Nemirovski [112] and then Kachiyan adopted it for LO [64]. This is the first polynomial algorithm for LO. In 1984, Karmarkar [62] showed a projective algorithm that had a huge impact on the development of IPAs.

IPAs have shown to be successful in solving many difficult optimization problems quite efficiently. The most important results related to IPAs for solving LO problems are summarized in [32, 69, 82, 93, 107, 109]. IPAs have been extended to more general problem classes, such as linearly constrained (convex) QP [4, 108], semidefinite programming (SDP) [31, 33], second-order cone optimization (SOCO) [97, 105], symmetric cone optimization (SCO) [91, 94], and linear complementarity problems (LCPs) [19, 47, 49, 50].

LCPs are a heavily studied topic nowadays. It is important to mention that currently a generalization of LCPs is studied as well. In 2009, Luo and Xiu, in [74] presented an IPA for solving Cartesian symmetric cone linear complementarity problems (SCLCPs). Later, Lešaja et al. [73] introduced an IPA based on kernel functions for solving SCLCPs. In 2016, Asadi et al. [9] presented their results about the Cartesian symmetric cone horizontal linear complementarity problems (SCHLCPs). Since then, several results have been published about SCHLCPs [10–12]. Finally, we remark that LO, SDP, SCO, QP, LCP can be reformulated as SCHLCP, hence these are special cases of SCHLCP.

The Karush-Kuhn-Tucker optimality conditions of LO and QP problems lead to LCPs. The monographs written by Cottle et al. [19] and Kojima et al. [69] summarize the most important results related to the theory and applications of LCPs. Furthermore, several important applications of LCPs arise in different fields, such as economics, finance, engineering, etc. This also motivates the study of these types of problems. In Section 1.6 we present several applications of LCPs. It is really important to mention that the complementarity condition appeared first in 1939 [63]. Later, in 1963 Lemke and Howson [71] introduced the well-known Lemke algorithm. This algorithm is a pivot method for getting a solution for LCPs. They showed that the Nash equilibria of a bimatrix game is equivalent to a solution of the corresponding LCP.

1.2 Matrix classes

It is well-known that the solvability of an LCP is highly influenced by the properties of the problem's matrix. Note that, in general, LCPs belong to the class of NP-complete problems [16]. It is proved that if the problem's matrix has a special property, such as skew-symmetric [93, 107, 109] or positive semidefinite (PSD) [68], then the problem can be solved using IPAs in polynomial time. However, an important class of LCPs, that will be discussed in a more detailed way, is the class of sufficient LCPs. Cottle, Pang, and Venkateswaran [18] introduced the class of *sufficient matrices.*

Definition 1.2.1. (Cottle et al. [18]) A matrix $M \in \mathbb{R}^{n\times n}$ is a *column sufficient matrix* if for all $\mathbf{x} \in \mathbb{R}^n$

$$X(M\mathbf{x}) \leq 0 \ \text{ implies } \ X(M\mathbf{x}) = 0,$$

where $X = \text{diag}(\mathbf{x})$. Analogously, the matrix M is *row sufficient* if M^T is column sufficient. The matrix M is *sufficient* if it is both row and column sufficient.

The matrix class name sufficient originates from the meaning that if the matrix satisfies the conditions defined for sufficiency, the solution set of the problem is a convex, closed, and bounded polyhedron [18]. On the other hand, there is another important matrix class, that is called $P_*(\kappa)$-matrix class. Kojima et al. [69] proposed the notion of $P_*(\kappa)$-matrices.

Definition 1.2.2. (Kojima et al. [69]) Let $\kappa \geq 0$ be a nonnegative real number. A matrix $M \in \mathbb{R}^{n\times n}$ is a $P_*(\kappa)$-matrix if

$$(1+4\kappa) \sum_{i\in I_+(\mathbf{x})} x_i(Mx)_i + \sum_{i\in I_-(\mathbf{x})} x_i(Mx)_i \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^n, \tag{1.2.1}$$

where

$$I_+(\mathbf{x}) = \{1 \leq i \leq n : x_i(Mx)_i > 0\} \text{ and } I_-(\mathbf{x}) = \{1 \leq i \leq n : x_i(Mx)_i < 0\}.$$

The smallest of the parameter κ is called the handicap of the matrix M and is denoted by $\hat{\kappa}(M)$.

Definition 1.2.3. (Kojima et al. [69]) We define the class of P_*-matrices as

$$P_* = \bigcup_{\kappa\geq 0} P_*(\kappa).$$

In 1991, Kojima et al. [69] proved that if a matrix is a $P_*(\kappa)$-matrix, it is column sufficient as well. In 1995 Guu and Cottle [45] proved the row sufficiency of a $P_*(\kappa)$-matrix, so in this case, it is proven that if a matrix is $P_*(\kappa)$, it is also sufficient. Finally, in 1996 Väliaho proved that the class of P_*-matrices is equivalent to the class of sufficient matrices [101]. It is well known that IPAs for $P_*(\kappa)$-LCPs have polynomial iteration complexity in the size of the problem, the handicap κ of the matrix, the starting point's duality gap and in the accuracy parameter. This is the largest class of LCPs for which convergence of the IPAs can be established without requiring boundedness of the level set condition. In this case it is enough to assume that the strict interior of the feasible set is nonempty. However, de Klerk and E.-Nagy showed in [33] that the handicap of a matrix can be exponential in the bit length of the data. This means that it is still unknown whether there exist polynomial algorithms in the size of the problem and bit length for sufficient matrices. Analyses of IPAs are usually dependent on the parameter κ in their complexity results. However, the numerical results in practice show far better iteration numbers than the theoretical complexity results. This was shown in some papers using different types of IPAs [25, 27], which motivated us to conduct research on the theory and applications of IPAs for solving LCPs.

1.3 Classification of interior-point algorithms

Several types of IPAs have been discussed in the literature. We can classify IPAs based on the step length. They can be short- and long-step algorithms. The short-step algorithms work in a small neighborhood, while the long-step algorithms work in a wide neighborhood of the central path. For a long time, there was a gap in theoretical and practical behavior of the two types of IPAs, since the short-step methods had better theoretical results than the long-step algorithms, while in practice the latter ones showed to be more efficient. Firstly, Peng et al. [86] introduced IPAs based on the class of the self-regular barriers to reduce this gap. For some instances of self-regular kernel functions, the iteration bound for the long-step IPAs is $\mathcal{O}\left(\sqrt{n}\log n \log\left(\frac{n}{\epsilon}\right)\right)$. Bai et al. [14] obtained the same improvement for solving LO with an IPA based on eligible kernel functions. Later, Lešaja and Roos [72] generalized this result to $P_*(\kappa)$-LCPs with IPAs.

Potra [87] introduced the first IPA acting in a wide neighborhood of the central path that has both superlinear convergence and $\mathcal{O}(\sqrt{n}L)$ iteration complexity, which is currently the best known iteration bound for short-step IPAs. Later on, with a new technique, Ai and Zhang [3] introduced a large-step IPA for monotone LCPs, that has the same complexity as the currently known best short-step IPAs. In [90], Potra generalized the algorithm that was introduced by Ai and Zhang to sufficient LCPs. It is also important to mention that this algorithm works in a wide neighborhood of the central path.

On the other hand, we can speak about primal-dual (PD) algorithms or predictor-corrector (PC) IPAs. In the case of PC IPAs, we have two types of steps. Predictor (greedy) step aims to reach the optimal solution in a greedy way and gives the iterate outside of the neighborhood of the central path and then corrector steps return it in the corresponding neighborhood of the central path.

It is necessary to mention that in the theory of IPAs the determination of the search direction plays a key role. There is the aforementioned method based on kernel function [86]. On the other hand, Darvay presented another new technique that is called the algebraically equivalent transformation (AET) of the centering equations of the central path system [22]. The main idea behind this technique is to apply a continuously differentiable and invertible function φ on the nonlinear equation of the central path system. He used the function $\varphi(t) = \sqrt{t}$ in [22]. It is important to mention that in 2006 Achache [1] introduced a new PD IPA based on the same AET function that was introduced by Darvay. However, Achache introduced this new approach to solve convex QP problems. Later, Achache [2] extended his results to solve LCPs as well. On the other hand, the first PC IPAs for solving LO and linearly constrained QP that used this technique were introduced by Darvay [23]. Kheirfam generalized these algorithms to $P_*(\kappa)$-horizontal LCPs [65]. In most articles, the identity function is used as AET function, however Darvay used $\varphi(t) = \sqrt{t}$ in his first results related to the AET approach [21, 22]. In papers [24, 25], Darvay et al. proposed another AET function, namely $\varphi(t) = t - \sqrt{t}$, to define new IPAs for LO and $P_*(\kappa)$-LCPs. Haddou et al. [46] introduced a class of concave functions using another type of transformation on the central path system. Note that $\varphi(t) = t - \sqrt{t}$ is not a member of this class. Rigó in her PhD thesis [92] presented short-step PD IPAs and PC IPAs based on different types of AET functions. An important and really interesting topic in this area was to introduce a class of functions which contains most of the already known AET functions and to prove that applying any function from the class would result in an IPA that has polynomial complexity. Varga [103] in her PhD thesis showed a class of AET functions for Ai-Zhang type long-step IPAs. Further results regarding Ai-Zhang type long-step IPAs can be found in [79, 80]. In this thesis a new class of AET functions for short-step IPAs and a new class of AET functions for PC IPAs will be presented [58].

1.4 Linear complementarity problems

In the first part of this section we present some basic concepts related to the theory of $P_*(\kappa)$-LCPs and $P_*(\kappa)$-matrices.

1.4.1 Linear complementarity problems and $P_*(\kappa)$-matrices

The aim of the LCPs is to find vectors $\mathbf{x}, \mathbf{s} \in \mathbb{R}^n$, that satisfy the following constraints:

$$-M\mathbf{x} + \mathbf{s} = \mathbf{q},\ \mathbf{x}, \mathbf{s} \geq \mathbf{0},\ \mathbf{x}\mathbf{s} = \mathbf{0}, \tag{LCP}$$

where $M \in \mathbb{R}^{n \times n}$, $\mathbf{q} \in \mathbb{R}^n$ and $\mathbf{xs}$ is the componentwise product of vectors $\mathbf{x}$ and $\mathbf{s}$. The feasible region, the interior and the solution set of the LCP are given as follows:

$$\begin{aligned}
\mathcal{F} &:= \{(\mathbf{x}, \mathbf{s}) \in \mathbb{R}^n_\oplus \times \mathbb{R}^n_\oplus : -M\mathbf{x} + \mathbf{s} = \mathbf{q}\}, \\
\mathcal{F}^+ &:= \{(\mathbf{x}, \mathbf{s}) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : -M\mathbf{x} + \mathbf{s} = \mathbf{q}\}, \\
\mathcal{F}^* &:= \{(\mathbf{x}, \mathbf{s}) \in \mathcal{F} : \mathbf{xs} = \mathbf{0}\}.
\end{aligned}$$

A problem is called $P_*(\kappa)$-LCP if the coefficient matrix of (LCP) is $P_*(\kappa)$-matrix. Throughout the thesis we have the following assumption.

Assumption 1.4.1. Suppose that $\mathcal{F}^+ \neq \emptyset$ and M is a $P_*(\kappa)$-matrix.

Hence, we are dealing with $P_*(\kappa)$-LCPs.
The *central path problem* in this case is

$$-M\mathbf{x} + \mathbf{s} = \mathbf{q} \qquad \mathbf{x}, \mathbf{s} > \mathbf{0}, \qquad \mathbf{xs} = \mu\mathbf{e}, \tag{CPP}$$

where $\mathbf{e}$ denotes the n-dimensional all-one vector and $\mu > 0$.

Theorem 1.4.1. *(Kojima et al. [69]) For a given LCP with matrix $M \in \mathcal{P}_*(\kappa)$ the following three statement are equivalent:*

- $\mathcal{F}^+ \neq 0$;
- $\forall \boldsymbol{w} \in \mathbb{R}^n_+, \exists!(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{F}^+ : \boldsymbol{xs} = \boldsymbol{w}$;
- $\forall \mu > 0, \exists!(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{F}^+ : \boldsymbol{xs} = \mu\boldsymbol{e}$, *so the central path exists and unique.*

Remark 1.4.1. T. Illés, C. Roos and T. Terlaky gave a constructive proof for the theorem in an unpublished manuscript in 1998. This proof can be found in the PhD thesis of M. E.-Nagy [77].

The unique solution to the central path system for a certain μ is called μ-center. The set of μ-centers form a path towards the solution when μ is running through positive real numbers.

In the following subsection we present the classical AET approach.

1.4.2 Algebraically equivalent transformation technique

In this subsection, we present the AET technique in the case of $P_*(\kappa)$-LCPs. It is important to mention that this technique was first proposed for LO problems by Darvay [22].

Let $\varphi : (\hat{\xi}, \infty) \to \mathbb{R}$, with $0 \le \hat{\xi} < 1$, be a continuously differentiable and invertible function, such that $\varphi'(t) > 0$, $\forall t > \hat{\xi}$, see [22]. We use the notation $\varphi(\mathbf{x}) = [\varphi(x_1), \varphi(x_2) \ldots, \varphi(x_n)]^T$. System ($CPP$) can be written:

$$-M\mathbf{x} + \mathbf{s} = \mathbf{q} \qquad \mathbf{x}, \mathbf{s} > \mathbf{0}, \qquad \varphi\left(\frac{\mathbf{xs}}{\mu}\right) = \varphi(\mathbf{e}). \tag{CCP_φ}$$

Applying Newton's method we obtain the following system, see [25]:

$$\begin{aligned} -M\Delta\mathbf{x} + \Delta\mathbf{s} &= \mathbf{0}, \\ S\Delta\mathbf{x} + X\Delta\mathbf{s} &= \mathbf{a}_\varphi, \end{aligned} \tag{1.4.2}$$

where

$$\mathbf{a}_\varphi = \mu \frac{\varphi(\mathbf{e}) - \varphi\left(\frac{\mathbf{xs}}{\mu}\right)}{\varphi'\left(\frac{\mathbf{xs}}{\mu}\right)}. \tag{1.4.3}$$

This system has unique solution, which follows from the following result.

Corollary 1.4.1. *(Kojima et al. [69]) Let $M \in \mathbb{R}^{n\times n}$ be a $P_*(\kappa)$-matrix, $\mathbf{x}, \mathbf{s} \in \mathbb{R}^n_+$. Then, for all $\mathbf{a}_\varphi \in \mathbb{R}^n$ the system*

$$\begin{aligned} -M\Delta\mathbf{x} + \Delta\mathbf{s} &= \mathbf{0} \\ S\Delta\mathbf{x} + X\Delta\mathbf{s} &= \mathbf{a}_\varphi \end{aligned} \tag{1.4.4}$$

has a unique solution $(\Delta\mathbf{x}, \Delta\mathbf{s})$, where $X = \operatorname{diag}(\boldsymbol{x})$ and $S = \operatorname{diag}(\boldsymbol{s})$ are the diagonal matrices obtained from the vectors $\mathbf{x}$ and $\mathbf{s}$.

Scaling plays important role in the theory of IPAs. For scaling, we introduce the following notations:

$$\mathbf{v}=\sqrt{\frac{\mathbf{xs}}{\mu}},\quad \mathbf{d}=\sqrt{\frac{\mathbf{x}}{\mathbf{s}}},\quad \mathbf{d}_x=\frac{\mathbf{d}^{-1}\Delta\mathbf{x}}{\sqrt{\mu}}=\frac{\mathbf{v}\Delta\mathbf{x}}{\mathbf{x}},\quad \mathbf{d}_s=\frac{\mathbf{d}\Delta\mathbf{s}}{\sqrt{\mu}}=\frac{\mathbf{v}\Delta\mathbf{s}}{\mathbf{s}}. \tag{1.4.5}$$

From (1.4.5), we obtain

$$\Delta\mathbf{x}=\frac{\mathbf{x}\mathbf{d}_x}{\mathbf{v}}\quad\text{and}\quad \Delta\mathbf{s}=\frac{\mathbf{s}\mathbf{d}_s}{\mathbf{v}}. \tag{1.4.6}$$

After substituting (1.4.6) into (1.4.2) we obtain the scaled system:

$$\begin{aligned} -\bar{M}\mathbf{d}_x+\mathbf{d}_s &= \mathbf{0},\\ \mathbf{d}_x+\mathbf{d}_s &= \mathbf{p}_\varphi, \end{aligned} \tag{1.4.7}$$

where $\bar{M}=DMD$, $D=\operatorname{diag}(\mathbf{d})$ and

$$\mathbf{p}_\varphi=\frac{\varphi(\mathbf{e})-\varphi(\mathbf{v}^2)}{\mathbf{v}\varphi'(\mathbf{v}^2)}. \tag{1.4.8}$$

Table 1.4.1 contains the classical AET functions used in the literature and the corresponding vectors $\mathbf{a}_\varphi$ and $\mathbf{p}_\varphi$. In 2003 Darvay introduced the AET technique in [22], where he considered

φ	$\mathbf{a}_\varphi$	$\mathbf{p}_\varphi$
$\varphi(t)=t$	$\mu\mathbf{e}-\mathbf{xs}$	$\mathbf{v}^{-1}-\mathbf{v}$
$\varphi(t)=\sqrt{t}$	$2(\sqrt{\mu\mathbf{xs}}-\mathbf{xs})$	$2(\mathbf{e}-\mathbf{v})$
$\varphi(t)=t-\sqrt{t}$	$\frac{\sqrt{\mu}\mathbf{xs}}{2\sqrt{\mathbf{xs}}-\sqrt{\mu}\mathbf{e}}-\mathbf{xs}$	$\frac{2(\mathbf{v}-\mathbf{v}^2)}{2\mathbf{v}-\mathbf{e}}$
$\varphi(t)=\frac{\sqrt{t}}{2(1+\sqrt{t})}$	$\sqrt{\frac{\mathbf{xs}}{\mu}}(\mu\mathbf{e}-\mathbf{xs})$	$\mathbf{e}-\mathbf{v}^2$

Table 1.4.1: AET functions used in the literature

the case of $\varphi(t)=\sqrt{t}$. Later, Darvay et al. [24] proposed an IPA based on the function $\varphi(t)=t-\sqrt{t}$ that was extended to $P_*(\kappa)$-LCPs in [26] and to SCHLCP in [12, 28]. Finally, Kheirfam and Haghighi in [66] presented a full-Newton step feasible IPA based on the function $\varphi(t)=\frac{\sqrt{t}}{2(1+\sqrt{t})}$.

In this thesis, we present a PD IPA and a PC IPA that are based on new classes of AET functions. These results can be found in Chapters 2 and 3.

1.5 Search directions based on kernel functions

In this section, we present the kernel based approach for determining search directions. Later, we show that these classes of kernel functions are different from the ones corresponding to the classes of AET functions defined in this thesis.

In the literature, there are several properties and definitions about kernel functions. For a kernel function, the following properties have to be satisfied.

Definition 1.5.1. A function $\psi : \mathbb{R}_+ \to \mathbb{R}_\oplus$ is called a *kernel function* if it is twice continuously differentiable and if the following conditions hold:

K1. $\psi(1) = \psi'(1) = 0$;

K2. $\psi''(t) > 0$, for all $t > 0$;

K3. $\lim_{t\to 0}\psi(t) = \lim_{t\to\infty}\psi(t) = \infty$.

We can construct a barrier function $\Psi : \mathbb{R}^n_+ \to \mathbb{R}_+$ as a separable function of the following form:

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

where $\mathbf{v} \in \mathbb{R}^n_+$.

In [86] the authors proposed a modification of the second equation of the scaled system as

$$\mathbf{d}_x + \mathbf{d}_s = -\nabla\Psi(\mathbf{v}).$$

Using the modified second equation and the scaled system (1.4.7), we have

$$\mathbf{d}_x + \mathbf{d}_s = -\nabla\Psi(\mathbf{v}) = \mathbf{p}_\varphi = \frac{\varphi(\mathbf{e}) - \varphi(\mathbf{v}^2)}{\mathbf{v}\varphi\prime(\mathbf{v}^2)}. \tag{1.5.1}$$

Therefore, the corresponding kernel functions could be assigned to several functions φ appearing in the AET technique in the following way, see [84]:

$$\psi(t) = \int_1^t \frac{\varphi(\bar\tau^2) - \varphi(1)}{\bar\tau\varphi'(\bar\tau^2)} d\bar\tau, \tag{1.5.2}$$

where the function ψ satisfies the properties *K1.-K3.*

1.5.1 Self-regular functions

In [86], Peng, Roos and Terlaky introduced a class of self-regular functions:

Definition 1.5.2. (Peng et al. [86]) A function $\psi : (0,\infty) \to \mathbb{R},\ \psi \in \mathcal{C}^2$ is *self-regular* if it satisfies the conditions

SR1. $\psi(t)$ is strictly convex with respect to $t > 0$ and $\psi(t) = 0$ at its global minimal point $t = 1$, i.e. $\psi(1) = \psi'(1) = 0$. Further, there exist positive constants $\nu_2 \geq \nu_1 > 0$ and $p \geq 1,\ q \geq 1$ such that

$$\nu_1\left(t^{p-1} + t^{-1-q}\right) \leq \psi''(t) \leq \nu_2\left(t^{p-1} + t^{-1-q}\right)\ \forall t \in (0,\infty); \tag{1.5.3}$$

SR2. For any $t_1, t_2 > 0$,

$$\psi\left(t_1^{\hat{r}} t_2^{1-\hat{r}}\right) \leq \hat{r}\psi(t_1) + (1-\hat{r})\psi(t_2)\ \forall \hat{r} \in [0,1]. \tag{1.5.4}$$

They managed to reduce the theoretical complexity of long-step IPAs by using self-regular functions.

Now, we present the class of eligible kernel functions proposed in [14].

1.5.2 Eligible kernel functions

First, we give the conditions of the eligible kernel functions.

Definition 1.5.3. (Bai et al. [14]) We call a kernel function an *eligible kernel function* if it satisfies the following conditions:

EK1. $t\psi''(t) + \psi'(t) > 0\ \ \forall t < 1$;

EK2. $\psi'''(t) < 0,\ \ \forall t > 0$;

EK3. $2\psi''(t)^2 - \psi'(t)\psi'''(t) > 0\ \ \forall t < 1$;

EK4. $\psi''(t)\psi'(\hat{\beta}t) - \hat{\beta}\psi'(t)\psi''(\hat{\beta}t) > 0,\ \ \forall t > 1,\ \hat{\beta} > 1$.

Lešaja and Roos [72] generalized the IPA presented in [14] based on eligible kernel functions to $P_*(\kappa)$-LCPs. Darvay et al. [30] proposed an IPA for $P_*(\kappa)$-LCPs that is based on a barrier function which is defined by a subclass of the eligible kernel functions, called standard kernel functions. They provided a unified, comprehensive complexity analysis of the IPA and a general procedure to determine the iteration bounds for long-step and short-step versions of the method for the entire class of standard kernel functions.

In the following subsection several applications of LCPs will be presented.

1.6 Applications of linear complementarity problems

LCPs have a wide variety of applications in different fields, such as engineering, finance, economics, game theory and physics [19, 39, 71, 95]. In this section, we present some of the applications. The Karush-Kuhn-Tucker optimality conditions of LO and QP can be formulated as LCPs [19, 85]. This shows the importance of analysing LCPs. Several game theory problems can be transformed into LCPs, for example, bimatrix games [71] or quitting games [95]. It is also important that testing the copositivity of a matrix has a connection with solving LCPs [15]. The Arrow-Debreu competitive market equilibrium problem with linear and Leontief utility functions can be modelled as LCP, as well [78, 111]. These special cases of LCPs are discussed in the sections below.

1.6.1 Linear Optimization

Let us consider the PD LO problem as

$$\begin{aligned} \min \mathbf{c}^T\mathbf{x}, & \qquad \max \mathbf{b}^T\mathbf{y}, \\ A\mathbf{x} = \mathbf{b}, & \qquad A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \\ \mathbf{x} \geq \mathbf{0}, & \qquad \mathbf{s} \geq \mathbf{0}, \end{aligned} \tag{1.6.1}$$

where $A \in \mathbb{R}^{m\times n}$ is with full row rank, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$ are given.

The set of feasible solutions for the LO pair:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n_\oplus | A\mathbf{x} = \mathbf{b}\} \text{ and } \mathcal{D} = \{(\mathbf{y},\mathbf{s}) \in \mathbb{R}^m \times \mathbb{R}^n_\oplus | A^T\mathbf{y} + \mathbf{s} = \mathbf{c}\}.$$

The set of primal and dual interior points are:

$$\mathcal{P}^+ = \{\mathbf{x} \in \mathcal{P} | \mathbf{x} > 0\} \text{ and } \mathcal{D}^+ = \{(\mathbf{y},\mathbf{s}) \in \mathcal{D} | \mathbf{s} > 0\}.$$

The optimal solution sets are:

$$\mathcal{P}^* = \{\mathbf{x}^* \in \mathcal{P} | \mathbf{c}^T\mathbf{x} \geq \mathbf{c}^T\mathbf{x}^*, \forall \mathbf{x} \in \mathcal{P}\} \text{ and } \mathcal{D}^* = \{(\mathbf{y}^*, \mathbf{s}^*) \in \mathcal{D} | \mathbf{b}^T\mathbf{y} \leq \mathbf{b}^T\mathbf{y}^*, \forall (\mathbf{y}, \mathbf{s}) \in \mathcal{D}\}.$$

The following result is presented in different equivalent forms, see for example the book of Roos, Terlaky and Vial [93] and the paper of Terlaky [99].

Theorem 1.6.1. *(Weak duality theorem) For any* $\boldsymbol{x} \in \mathcal{P}$ *and* $(\boldsymbol{y}, \boldsymbol{s}) \in \mathcal{D}$ *the inequality* $\boldsymbol{c}^T\boldsymbol{x} \geq \boldsymbol{b}^T\boldsymbol{y}$ *holds, and with equality if and only if* $\boldsymbol{x}^T\boldsymbol{s} = 0$.

The optimality conditions of the PD LO problem are the following:

$$\begin{aligned} A\mathbf{x} = \mathbf{b}, &\quad \mathbf{x} \geq \mathbf{0}, \\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, &\quad \mathbf{s} \geq \mathbf{0}, \\ \mathbf{xs} = \mathbf{0}, & \end{aligned} \tag{1.6.2}$$

where the last equation is called complementarity condition and it is the componentwise product of $\mathbf{x}$ and $\mathbf{s}$. The first two equations are called primal and dual feasibility conditions.

If we consider (1.6.2), we have the following:

$$\widehat{M} = \begin{bmatrix} O & A \\ -A^T & O \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{q}} = \begin{pmatrix} -\mathbf{b} \\ \mathbf{c} \end{pmatrix}, \tag{1.6.3}$$

where $\widehat{M} \in \mathbb{R}^{(m+n)\times(m+n)}$ is skew-symmetric and $\hat{\mathbf{q}} \in \mathbb{R}^{m+n}$. With this reformulation, we can solve the LO problem as LCP as well.

1.6.2 Quadratic Programming

Let us consider the well-known linearly constrained QP. We would like to find the vector $\mathbf{x} \in \mathbb{R}^n$ that satisfies the following:

$$\begin{aligned} \min \frac{1}{2}\mathbf{x}^T\widehat{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}&, \\ A\mathbf{x} \geq \mathbf{b}&, \\ \mathbf{x} \geq \mathbf{0}&, \end{aligned} \tag{1.6.4}$$

where $\widehat{Q} \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$.

For this problem, the Karush-Kuhn-Tucker optimality conditions are the following:

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{c} + \widehat{Q}\mathbf{x} - A^T\mathbf{y} \geq \mathbf{0}, \quad && \mathbf{x} \geq \mathbf{0}, \quad && \mathbf{x}^T\tilde{\mathbf{u}} = 0, \\ \tilde{\mathbf{v}} &= -\mathbf{b} + A\mathbf{x} \geq \mathbf{0}, && \mathbf{y} \geq \mathbf{0}, && \mathbf{y}^T\tilde{\mathbf{v}} = 0. \end{aligned}$$

If we introduce the notation

$$\widehat{M} = \begin{bmatrix} \widehat{Q} & -A^T \\ A & O \end{bmatrix} \qquad \text{and} \qquad \hat{\mathbf{q}} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{b} \end{pmatrix}, \tag{1.6.5}$$

where $\widehat{M} \in \mathbb{R}^{(n+m)\times(n+m)}$ is bisymmetric and $\hat{\mathbf{q}} \in \mathbb{R}^{n+m}$, the QP-KKT can be reformulated as an LCP.

1.6.3 Exchange market equilibrium problems

The exchange market equilibrium problem was considered first by Walras in 1874 [104]. In this model, we have n traders in the market, each with an initial endowment of m divisible goods and a utility function that represents their consumption of their own and the other traders' goods. We assume that each trader sells their entire initial endowment and they buy goods in such a way that they are maximizing their utility function. We would like to find the market clearing prices that ensure every product is sold and all money is spent. In 1954, Arrow and Debreu [5] showed that such a price equilibrium exists under some mild conditions and if the utility function is concave. However, it is important to mention that Arrow and Debreu did not provide an algorithm to solve this problem, and even nowadays, finding solutions remains a heavily studied topic, typically limited to special cases of the model they considered.

A special type of market exchange model, later called the Fisher model, was proposed by Irving Fisher in his 1892 thesis [40]. Fisher divided the people on the market into producers and consumers. Producers bring their goods to the market, while consumers arrive with money. In this case, price equilibrium means that prices are set in such a way that each consumer spends all of their money, and each producer sells their entire endowment. It can be seen that this is a special case of the original problem if we consider money as a special type of good.

Let us consider the following notations:

$\hat{\mathbf{x}}^i = [\hat{x}_1^i, \hat{x}_2^i, \ldots, \hat{x}_m^i]^T \in \mathbb{R}^m_\oplus$: the goods of the ith trader,

$\hat{\mathbf{w}}^i = [\hat{w}_1^i, \hat{w}_2^i, \ldots, \hat{w}_m^i]^T \in \mathbb{R}^m_\oplus$: the initial endowments of the ith trader,

$\hat{\mathbf{p}} = [\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_m]^T \in \mathbb{R}^m_\oplus$: the prices of the goods,

$\hat{u}_i(\hat{\mathbf{x}}^i)$: the utility function of the ith trader,

$\hat{\mathbf{u}} = [\hat{u}_1(\hat{\mathbf{x}}^1), \hat{u}_2(\mathbf{x}^2), \ldots, \hat{u}_n(\mathbf{x}^n)]^T \in \mathbb{R}^n_\oplus$: the vector of the trader's utility functions.

Consider the special case where each trader has only one good (pairing model). This implies $n = m$. The linear and Leontief utility functions are well known in the literature. First, let us define the linear utility function:

$$\hat{u}_i(\hat{\mathbf{x}}^i) = \sum_{j=1}^{m} \hat{w}_j^i \hat{x}_j^i.$$

The Leontief utility function can be written as

$$\hat{u}_i(\hat{\mathbf{x}}^i) = \min_{j, \hat{a}_{ij}>0} \frac{\hat{x}_{ij}}{\hat{a}_{ij}},$$

where $\widehat{A} = (\hat{a}_{ij}) \in \mathbb{R}^{m\times m}_\oplus$ is the Leontief coefficient matrix.

Eisenberg and Gale [36, 37, 42] gave a convex programming approach of the Fisher case of the Walras problem with both linear and Leontief utility functions. With this reformulation, the problem can theoretically be solved using the ellipsoid method. In the case of linear utility functions, Eaves [35] presented the reformulation of the problem as a LCP, which means that it can also be solved by Lemke's algorithm. In this thesis, we consider the Fisher problem with Leontief utility functions and its reformulation.

In this case, the convex programming model of Eisenberg and Gale can be written as

$$\begin{aligned} &\max \sum_{i=1}^{n} \hat{w}_i \log \hat{u}_i(\hat{\mathbf{x}}^i), \\ &\widehat{A}^T \hat{\mathbf{u}} \le \mathbf{e}, \\ &\hat{\mathbf{u}} \ge \mathbf{0}, \end{aligned} \tag{1.6.6}$$

where $\mathbf{e}$ is the n dimensional vector of all ones. It was proven in [36] that an equilibrium price vector is an optimal Lagrangian multiplier of the convex programming problem (1.6.6). Thus, we search for such weights $\hat{w}_i$ that an optimal Lagrangian multiplier vector $\hat{\mathbf{p}}$ of (1.6.6) is equal to $\hat{\mathbf{w}}$. Therefore, $\hat{\mathbf{p}}$ is the solution of the following system (which can be derived from the KKT conditions for problem (1.6.6))

$$\begin{aligned} \widehat{U}\widehat{A}\hat{\mathbf{p}} &= \hat{\mathbf{p}}, \\ \widehat{P}(\mathbf{e} - \widehat{A}^T\hat{\mathbf{u}}) &= \mathbf{0}, \\ \widehat{A}^T\hat{\mathbf{u}} &\leq \mathbf{e}, \\ \hat{\mathbf{u}}, \hat{\mathbf{p}} &\geq \mathbf{0}, \\ \hat{\mathbf{p}} &\neq \mathbf{0}, \end{aligned} \tag{1.6.7}$$

where $\widehat{U} = \operatorname{diag}(\hat{\mathbf{u}})$ and $\widehat{P} = \operatorname{diag}(\hat{\mathbf{p}})$.

Based on this formulation, Ye [110] showed that the pairing Arrow-Debreu model with Leontief utilities is equivalent to a LCP problem. First, let us take into account the partition of the traders:

$$\widehat{B} = \{j : \hat{p}_j > 0\}, \qquad \widehat{N} = \{j : \hat{p}_j = 0\},$$

where $\widehat{B}$ is the index set of traders with a product that has a positive price in an equilibrium solution, and $\widehat{N}$ contains the indices of the remaining traders. It is important to mention that in a solution $\hat{u}_j(\hat{x}_j) > 0$ if and only if $\hat{p}_j > 0$. This is explained by the fact that positive prices are necessary for them to have positive utilities. Furthermore, if $\hat{p}_j > 0$, then the entire quantity of product j will be sold, that is, $\sum_{i=1}^m \hat{a}_{ij}\hat{u}_i(\hat{\mathbf{x}}^i) = 1$. Using the $\hat{B} - \hat{N}$ partition we can reformulate system (1.6.7) as follows:

$$\begin{aligned} (\widehat{U}_{\widehat{B}\widehat{B}}\widehat{A}_{\widehat{B}\widehat{B}})\hat{\mathbf{p}}_{\widehat{B}} &= \hat{\mathbf{p}}_{\widehat{B}}, \\ \widehat{A}^T_{\widehat{B}\widehat{B}}\hat{\mathbf{u}}_{\widehat{B}} &= \mathbf{e}, \\ \widehat{A}^T_{\widehat{B}\widehat{N}}\hat{\mathbf{u}}_{\widehat{B}} &\leq \mathbf{e}, \\ \hat{\mathbf{u}}_{\widehat{B}}, \hat{\mathbf{p}}_{\widehat{B}} &> \mathbf{0}, \end{aligned} \tag{1.6.8}$$

where $\widehat{U}_{\widehat{B}\widehat{B}}, \widehat{A}_{\widehat{B}\widehat{B}}$ are submatrices of $\widehat{U}$ and $\widehat{A}$, $\hat{\mathbf{u}}_{\widehat{B}}, \hat{\mathbf{p}}_{\widehat{B}}$ are subvectors of $\hat{\mathbf{u}}$ and $\hat{\mathbf{p}}$ corresponding to the index set of $\widehat{B}$, and $\widehat{A}_{\widehat{B}\widehat{N}}$ is the submatrix of $\widehat{A}$ based on the index set of $\widehat{B}$ as row indices and index set of $\widehat{N}$ as column indices. Since $\widehat{U}_{\widehat{B}\widehat{B}}\widehat{A}_{\widehat{B}\widehat{B}}$ is column stochastic, which is a square matrix where all entries are non-negative, and each column sums to 1. The vector $\hat{\mathbf{p}}_{\widehat{B}}$ is the right Perron-Frobenius eigenvector of this matrix. Finally, to derive the LCP problem let us take into account Theorem 4 from [110].

Theorem 1.6.2. *(Theorem 4 in [110]) Let* $\widehat{B} \subset \{1,2,\dots,n\}$, $\widehat{N} = \{1,2\dots,n\} \setminus \widehat{B}$, $\widehat{A}_{\widehat{B}\widehat{B}}$ *be irreducible, and* $\hat{\mathbf{u}}_{\widehat{B}}$ *satisfy the linear system*

$$\widehat{A}^T_{\widehat{B}\widehat{B}}\hat{\mathbf{u}}_{\widehat{B}} = \boldsymbol{e}, \quad \widehat{A}^T_{\widehat{B}\widehat{N}}\hat{\mathbf{u}}_{\widehat{B}} \leq \boldsymbol{e} \quad \text{and} \quad \hat{\mathbf{u}}_{\widehat{B}} > \boldsymbol{0}.$$

Then the (right) Perron-Frobenius eigenvector $\hat{\mathbf{p}}_{\widehat{B}}$ *of* $\widehat{U}_{\widehat{B}\widehat{B}}\widehat{A}_{\widehat{B}\widehat{B}}$ *together with* $\hat{\mathbf{p}}_{\widehat{N}} = \boldsymbol{0}$ *will be a solution to system (1.6.7). The converse is also true.*

Based on the previous theorem, it can be seen that we can find the equilibrium of the problem, if we find a non-trivial solution $\hat{\mathbf{u}} \neq \mathbf{0}$, to the following LCP:

$$\begin{aligned} \widehat{A}^T\hat{\mathbf{u}} + \hat{\mathbf{s}} &= \mathbf{e}, \\ \hat{\mathbf{u}}, \hat{\mathbf{v}} &\geq \mathbf{0}, \\ \hat{\mathbf{u}}\hat{\mathbf{v}} &= \mathbf{0}. \end{aligned} \tag{1.6.9}$$

Lastly, it is important to note that there exist polynomial algorithms for solving the Fisher model with both the linear and Leontief utility functions [111]. For the Arrow-Debreu model with linear utility functions there exist algorithms for solving it in polynomial iteration time [61, 111]. However, in the case of Arrow-Debreu model with Leontief utility functions the problem is NP-hard, as it is equivalent to general LCPs with a non-negative coefficient matrix [17].

1.7 Structure of the thesis

In this section, we discuss the structure of the thesis. Firstly, we present a literature review on the topic of LCPs, especially in the case of sufficient LCPs. The results from [56] were used in the subsequent chapters of the dissertation (Chapters 1-4).

In Chapter 2, we introduce the first class of AET functions for determining a PD IPA for solving sufficient LCPs. Firstly, we introduce the new class of AET functions. Next, we present the PD IPA that contains an AET function from this class as an input. Finally, in the last section, we provide the complexity analysis of the algorithm. These results are presented in [58] and also can be found in the research report [56].

In Chapter 3 we introduce a class of AET functions that is used in designing a PC IPA for solving $P_*(\kappa)$-LCPs, as described in Section 3.1. Up to our best knowledge, this is the first class of AET functions for determining a PC IPA. Firstly, we present the new class of AET functions. Next, we introduce the PC IPA that contains a function from this class of AETs as an input. Finally, we show the complexity analysis for this presented PC IPA. In Section 3.2 we present a special case of this class of AET functions that has inflection point. We introduce the case of $\varphi(t) = t^2 - t + \sqrt{t}$ as an input for the previously mentioned PC IPA. For this special function we present the main results related to the complexity analysis of the PC IPA. These results can be found in [59] and also in the research report [55].

In Chapter 4, we introduce a new type of PC IPA that works in a wide neighborhood of the central path. First, we introduce a generalization of the wide neighborhood that was introduced by Potra and Liu in [88]. We present the PC IPA in the case of $\varphi(t) = \sqrt{t}$ that works in the wide neighborhood of the central path. Finally, we present the complexity analysis of this algorithm. We prove the polynomial iteration complexity of the IPA in the size of the problem, parameter κ, the starting point's duality gap and in the accuracy parameter. Our obtained results can be found in [57] and in the research report [54].

In Chapter 5, we present our numerical results. First, we show how to calculate the Newton directions in a way that makes numerical errors less likely to occur during the calculation. After that, we present methods to calculate the step length in each iteration. Furthermore, we also give a condition to verify the feasibility of the new iterates. We present the implemented algorithms and we point out the differences between them and the theoretical IPAs. Finally, we present our numerical results based on two test sets that we generated during our research.

The summary of the thesis is given in Chapter 6. It can be observed that the thesis is partitioned into three main parts. The literature review is given in Chapter 1, the second part contains our theoretical research given in Chapter 2, 3, and 4. Finally, the third part of the thesis contains the obtained numerical results in Chapter 5.

Chapter 2

Primal-dual interior-point algorithm based on a new class of AET functions

Several articles were published including IPAs based on special AET functions. Our aim was to determine a whole class of AET functions for which we can define a short-step PD IPA and provide convergence and complexity analysis for the entire class. This result was later extended to SCHLCP in [29]. We also present an example for an AET function belonging to this new class, for which the corresponding kernel function is not an eligible kernel function. This chapter contains the theoretical results about this new class of AET functions for determining search directions with which we defined a new PD IPA [56, 58]. Up to our best knowledge, this is the first class of AET functions in case of PD IPAs. Furthermore, we present the complexity analysis of this new PD IPA based on the new class of AET functions. We provide polynomial iteration complexity in the size of the problem, in the accuracy parameter, the starting point's duality gap and in the parameter κ.

2.1 New class of AET functions for primal-dual interior-point algorithm

In this section, we introduce a new class of AET functions that defines different search directions for a PD IPA.

Definition 2.1.1. (Definition 2.4 in [58]) Let $\varphi : (\xi^2, \infty) \to \mathbb{R}$ be a continuously differentiable, invertible function, such that $\varphi'(t) > 0 \; \forall t > \xi^2$, where $0 \leq \xi < 1$. All functions φ satisfying the conditions below belong to the new class.

AET1. $\exists\, \bar{c}_1 \in \mathbb{R}_+$, such that
$$\left| \frac{\varphi(1)-\varphi(t^2)}{2t(1-t^2)\varphi'(t^2)} \right| \le \bar{c}_1,$$

AET2. $\exists\, \bar{c}_2 \in \mathbb{R}_+$, such that
$$\left| \frac{4t^2\varphi'(t^2)\left[(1-t^2)\varphi'(t^2)-\varphi(1)+\varphi(t^2)\right]}{(\varphi(1)-\varphi(t^2))^2} \right| \le \bar{c}_2,$$

AET3. $\exists\, \bar{c}_3 \in \mathbb{R}_+$ such that the inequality
$$\begin{aligned} 4t^2(\varphi(1)-\varphi(t^2))\varphi'(t^2) \quad &- \quad \bar{c}_3\left(\varphi(1)-\varphi(t^2)\right)^2 \le 4t^2(1-t^2)\left(\varphi'(t^2)\right)^2 \\ &\le \quad 4t^2(\varphi(1)-\varphi(t^2))\varphi'(t^2)+\left(\varphi(1)-\varphi(t^2)\right)^2 \end{aligned}$$

holds in all three conditions for all $t>\xi$.

It should be mentioned that the point $t=1$ is removable discontinuity of the functions given in AET1 and AET2, respectively.

Remark 2.1.1. (Remark 2.1 in [58]) Condition AET1 can be rewritten in the following form:
$$\left| \frac{\frac{\varphi(1)-\varphi(t^2)}{2t\varphi'(t^2)}}{1-t^2} \right| \le \bar{c}_1.$$

Conditions AET2 and AET3 can be rewritten in a similar way and these forms show that the functions belonging to this class of AET functions have transformations with growth proportional to the identical function. Hence, we may call these functions *AET functions with bounded proportional growth rate.*

Let the function $f:(\xi,\infty)\to\mathbb{R}$ be as:
$$f(t) := \frac{\varphi(1)-\varphi(t^2)}{t\varphi'(t^2)}, \tag{2.1.1}$$

Using (2.1.1) we can reformulate the properties given in Definition 2.1.1.

Proposition 2.1.1. *(Proposition 2.1 in [58]) Let $\varphi : (\xi^2, \infty) \to \mathbb{R}$ be a continuously differentiable, invertible function, such that $\varphi'(t) > 0 \; \forall t > \xi^2$, where $0 \le \xi < 1$. Consider the function f given in (2.1.1). The conditions given in Definition 2.1.1 can be formulated in the following equivalent form:*

AETa. $\exists\, \bar{c}_1 \in \mathbb{R}_+$*, such that* $g(t) = \frac{f(t)}{2(1-t^2)}$ *and* $|g(t)| \le \bar{c}_1$*, holds for all* $t > \xi$*;*

AETb. $\exists\, \bar{c}_2 \in \mathbb{R}_+$*, such that* $h(t) = \frac{4(1-t^2-tf(t))}{f(t)^2} = \frac{1-2tg(t)}{(1-t^2)g(t)^2}$ *and* $|h(t)| \le \bar{c}_2$*, holds for all* $t > \xi$*;*

AETc. $\exists\, \bar{c}_3 \in \mathbb{R}_+$ *such that the inequality*

$$tf(t) - \bar{c}_3 \frac{f(t)^2}{4} \le 1 - t^2 \le tf(t) + \frac{f(t)^2}{4}$$

holds for all $t > \xi$*.*

Proof. Considering the function f introduced in (2.1.1), it is easy to see that conditions AET1-3 can be reformulated as AETa-c. □

Remark 2.1.2. (Remark 2.2 in [58]) The values of the parameters $\bar{c}_1$, $\bar{c}_2$ and $\bar{c}_3$ will have influence on the well-definedness of the algorithm. For this, we will give a relation between these parameters in Section 2.3.

We show some examples for φ belonging to our class of AET functions in Table 2.1.1 with the value of ξ and the values $\bar{c}_1, \bar{c}_2$ and $\bar{c}_3$. Table 1.4.1 shows that almost all AET functions that can be found in the literature belong to our class of AET. This shows that some of the short-step IPAs proposed in the literature e.g. [7, 26] are special cases of our unified approach. However, there is an exception, in case of the function $\varphi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}$, where condition AET3 is not satisfied.

It is important to mention the values we gave in Table 2.1.1 define a short-step IPA that is well defined and for which we could prove the polynomial iteration complexity in the size of the problem, handicap of the matrix, the starting point's duality gap and in the accuracy parameter. However, there are several other acceptable values for these parameters.

Table 2.1.1: Examples for φ belonging to the new class of AET functions

$\varphi(t)$	ξ	$\bar{c}_1$	$\bar{c}_2$	$\bar{c}_3$
t	0.25	2	6	1
$\sqrt{t}$	0	2	6	1
$t-\sqrt{t}$	0.7	2	6	1
$t^2-t+\sqrt{t}$	0	2	8	8
$t^2+\sqrt{t}$	0	2	6	8

Let us take into consideration the case of $\varphi(t)=t$. This function is not a member function of our class of AET functions if $\xi=0$, since AET1 would not be satisfied. This remark is similar to the case of $\varphi(t)=t-\sqrt{t}$.

Remark 2.1.3. (Remark 2.4 in [58]) By using (2.1.1) and (1.5.2), we have

$$\psi'(t) = -f(t). \tag{2.1.2}$$

Using (2.1.2) we can check whether the AET functions corresponding to kernel functions are members of our class of AET functions. It is important to mention that for a function φ from our class of AET functions we can easily check whether the corresponding kernel function satisfies the conditions K1-3 from Definition 1.5.1 and EK1-4 from Definition 1.5.3.

We present an example of eligible kernel function for which the corresponding AET function does not belong to our class.

Example 2.1.1. (Example 2.2 in [58]) Let us consider the function $\psi(t)=\frac{1}{2}\left(t-\frac{1}{t}\right)^2$. It is proven in [72] that it is an eligible kernel function. It is also a self-regular kernel function. From (1.5.2) we get

$$f(t)=\frac{\varphi(1)-\varphi(t^2)}{t\varphi'(t^2)}=-\psi'(t)=\frac{1}{t^3}-t.$$

After some calculation it is easy to see that this is equivalent to

$$\varphi(1)-\varphi(t^2)=\left(\frac{1}{t^2}-t^2\right)\varphi'(t^2),$$

therefore we have

$$\varphi'(t^2) = \frac{t^2}{1-t^4}\left(\varphi(1) - \varphi(t^2)\right). \tag{2.1.3}$$

It is easy to see that $f(t) = \frac{1}{t^3} - t$ does not satisfy condition AETb from Proposition 2.1.1. This means that the AET function corresponding to this eligible kernel function is not a member of our class of functions.

Example 2.1.2. From Table 2.1.1 we can see that $\varphi(t) = t^2 - t + \sqrt{t}$ is a member function of our new class of AET functions, that has an inflection point. The corresponding kernel function for this φ is not an eligible kernel function.

2.2 New short-step interior-point algorithm for solving $P_*(\kappa)$-linear complementarity problems

Let us consider the (1.4.2), where $\mathbf{a}_\varphi$ depends on a member function φ of our class of AET functions.

In this case, consider the centrality measure $\delta : \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathbb{R}_+ \to \mathbb{R} \cup \{\infty\}$ as

$$\delta(\mathbf{x}, \mathbf{s}; \mu) := \frac{\|\mathbf{p}_\varphi\|}{2}, \tag{2.2.1}$$

where $\mathbf{p}_\varphi$ is given in (1.4.8).

Let us consider the τ-neighborhood as the following:

$$\mathcal{N}_2(\tau, \mu) := \{(\mathbf{x}, \mathbf{s}) \in \mathcal{F}^+ : \delta(\mathbf{x}, \mathbf{s}; \mu) \le \tau\}, \tag{2.2.2}$$

where $\delta(\mathbf{x}, \mathbf{s}; \mu)$ is given in (2.2.1), $\mu > 0$ is fixed and τ is a threshold parameter.

In Algorithm 1 we define a PD IPA for solving $P_*(\kappa)$-LCPs with a function φ as an input.

Algorithm 1 : PD IPA for $P_*(\kappa)$-LCPs based on a new class of AETs [58]

Input:
Let φ be a function that satisfies the properties given in Definition 2.1.1;
Let $\epsilon > 0$ be the accuracy parameter;
Let $0 < \theta < 1$ be the update parameter ;
Let τ be the proximity parameter;
Assume a known upper bound κ of the handicap $\hat{\kappa}(M)$ is given;
Assume that for $(\mathbf{x}^0, \mathbf{s}^0)$ we have $\left(\mathbf{x}^0\right)^T \mathbf{s}^0 = n\mu^0$ with $\mu^0 > 0$, such that $\delta(\mathbf{x}^0, \mathbf{s}^0; \mu^0) \leq \tau$;
Suppose that $\mathbf{v}^0 > \xi \mathbf{e}$.
Iteration:
begin
$k := 0$;
while $\left(\mathbf{x}^k\right)^T \mathbf{s}^k > \epsilon$ **do**
begin
(determination of the search directions)
compute $(\Delta x^k, \Delta s^k)$ from (1.4.2) with φ from Definition 2.1.1;
let $\mathbf{x}^k := \mathbf{x}^k + \Delta x^k$ and $\mathbf{s}^k := \mathbf{s}^k + \Delta s^k$, a full Newton-step update;
(update of the parameter μ)
$\mu^{k+1} := (1-\theta)\mu^k$;
$k := k+1$;

end
end

Remark 2.2.1. (Remark 3.1 in [58]) We will give a default value for τ and θ during the complexity analysis of the Algorithm 1.

The complexity analysis of the Algorithm 1 is given in the next section.

2.3 Complexity analysis of Algorithm 1

First, let us denote vectors $\mathbf{x}^+ = \mathbf{x} + \Delta\mathbf{x}$ and $\mathbf{s}^+ = \mathbf{s} + \Delta\mathbf{s}$. In the next lemma we prove the strict feasibility of the full-Newton step.

Lemma 2.3.1. *(Lemma 3.1 in [58]) Let $(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{F}^+$ be given, such that $\delta(\boldsymbol{x}, \boldsymbol{s}; \mu) < \frac{1}{\sqrt{1+4\kappa}}$ and $\mathbf{v} > \xi\mathbf{e}$. Let φ be a function satisfying AET3 of Definition 2.1.1. Then, we have $(\mathbf{x}^+, \mathbf{s}^+) \in \mathcal{F}^+$.*

Proof. Using AET3 we have

$$4t^2(\varphi(1)-\varphi(t^2))\varphi'(t^2)+\left(\varphi(1)-\varphi(t^2)\right)^2 \geq 4t^2(1-t^2)\left(\varphi'(t^2)\right)^2, \tag{2.3.1}$$

where $t>\xi$. For each $0\leq\alpha\leq 1$ let us denote $\mathbf{x}(\alpha)=\mathbf{x}+\alpha\Delta\mathbf{x}$ and $\mathbf{s}(\alpha)=\mathbf{s}+\alpha\Delta\mathbf{s}$. Then, we have

$$\begin{aligned}
\mathbf{x}(\alpha)\mathbf{s}(\alpha) &= (\mathbf{x}+\alpha\Delta\mathbf{x})(\mathbf{s}+\alpha\Delta\mathbf{s}) = \mathbf{xs}+\alpha(\mathbf{s}\Delta\mathbf{x}+\mathbf{x}\Delta\mathbf{s})+\alpha^2\Delta\mathbf{x}\Delta\mathbf{s}\\
&= \mu\mathbf{v}^2+\mu\mathbf{v}\alpha(\mathbf{d}_x+\mathbf{d}_s)+\alpha^2\mu\mathbf{d}_x\mathbf{d}_s\\
&= \mu((1-\alpha)\mathbf{v}^2+\alpha(\mathbf{v}^2+\mathbf{v}\mathbf{p}_\varphi+\alpha\mathbf{d}_x\mathbf{d}_s)).
\end{aligned} \tag{2.3.2}$$

We have to show that $\alpha(\mathbf{v}^2+\mathbf{v}\mathbf{p}_\varphi+\alpha\mathbf{d}_x\mathbf{d}_s)>0$. We have $\mathbf{p}_\varphi=\mathbf{d}_x+\mathbf{d}_s$. Let us denote $\mathbf{q}_\varphi=\mathbf{d}_x-\mathbf{d}_s$. With these equations it is easy to see that we obtain

$$\mathbf{d}_x\mathbf{d}_s=\frac{\mathbf{p}_\varphi^2-\mathbf{q}_\varphi^2}{4}. \tag{2.3.3}$$

Using $\mathbf{p}_\varphi$ given in (1.4.8) and (2.3.1) we have

$$\mathbf{v}\mathbf{p}_\varphi+\frac{\mathbf{p}_\varphi^2}{4}\geq\mathbf{e}-\mathbf{v}^2, \tag{2.3.4}$$

from this equation we get

$$\mathbf{v}^2+\mathbf{v}\mathbf{p}_\varphi\geq\mathbf{e}-\frac{\mathbf{p}_\varphi^2}{4}. \tag{2.3.5}$$

Using (2.3.2), (2.3.3) and (2.3.5) we obtain

$$\begin{aligned}
\frac{\mathbf{x}(\alpha)\mathbf{s}(\alpha)}{\mu} &= (1-\alpha)\mathbf{v}^2+\alpha\left(\mathbf{v}^2+\mathbf{v}\mathbf{p}_\varphi+\alpha\frac{\mathbf{p}_\varphi^2}{4}-\alpha\frac{\mathbf{q}_\varphi^2}{4}\right)\\
&\geq (1-\alpha)\mathbf{v}^2+\alpha\left(\mathbf{e}-(1-\alpha)\frac{\mathbf{p}_\varphi^2}{4}-\alpha\frac{\mathbf{q}_\varphi^2}{4}\right).
\end{aligned}$$

From [25] we considered $\left\|\mathbf{q}_\varphi\right\|$ and obtained

$$\left\|\mathbf{q}_\varphi\right\|\leq\sqrt{1+4\kappa}\,\left\|\mathbf{p}_\varphi\right\|=2\sqrt{1+4\kappa}\,\delta. \tag{2.3.6}$$

Moreover, we have

$$\left\|(1-\alpha)\frac{\mathbf{p}_\varphi^2}{4}-\alpha\frac{\mathbf{q}_\varphi^2}{4}\right\|_\infty \le (1-\alpha)\frac{\left\|\mathbf{p}_\varphi\right\|^2}{4}+\alpha\frac{\left\|\mathbf{q}_\varphi\right\|^2}{4}$$
$$\le (1-\alpha)\frac{\left\|\mathbf{p}_\varphi\right\|^2}{4}+\alpha(1+4\kappa)\frac{\left\|\mathbf{p}_\varphi\right\|^2}{4}$$
$$=(1+4\alpha\kappa)\delta^2 \le (1+4\kappa)\delta^2.$$

Therefore, $\left\|(1-\alpha)\frac{\mathbf{p}_\varphi^2}{4}-\alpha\frac{\mathbf{q}_\varphi^2}{4}\right\|_\infty < 1$ holds if $\delta(\mathbf{x},\mathbf{s};\mu) < \frac{1}{\sqrt{1+4\kappa}}$ is satisfied. Hence, for each $0 \le \alpha \le 1$ vector $\mathbf{x}(\alpha)\mathbf{s}(\alpha)$ is strictly positive. This means that $\mathbf{x}(\alpha)$ and $\mathbf{s}(\alpha)$ do not change sign on $0 \le \alpha \le 1$ and since $\mathbf{x}(0)=\mathbf{x}>\mathbf{0}$ and $\mathbf{s}(0)=\mathbf{s}>\mathbf{0}$, we obtain that $\mathbf{x}(1)=\mathbf{x}^+>\mathbf{0}$ and $\mathbf{s}(1)=\mathbf{s}^+>\mathbf{0}$. □

In the second lemma we prove the quadratic convergence of the full-Newton step.

Lemma 2.3.2. *(Lemma 3.2 in [58]) Let $(\boldsymbol{x},\boldsymbol{s}) \in \mathcal{F}^+$, $\mathbf{v} > \xi\mathbf{e}$ and $\bar{\mathbf{v}} = \sqrt{\frac{\boldsymbol{x}^+\boldsymbol{s}^+}{\mu}}$ be given and suppose that $\delta(\mathbf{x},\mathbf{s};\mu) < \sqrt{\frac{1-\xi^2}{1+4\kappa}}$. Let φ satisfy conditions AET1-3 of Definition 2.1.1 with constants $\bar{c}_1, \bar{c}_2$ and $\bar{c}_3 \in \mathbb{R}_+$. Then, after a primal-dual Newton barrier step we have $\bar{\mathbf{v}} > \xi\mathbf{e}$ and*

$$\delta(\boldsymbol{x}^+,\boldsymbol{s}^+;\mu) \le \bar{c}_1(\bar{c}_2+2+4\kappa)\delta(\boldsymbol{x},\boldsymbol{s};\mu)^2.$$

Proof. From condition AETc of Proposition 2.1.1 and Lemma 5.6 in [12], we obtain

$$\min(\bar{\mathbf{v}}) \ge \sqrt{1-(1+4\kappa)\delta^2}.$$

Using the above inequality and $\delta < \sqrt{\frac{1-\xi^2}{1+4\kappa}}$ it can be seen that $\bar{\mathbf{v}} > \xi\mathbf{e}$. Consider AETa and AETb of Proposition 2.1.1 and from (2.2.1) we obtain

$$\delta(\mathbf{x}^+,\mathbf{s}^+;\mu) = \left\|(\mathbf{e}-\bar{\mathbf{v}}^2)g(\bar{\mathbf{v}})\right\|, \tag{2.3.7}$$

where the function g is given in Proposition 2.1.1. Let us assume that $\exists \bar{c}_1 \in \mathbb{R}_+$, for which $|g(t)| \le \bar{c}_1$ for all $t > \xi$, in this case we have

$$\delta(\mathbf{x}^+,\mathbf{s}^+;\mu) \le \bar{c}_1\left\|\mathbf{e}-\bar{\mathbf{v}}^2\right\|. \tag{2.3.8}$$

Using (2.3.2) we have

$$\left\|\mathbf{e}-\bar{\mathbf{v}}^2\right\| = \left\|\mathbf{e}-\frac{\mathbf{x}^+\mathbf{s}^+}{\mu}\right\| = \left\|\mathbf{e}-\mathbf{v}^2-\mathbf{v}\mathbf{p}_\varphi-\frac{\mathbf{p}_\varphi^2}{4}+\frac{\mathbf{q}_\varphi^2}{4}\right\|. \tag{2.3.9}$$

From condition AETb of Proposition 2.1.1 we obtain

$$\mathbf{e}-\mathbf{v}^2-\mathbf{v}\mathbf{p}_\varphi \quad = \quad h(\mathbf{v})\frac{\mathbf{p}_\varphi^2}{4}. \tag{2.3.10}$$

Furthermore, we obtain

$$\left\|\mathbf{e}-\bar{\mathbf{v}}^2\right\| \quad \leq \quad \left\|\mathbf{e}-\mathbf{v}^2-\mathbf{v}\mathbf{p}_\varphi\right\| + \left\|\frac{\mathbf{p}_\varphi^2}{4}\right\| + \left\|\frac{\mathbf{q}_v^2}{4}\right\| \leq (2+\bar{c}_2+4\kappa)\delta^2. \tag{2.3.11}$$

by using Lemma 5.4 from [25], (2.3.9) and (3.1.20).

Finally, using (2.3.8) and (3.1.21) we derive

$$\delta(\mathbf{x}^+,\mathbf{s}^+;\mu) \leq \bar{c}_1(\bar{c}_2+2+4\kappa)\delta(\mathbf{x},\mathbf{s};\mu)^2,$$

which proves the lemma.

□

In Lemma 3.3 we gave an upper bound on the duality gap after a full-Newton step.

Lemma 2.3.3. *(Lemma 3.3 in [58]) Let* $\delta := \delta(\boldsymbol{x},\boldsymbol{s};\mu)$ *and suppose that the vectors* $\boldsymbol{x}^+$ *and* $\boldsymbol{s}^+$ *are obtained using a full-Newton step, thus,* $\boldsymbol{x}^+ = \boldsymbol{x}+\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^+ = \boldsymbol{s}+\Delta\boldsymbol{s}$*. Let* φ *be a function satisfying AET3 of Definition 2.1.1 with* $\bar{c}_3 \in \mathbb{R}_+$*. Then, we have*

$$\left(\boldsymbol{x}^+\right)^T\boldsymbol{s}^+ \leq \mu\left(n+(\bar{c}_3+1)\delta^2\right).$$

Proof. It is important to mention that we use only the first inequality of AET3 from Definition 2.1.1.

$$\begin{aligned} 4t^2(\varphi(1)-\varphi(t^2))\varphi'(t^2) \quad &- \quad \bar{c}_3\left(\varphi(1)-\varphi(t^2)\right)^2 \\ &\leq \quad 4t^2(1-t^2)\left(\varphi'(t^2)\right)^2, \end{aligned} \tag{2.3.12}$$

where $t > \xi$. Using (1.4.8) and (2.3.12) we obtain

$$\mathbf{v}\mathbf{p}_{\varphi} - \bar{c}_3 \frac{\mathbf{p}_{\varphi}^2}{4} \leq \mathbf{e} - \mathbf{v}^2. \tag{2.3.13}$$

From (2.3.2) and (2.3.13) we have

$$\frac{1}{\mu}\mathbf{x}^+\mathbf{s}^+ = \mathbf{v}^2 + \mathbf{v}\mathbf{p}_{\varphi} + \mathbf{d}_x\mathbf{d}_s \leq \mathbf{e} + \frac{\bar{c}_3}{4}\mathbf{p}_{\varphi}^2 + \mathbf{d}_x\mathbf{d}_s.$$

After some calculations we derive

$$\begin{aligned}
\left(\mathbf{x}^+\right)^T \mathbf{s}^+ = \mathbf{e}^T\left(\mathbf{x}^+\mathbf{s}^+\right) &\leq \mu\left(\mathbf{e}^T\mathbf{e} + \frac{\bar{c}_3}{4}\mathbf{e}^T\mathbf{p}_{\varphi}^2 + \mathbf{e}^T\mathbf{d}_x\mathbf{d}_s\right) \\
&= \mu\left(n + \frac{\bar{c}_3}{4}\left\|\mathbf{p}_{\varphi}\right\|^2 + \mathbf{d}_x^T\mathbf{d}_s\right) \\
&\leq \mu\left(n + \bar{c}_3\delta^2 + \delta^2\right) = \mu\left(n + (\bar{c}_3+1)\delta^2\right).
\end{aligned}$$

And finally, the last inequality holds, since

$$\mathbf{d}_x^T\mathbf{d}_s = \frac{\|\mathbf{d}_x + \mathbf{d}_s\|^2 - \|\mathbf{d}_x - \mathbf{d}_s\|^2}{4} \leq \frac{\|\mathbf{d}_x + \mathbf{d}_s\|^2}{4} = \frac{\left\|\mathbf{p}_{\varphi}\right\|^2}{4},$$

which proves the result. □

In the following lemma we show the effect of the update of the parameter μ on the proximity measure.

Lemma 2.3.4. *(Lemma 3.4 in [58]) Let* $\mathbf{v} > \xi\mathbf{e}$ *and* $\mathbf{v}^+ = \sqrt{\frac{\boldsymbol{x}^+\boldsymbol{s}^+}{\mu^+}}$*, where* $\mu^+ = (1-\theta)\mu$ *and let* $\eta = \sqrt{1-\theta}$*. Assume that* $\delta = \delta(\mathbf{x},\mathbf{s};\mu) < \sqrt{\frac{1-\xi^2}{1+4\kappa}}$*. Let* φ *be a function satisfying conditions AET1-3 of Definition 2.1.1 with* $\bar{c}_1, \bar{c}_2$ *and* $\bar{c}_3 \in \mathbb{R}_+$*. Then, after a full-Newton step we have* $\mathbf{v}^+ > \xi\mathbf{e}$ *and*

$$\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu^+) \leq \frac{\bar{c}_1}{\eta^2}\left(\theta\sqrt{n} + ((\bar{c}_2+2) + 4\kappa)\delta^2\right).$$

Proof. From Lemma 2.3.2 we know that $\bar{\mathbf{v}} > \xi\mathbf{e}$. Using this inequality and the fact that $0 < \theta < 1$ we have $\mathbf{v}^+ > \xi\mathbf{e}$. Using AETa and AETb of Proposition 2.1.1 and the proximity measure given in (2.2.1), we obtain

$$\delta(\mathbf{x}^+, \mathbf{s}^+; \mu^+) = \left\|(\mathbf{e} - (\mathbf{v}^+)^2)g(\mathbf{v}^+)\right\|.$$

Using AETa of Proposition 2.1.1 we have

$$\delta(\mathbf{x}^+,\mathbf{s}^+;\mu^+) \leq \bar{c}_1 \left\| \mathbf{e} - (\mathbf{v}^+)^2 \right\|. \tag{2.3.14}$$

It can be seen that

$$\begin{aligned} \left\| \mathbf{e} - (\mathbf{v}^+)^2 \right\| &= \left\| \mathbf{e} - \frac{1}{\eta^2} \frac{\mathbf{x}^+\mathbf{s}^+}{\mu} \right\| = \left\| \mathbf{e} - \frac{1}{\eta^2} \left(\mathbf{v}^2 + \mathbf{v}\mathbf{p}_\varphi + \frac{\mathbf{p}_\varphi^2}{4} - \frac{\mathbf{q}_\varphi^2}{4} \right) \right\| \\ &= \frac{1}{\eta^2} \left\| \eta^2 \mathbf{e} - \left(\mathbf{v}^2 + \mathbf{v}\mathbf{p}_\varphi + \frac{\mathbf{p}_\varphi^2}{4} - \frac{\mathbf{q}_\varphi^2}{4} \right) \right\| \\ &= \frac{1}{\eta^2} \left\| -\theta \mathbf{e} + \mathbf{e} - \mathbf{v}^2 - \mathbf{v}\mathbf{p}_\varphi - \frac{\mathbf{p}_\varphi^2}{4} + \frac{\mathbf{q}_\varphi^2}{4} \right\|. \end{aligned} \tag{2.3.15}$$

Finally, using (3.1.20), (2.3.14), (2.3.15) and condition AETb of Proposition 2.1.1 we have

$$\begin{aligned} \delta(\mathbf{x}^+,\mathbf{s}^+;\mu^+) &\leq \frac{\bar{c}_1}{\eta^2} \left(\|\theta \mathbf{e}\| + \left\| \mathbf{e} - \mathbf{v}^2 - \mathbf{v}\mathbf{p}_\varphi \right\| + \left\| \frac{\mathbf{p}_\varphi^2}{4} \right\| + \left\| \frac{\mathbf{q}_\varphi^2}{4} \right\| \right) \\ &\leq \frac{\bar{c}_1}{\eta^2} \left(\theta\sqrt{n} + ((\bar{c}_2 + 2) + 4\kappa)\delta^2 \right), \end{aligned} \tag{2.3.16}$$

which proves the lemma. □

In Lemma 3.5 in [58] we gave exact values of parameters θ and τ and with those values it is proven that the IPA, which is using a function from our class of AET functions as an input, is well defined.

Lemma 2.3.5. *(Lemma 3.5 in [58]) Let φ be a function satisfying AET1-3 of Definition 2.1.1. Assume that $n \geq 1$, $\theta = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)\sqrt{n}}$ and $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$. Suppose that $\delta(\mathbf{x},\mathbf{s};\mu) \leq \tau$. If $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$ and $\bar{c}_2 \geq 6$, then we have*

$$\delta(\boldsymbol{x}^+,\boldsymbol{s}^+;\mu^+) \leq \tau.$$

Hence, the PD IPA defined in Algorithm 1 is well defined.

Proof. First, consider $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$ and $\bar{c}_2 \geq 6 > \frac{1}{2}$. Then, $\frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)} < \frac{\sqrt{1-\xi^2}}{2+\kappa} < \sqrt{\frac{1-\xi^2}{1+4\kappa}} < \frac{1}{\sqrt{1+4\kappa}}$. Using the previous inequality and assuming the conditions AET1-3 of Definition 2.1.1 are satisfied, then using Lemma 2.3.1, we obtain that $(\mathbf{x}^+,\mathbf{s}^+) \in \mathcal{F}^+$.

From Lemma 2.3.4, we get

$$\delta(\mathbf{x}^+, \mathbf{s}^+; \mu^+) \;\leq\; \frac{\bar{c}_1}{\eta^2}\left(\theta\sqrt{n} + ((2+\bar{c}_2)+4\kappa)\delta^2\right). \tag{2.3.17}$$

Let us examine the case where $\theta = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)\sqrt{n}}$ and $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$. It is easy to see that

$$\theta\sqrt{n} = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)} = \frac{4}{25}\tau. \tag{2.3.18}$$

Using $n \geq 1$ and $\kappa \geq 0$, we obtain

$$\frac{1}{1-\theta} \leq \frac{1}{1-\frac{2}{50\bar{c}_2}} = \frac{25\bar{c}_2}{25\bar{c}_2 - 1}. \tag{2.3.19}$$

Furthermore, considering $\delta \leq \tau$ and from $\eta = \sqrt{1-\theta}$, (2.3.17), (2.3.18), and (2.3.19), we have

$$\begin{aligned}\delta(\mathbf{x}^+, \mathbf{s}^+; \mu^+) \;&\leq\; \bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25}\tau + ((2+\bar{c}_2)+4\kappa)\tau^2\right)\\ &=\; \bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25} + ((2+\bar{c}_2)+4\kappa)\tau\right)\tau.\end{aligned} \tag{2.3.20}$$

Finally, we have to show that $\delta(\mathbf{x}^+, \mathbf{s}^+; \mu^+) \leq \tau$. We need to prove that

$$\bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25} + ((2+\bar{c}_2)+4\kappa)\tau\right) \leq 1. \tag{2.3.21}$$

By using $\kappa \geq 0$ and $\delta \leq \tau \leq \frac{1}{2\bar{c}_2(2+\kappa)}$ we obtain

$$\begin{aligned}&\bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25} + ((2+\bar{c}_2)+4\kappa)\tau\right) \leq \bar{c}_1\frac{25}{25\bar{c}_2-1}\left(\frac{4}{25} + \frac{2+\bar{c}_2+4\kappa}{2\bar{c}_2(2+\kappa)}\right)\\ =\;& \bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25} + \frac{\bar{c}_2-6}{2\bar{c}_2(2+\kappa)} + \frac{4(2+\kappa)}{2\bar{c}_2(2+\kappa)}\right)\\ \leq\;& \bar{c}_1\frac{25\bar{c}_2}{25\bar{c}_2-1}\left(\frac{4}{25} + \frac{\bar{c}_2-6}{4\bar{c}_2} + \frac{2}{\bar{c}_2}\right) = \frac{\bar{c}_1(41\bar{c}_2+50)}{100\bar{c}_2-4}.\end{aligned} \tag{2.3.22}$$

Considering that $\bar{c}_2 \geq 6 > \frac{1}{25}$ and $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$, we get that $\delta(\mathbf{x}^+, \mathbf{s}^+; \mu^+) \leq \tau$, which proves the lemma. $\square$

Remark 2.3.1. (Remark 3.2 in [58]) It is important to note that $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$ and $\bar{c}_2 \geq 6$ are satisfied for the functions that we gave in Table 2.1.1.

Finally, the last lemma gives an upper bound on the iteration number.

Lemma 2.3.6. *(Lemma 3.6 in [58]) Let φ be a function satisfying AET1-3 of Definition 2.1.1. Let $n \geq 1$, $\theta = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)\sqrt{n}}$, $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$, $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$, $\bar{c}_2 \geq 6$, and $\bar{c}_3 \leq 16\bar{c}_2^2 - 1$. Assume that the pair $(\boldsymbol{x}^0, \boldsymbol{s}^0) \in \mathcal{F}^+$, $\mu^0 = \frac{(\boldsymbol{x}^0)^T \boldsymbol{s}^0}{n}$ and $\delta(\boldsymbol{x}^0, \boldsymbol{s}^0; \mu^0) \leq \tau$. Let $\boldsymbol{x}^k$ and $\boldsymbol{s}^k$ be the two vectors obtained by the IPA given in Algorithm 1 after k iterations. Then,*

$$k \geq \left\lceil \frac{1}{\theta} \log \frac{\mu^0(n+1)}{\varepsilon} \right\rceil$$

implies $(\boldsymbol{x}^k)^T \boldsymbol{s}^k \leq \varepsilon$.

Proof. By using Lemma 2.3.3 and $\bar{c}_3 \leq 16\bar{c}_2^2 - 1$ we obtain

$$\begin{aligned} \left(\mathbf{x}^k\right)^T \mathbf{s}^k &\leq \mu^k \left(n + (\bar{c}_3 + 1) \cdot \frac{1}{(2\bar{c}_2(2+\kappa))^2} \right) \\ &= (1-\theta)^k \mu^0 \left(n + (\bar{c}_3 + 1) \cdot \frac{1}{(2\bar{c}_2(2+\kappa))^2} \right) \\ &\leq (1-\theta)^k \mu^0 (n+1). \end{aligned} \tag{2.3.23}$$

It can be seen that condition $(\mathbf{x}^k)^T \mathbf{s}^k \leq \varepsilon$ holds if

$$(1-\theta)^k \mu^0 (n+1) \leq \varepsilon. \tag{2.3.24}$$

Taking the logarithm of both sides of (2.3.24), we get

$$k \log(1-\theta) + \log\left(\mu^0 (n+1)\right) \leq \log \varepsilon.$$

Finally, taking into consideration that $-\log(1-\theta) \geq \theta$, we have

$$k\theta \geq \log\left(\mu^0 (n+1)\right) - \log \varepsilon = \log \frac{\mu^0(n+1)}{\varepsilon},$$

this proves the last lemma. □

Remark 2.3.2. (Remark 3.3 in [58]) For all AET functions given in Table 2.1.1 it is easy to see that condition $\bar{c}_3 \leq 16\bar{c}_2^2 - 1$ is satisfied.

Theorem 2.3.1. *(Theorem 3.1 in [58]) Let φ be a function satisfying AET1-3 of Definition 2.1.1. Consider $n \geq 1$, $\theta = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)\sqrt{n}}$ and $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$. If $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$, $\bar{c}_2 \geq 6$ and $\bar{c}_3 \leq 16\bar{c}_2^2-1$, then the IPA given in Algorithm 1 requires no more than*

$$\mathcal{O}\left((2+\kappa)\sqrt{n}\log\frac{(n+1)\mu^0}{\epsilon}\right)$$

iterations.

Corollary 2.3.1. *(Corollary 3.1 in [58]) Let φ be a function satisfying AET1-3 of Definition 2.1.1. Consider $n \geq 1$, $\theta = \frac{2\sqrt{1-\xi^2}}{25\bar{c}_2(2+\kappa)\sqrt{n}}$ and $\tau = \frac{\sqrt{1-\xi^2}}{2\bar{c}_2(2+\kappa)}$. If $\bar{c}_1 \leq \frac{100\bar{c}_2-4}{41\bar{c}_2+50}$, $\bar{c}_2 \geq 6$ and $\bar{c}_3 \leq 16\bar{c}_2^2-1$, then the IPA given in Algorithm 1 has polynomial iteration complexity in the size of the problem, in the starting point's duality gap μ^0, the accuracy parameter ε and in the parameter κ.*

In the following chapter we present a new class of AET functions for PC IPAs for solving $P_*(\kappa)$-LCPs. We present the PC IPA based on a special function belonging to the new class of AET functions.

Chapter 3

Predictor-corrector interior-point algorithm based on a new class of AET functions

In this chapter, we define a new class of AET functions for PC IPAs for solving $P_*(\kappa)$-LCPs and we give the complexity analysis of the introduced PC IPA [60]. A function being member of the new class of AET functions is an input data. We show that the PC IPA using any function of the new class of AET functions has polynomial iteration complexity in the size of the problem, starting point's duality gap, accuracy parameter and in the parameter κ. Furthermore, in Section 3.2 we show the main steps of the analysis for a special AET function belonging to the new class of AET functions. This result can be found in [55, 59].

3.1 New predictor-corrector interior-point algorithm

In this section, we introduce the new class of AET functions that will be used as an input in the PC IPA in case of $P_*(\kappa)$-LCPs.

3.1.1 New class of AET functions for predictor-corrector interior-point algorithm

We define a class of AET functions as follows.

Definition 3.1.1. (Definition 2.3 in [60]) Let $\varphi : (\xi^2, \infty) \to \mathbb{R}$ be a continuously differentiable, invertible function, such that $\varphi'(t) > 0$, $\forall t > \xi^2$, where $0 \leq \xi \leq \frac{1}{2}$. All functions φ satisfying the following conditions belong to our class of AET functions. There exist the positive real numbers $c_0 \geq \frac{1}{4}$, $c_1, c_2, c > 0$ such that

AET1.

$$c_0 \left| t(1-t)\varphi'(t^2) \right| \leq \left| \varphi(1) - \varphi(t^2) \right| \leq 2c_1(1+t) \left| t(1-t)\varphi'(t^2) \right|$$

holds for all $t > \xi$,

AET2.

$$\begin{aligned} 4t^2\varphi'(t^2)(\varphi(1)-\varphi(t^2)) &- c_2(\varphi(1)-\varphi(t^2))^2 \leq 4t^2(1-t^2)\left(\varphi'(t^2)\right)^2 \\ &\leq 4t^2\varphi'(t^2)(\varphi(1)-\varphi(t^2)) + (\varphi(1)-\varphi(t^2))^2 \end{aligned}$$

holds for all $t > \xi$,

AET3.

$$\lim_{t\to\infty} \frac{\varphi(1)-\varphi(t^2)}{t^2\varphi'(t^2)} = -c.$$

Remark 3.1.1. Note that conditions AET2 and AET3 in Definition 2.1.1 from Chapter 2 are included in condition AET2 of Definition 3.1.1. The parameter $\bar{c}_3$ given in Chapter 2 is in this case c_2. The $\bar{c}_1$ from Chapter 2 is the same as c_1 in Definition 3.1.1. Furthermore, we introduce $c_3 = \max\{1, c_2\}$.

Let us consider the function f that was defined in (2.1.1). Using (2.1.1) we can reformulate the conditions from Definition 3.1.1.

Proposition 3.1.1. *(Proposition 2.1 in [60]) Consider the function f given in (2.1.1). The conditions given in Definition 3.1.1 can be formulated in the following way. There exist positive constants $c_0 \geq \frac{1}{4}$, $c_1, c_2, c > 0$ such that*

AETa. $c_0|1-t| \leq |f(t)| \leq 2c_1|1-t^2|$, *holds for all $t > \xi$,* *AETb.* $-c_2\frac{f(t)^2}{4} \leq 1-t^2-tf(t) \leq \frac{f(t)^2}{4}$, *holds for all* $t > \xi$,

AETc. $\lim_{t\to\infty}\frac{f(t)}{t} = -c$.

Proof. It is easy to see that if we substitute (2.1.1) back into Definition 3.1.1 we obtain the result after some calculations. □

The next lemma helps to define the right hand side of the Newton system in case of the predictor step.

Lemma 3.1.1. *(Lemma 2.1 in [60]) Assume that* $\boldsymbol{x}$ *and* $\boldsymbol{s}$ *are the fixed current iterates. If Condition AET3 of Definition 3.1.1 and AETc of Proposition 3.1.1 is satisfied, then*

$$\lim_{\mu\to 0^+} \mathbf{a}_\varphi = -c\,\mathbf{xs}.$$

Proof. Let us consider (1.4.8) and if we use condition AETc from Proposition 3.1.1 and $t_i = \sqrt{\frac{x_i s_i}{\mu}}$, $i = 1,\dots n$, we obtain

$$\lim_{t_i\to\infty}\frac{\varphi(1)-\varphi(t_i^2)}{t_i^2\varphi'(t_i^2)} = \lim_{\mu\to 0^+}\frac{\varphi(1)-\varphi\left(\frac{x_i s_i}{\mu}\right)}{\frac{x_i s_i}{\mu}\varphi'\left(\frac{x_i s_i}{\mu}\right)} = \lim_{\mu\to 0^+}\frac{\mathbf{a}_{\varphi i}}{x_i s_i} = -c,$$

this proves the lemma. □

In order to reach a proper neighborhood of the central path, we first take a corrector step and then a predictor one. These algorithms are called corrector-predictor IPAs, see Potra [89]. Since the corrector step is a full-Newton step, we determine the corrector search directions from the system (1.4.2). This means that we have the scaled corrector system as (1.4.7), whose solution is

$$\mathbf{d}_x^c = (I+\bar{M})^{-1}\mathbf{p}_\varphi, \quad \mathbf{d}_s^c = \bar{M}(I+\bar{M})^{-1}\mathbf{p}_\varphi.$$

Using this solution and (1.4.5), we calculate the Newton directions $\Delta^c\mathbf{x}$ and $\Delta^c\mathbf{s}$. The new points after a corrector step are

$$\mathbf{x}^c = \mathbf{x}+\Delta^c\mathbf{x}, \quad \mathbf{s}^c = \mathbf{s}+\Delta^c\mathbf{s}. \tag{3.1.1}$$

In a predictor step we want to find the optimal solution in a greedy way. That is why we set $\mu = 0$ in $\mathbf{a}_\varphi$ of system (1.4.2). We determine the predictor search directions by using system (1.4.2). By setting $\mu = 0$ in $\mathbf{a}_\varphi$ and using Lemma 3.1.1, we get the following system:

$$\begin{aligned} -M\Delta^p\mathbf{x} + \Delta^p\mathbf{s} &= \mathbf{0}, \\ S\Delta^p\mathbf{x} + X\Delta^p\mathbf{s} &= -c\mathbf{x}^c\mathbf{s}^c. \end{aligned} \tag{3.1.2}$$

The scaled predictor system obtained from (3.1.2) is

$$\begin{aligned} -\bar{M}_+\mathbf{d}_x^p + \mathbf{d}_s^p &= \mathbf{0}, \\ \mathbf{d}_x^p + \mathbf{d}_s^p &= -c\mathbf{v}^+, \end{aligned} \tag{3.1.3}$$

where $\mathbf{v}^c = \sqrt{\frac{\mathbf{x}^c\mathbf{s}^c}{\mu}}$, $\mathbf{d}^+ = \sqrt{\frac{\mathbf{x}^c}{\mathbf{s}^c}}$, $D^+ = \operatorname{diag}(\mathbf{d}^+)$, $\bar{M}_+ = D^+MD^+$. System (3.1.3) has the solution:

$$\mathbf{d}_x^p = -c\mathbf{v}^c(I + \bar{M}_+)^{-1} \quad \text{and} \quad \mathbf{d}_s^p = -c\mathbf{v}^c\bar{M}_+(I + \bar{M}_+)^{-1}.$$

It is easy to get the predictor search directions $\Delta^p\mathbf{x}$ and $\Delta^p\mathbf{s}$ from this solution using (1.4.5). The predictor iterate is then calculated as

$$\mathbf{x}^p = \mathbf{x}^c + \theta\Delta^p\mathbf{x}, \quad \mathbf{s}^p = \mathbf{s}^c + \theta\Delta^p\mathbf{s}, \quad \mu^p = (1 - c\theta)\,\mu,$$

where $\theta \in \left(0, \frac{1}{c_4}\right)$, with $c_4 = \max\{1, c\}$. Parameter θ is called an update parameter.

In Table 3.1.1 we show some of the functions belonging to our class of AET functions that were defined in Definition 3.1.1. Furthermore, we give the corresponding vectors $\mathbf{a}_\varphi$, $\mathbf{p}_\varphi$ for these functions and the values of c, respectively in Table 3.1.1.

Table 3.1.1: Functions belonging to the new class AET functions

φ	$\mathbf{a}_\varphi$	c	$\mathbf{p}_\varphi$
$\varphi(t) = t$	$\mu\mathbf{e} - \mathbf{xs}$	1	$\mathbf{v}^{-1} - \mathbf{v}$
$\varphi(t) = \sqrt{t}$	$2(\sqrt{\mu\mathbf{xs}} - \mathbf{xs})$	2	$2(\mathbf{e} - \mathbf{v})$
$\varphi(t) = t - \sqrt{t}$	$\frac{\sqrt{\mu}\mathbf{xs}}{2\sqrt{\mathbf{xs}} - \sqrt{\mu}\mathbf{e}} - \mathbf{xs}$	1	$\frac{2(\mathbf{v} - \mathbf{v}^2)}{2\mathbf{v} - \mathbf{e}}$
$\varphi(t) = t^2 - t + \sqrt{t}$	$\frac{4\mu^2\sqrt{\mathbf{xs}} + 2\mu\mathbf{xs}\sqrt{\mathbf{xs}} - 3\mu\mathbf{xs}\sqrt{\mu}}{2(4\mathbf{xs}\sqrt{\mathbf{xs}} - 2\mu\sqrt{\mathbf{xs}} + \mu\sqrt{\mu})} - \frac{\mathbf{xs}}{2}$	$\frac{1}{2}$	$\frac{2(\mathbf{e} - \mathbf{v})(\mathbf{e} + \mathbf{v}^2 + \mathbf{v}^3)}{4\mathbf{v}^3 - 2\mathbf{v} + \mathbf{e}}$

In the following subsection the new PC IPA for solving $P_*(\kappa)$-LCPs is presented.

3.1.2 New predictor-corrector interior-point algorithm for solving $P_*(\kappa)$-linear complementarity problems

Consider the proximity measure defined in (2.2.1) and the τ-neighborhood of the central path given in (2.2.2).

In Algorithm 2 we introduce the PC IPA for solving $P_*(\kappa)$-LCPs.

Algorithm 2 : PC IPA for $P_*(\kappa)$-LCPs based on a new class of AET functions [60]

Input:
Function φ satisfying the properties from Definition 3.1.1;
Accuracy parameter $\epsilon > 0$;
Update parameter $\theta \in \left(0, \frac{1}{c_4}\right)$, with $c_4 = \max\{1, c\}$;
Proximity parameter $\tau > 0$;
A $P_*(\kappa)$ matrix M with a known upper bound κ of the handicap $\hat{\kappa}(M)$;
Starting point $(\mathbf{x}^0, \mathbf{s}^0) \in \mathcal{F}^+$, satisfying $\delta(\mathbf{x}^0, \mathbf{s}^0; \mu^0) \leq \tau$ and $\mathbf{v}^0 > \xi \mathbf{e}$.
Iteration:
begin
$k := 0$;
while $\left(\mathbf{x}^k\right)^T \mathbf{s}^k > \epsilon$ **do begin**
(corrector step)
compute $(\Delta^c x^k, \Delta^c s^k)$ from system (1.4.7) by using (1.4.5) with φ;
update $(\mathbf{x}^c)^k := \mathbf{x}^k + \Delta^c x^k$ and $(\mathbf{s}^c)^k := \mathbf{s}^k + \Delta^c s^k$;
(predictor step)
compute $(\Delta^p x^k, \Delta^p s^k)$ from system (3.1.3) by using (1.4.5);
update $(\mathbf{x}^p)^k := (\mathbf{x}^c)^k + \theta \Delta^p \mathbf{x}^k$ and $(\mathbf{s}^p)^k := (\mathbf{s}^c)^k + \theta \Delta^p \mathbf{s}^k$;
(update of the parameters and the iterates)
$(\mu^p)^k := (1 - c\theta)\, \mu^k$;
$\mathbf{x}^{k+1} := (\mathbf{x}^p)^k, \quad \mathbf{s}^{k+1} := (\mathbf{s}^p)^k, \quad \mu^{k+1} := (\mu^p)^k, \quad k := k+1;$
end
end.

In the next subsection we present the complexity analysis of the PC IPA.

3.1.3 Complexity analysis of the predictor-corrector interior-point algorithm

First, we consider the analysis of the corrector step and after that we provide the analysis of the predictor step.

3.1.3.1 Corrector step

It is important to note again that the corrector step is a full-Newton step. This means that the analysis of this part is similar to the one of the short-step IPA presented in Chapter 2. Considering Lemma 2.3.1 we can prove the strict feasibility of the corrector step.

Lemma 3.1.2. *(Lemma 3.1. in [58]) Let* $(\boldsymbol{x},\boldsymbol{s}) \in \mathcal{F}^+$ *be given, such that* $\delta(\boldsymbol{x},\boldsymbol{s};\mu) < \frac{1}{\sqrt{1+4\kappa}}$ *and* $\mathbf{v} > \xi\mathbf{e}$*. For any function satisfying AET2 of Definition 3.1.1, we have that* $(\mathbf{x}^c,\, \mathbf{s}^c) \in \mathcal{F}^+$.

Based on Lemma 2.3.2 we can show the quadratic convergence of the corrector step.

Lemma 3.1.3. *(Lemma 3.2. in [58]) Let* $(\boldsymbol{x},\boldsymbol{s}) \in \mathcal{F}^+$, $\mathbf{v} > \xi\mathbf{e}$ *and* $\mathbf{v}^c = \sqrt{\frac{\boldsymbol{x}^c\boldsymbol{s}^c}{\mu}}$ *be given and suppose that* $\delta(\mathbf{x},\mathbf{s};\mu) < \sqrt{\frac{1-\xi^2}{1+4\kappa}}$*. For any function* φ *satisfying AET1 and AET2 of Definition 3.1.1, it is shown after a full-Newton step we have* $\mathbf{v}^c > \xi\mathbf{e}$ *and*

$$\delta(\boldsymbol{x}^c,\boldsymbol{s}^c;\mu) \leq c_1(c_3+2+4\kappa)\delta(\boldsymbol{x},\boldsymbol{s};\mu)^2,$$

where $c_1, c_3 = \max\{1, c_2\} \in \mathbb{R}_+$.

We give an upper bound on the duality gap.

Lemma 3.1.4. *(Lemma 3.3. in [58]) Let* $\delta := \delta(\boldsymbol{x},\boldsymbol{s};\mu)$ *and suppose that the vectors* $\boldsymbol{x}^c$ *and* $\boldsymbol{s}^c$ *are obtained using a full-Newton step, thus* $\boldsymbol{x}^c = \boldsymbol{x}+\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^c = \boldsymbol{s}+\Delta\boldsymbol{s}$*. For any function* φ *satisfying AET2 of Definition 3.1.1 with* $c_2 \in \mathbb{R}_+$*, we have that*

$$(\boldsymbol{x}^c)^T \boldsymbol{s}^c \leq \mu\left(n+(c_2+1)\delta^2\right).$$

In the next subsubsection we present some technical lemmas that will be important in the analysis.

3.1.3.2 Technical lemmas

Let us consider the first lemma.

Lemma 3.1.5. *(Lemma 3.4. in [60]) Let* f *be a function satisfying AETa from Proposition 3.1.1. Then,*

$$c_0\|\mathbf{e}-\mathbf{v}\| \leq \|f(\mathbf{v})\| \leq 2c_1\|\mathbf{e}-\mathbf{v}^2\|.$$

Proof. We have

$$\|f(\mathbf{v})\| \;=\; \sqrt{\sum_{i=1}^{n}(f(v_i))^2} \geq c_0\sqrt{\sum_{i=1}^{n}(1-v_i)^2} = c_0\|\mathbf{e}-\mathbf{v}\|,$$

on the other hand, we also have

$$\|f(\mathbf{v})\| \;=\; \sqrt{\sum_{i=1}^{n}(f(v_i))^2} \leq 2c_1\sqrt{\sum_{i=1}^{n}\left(1-v_i^2\right)^2} = 2c_1\|\mathbf{e}-\mathbf{v}^2\|,$$

which proves the lemma. □

We generalize Lemma 5.3. in [25] in the following technical lemma.

Lemma 3.1.6. *(Lemma 3.5. in [60]) Consider a function φ from the new class of AET functions given in Definition 3.1.1. Then,*

$$\|\mathbf{d}_x^p\mathbf{d}_s^p\| \leq \frac{c^2\, n(1+2\kappa)\left(1+\frac{2}{c_0}\,\delta^+\right)^2}{2},$$

where $\delta^+ = \delta(\mathbf{x}^c,\mathbf{s}^c;\mu)$.

Proof. According to [25] (page 2641), we have

$$(1+4\kappa)\sum_{i\in I_+} d^p_{x_i}d^p_{s_i} + \sum_{i\in I_-} d^p_{x_i}d^p_{s_i} \;\;\geq\;\; 0, \tag{3.1.4}$$

From the second equation of system (3.1.3) we obtain

$$\sum_{i\in I_+} d^p_{x_i}d^p_{s_i} \leq \frac{1}{4}\|\mathbf{d}_x^p+\mathbf{d}_s^p\|^2 = \frac{1}{4}\,\|c\mathbf{v}^c\|^2. \tag{3.1.5}$$

Using (3.1.4) and (3.1.5), we get

$$\begin{aligned} c^2\|\mathbf{v}^c\|^2 = \|\mathbf{d}_x^p+\mathbf{d}_s^p\|^2 \;&=\; \|\mathbf{d}_x^p\|^2+\|\mathbf{d}_s^p\|^2+2\left(\sum_{i\in I_+} d^p_{x_i}d^p_{s_i}+\sum_{i\in I_-} d^p_{x_i}d^p_{s_i}\right)\\ &\geq\; \|\mathbf{d}_x^p\|^2+\|\mathbf{d}_s^p\|^2-8\kappa\sum_{i\in I_+} d^p_{x_i}d^p_{s_i}\\ &\geq\; \|\mathbf{d}_x^p\|^2+\|\mathbf{d}_s^p\|^2-2c^2\kappa\,\|\mathbf{v}^c\|^2. \end{aligned}$$

Hence, we showed that $\|\mathbf{d}_x^p\|^2 + \|\mathbf{d}_s^p\|^2 \leq c^2\,(1+2\kappa)\,\|\mathbf{v}^c\|^2$. Considering the calculations given in [25] (page 2642), we have

$$\|\mathbf{v}^c\| \quad \leq \quad \sqrt{n}(\sigma^+ + 1), \tag{3.1.6}$$

where $\sigma^+ = \|\mathbf{e} - \mathbf{v}^c\|$. Using Lemma 3.1.5 we obtain a lower bound on δ^+ as the following

$$\delta^+ \quad = \quad \frac{1}{2}\,\|\mathbf{p}_{\mathbf{v}^c}\| = \frac{1}{2}\|f(\mathbf{v}^c)\| \geq \frac{c_0}{2}\,\left\|\mathbf{e} - \mathbf{v}^+\right\| = \frac{c_0}{2}\sigma^+, \tag{3.1.7}$$

where f is given in (2.1.1). Based on (3.1.6) and (3.1.7) we see that

$$\|\mathbf{v}^c\| \quad \leq \quad \sqrt{n}\left(1 + \frac{2}{c_0}\,\delta^+\right). \tag{3.1.8}$$

Finally, it can be seen that

$$\|\mathbf{d}_x^p \mathbf{d}_s^p\| \leq \frac{1}{2}\left(\|\mathbf{d}_x^p\|^2 + \|\mathbf{d}_s^p\|^2\right) \leq \frac{c^2}{2}\,(1+2\kappa)\,\|\mathbf{v}^c\|^2 \leq \frac{c^2\,n(1+2\kappa)\left(1+\frac{2}{c_0}\,\delta^+\right)^2}{2}.$$

This proves the lemma. □

Next we analyse the predictor step.

3.1.3.3 Predictor step

First, let us consider the strict feasibility of the predictor step.

Lemma 3.1.7. *(Lemma 3.6. in [60]) Let* $(\mathbf{x}^c, \mathbf{s}^c) > \mathbf{0}$ *and* $\mu > 0$ *such that* $\delta^+ := \delta(\mathbf{x}^c, \mathbf{s}^c; \mu) < \frac{c_0}{2}$. *Furthermore, let* $\theta \in \left(0, \frac{1}{c_4}\right)$*, where* $c_4 = \max\{1, c\}$. *Let* $\mathbf{x}^p = \mathbf{x}^c + \theta \Delta^p \mathbf{x}$, $\quad \mathbf{s}^p = \mathbf{s}^c + \theta \Delta^p \mathbf{s}$ *be the iterates after a predictor step. Then, for all functions* φ *satisfying the conditions given in Definition 3.1.1, we have* $\mathbf{x}^p, \mathbf{s}^p > \mathbf{0}$*, if* $u(\delta^+, \theta, n) > 0$*, where*

$$u(\delta^+, \theta, n) := \left(1 - \frac{2}{c_0}\,\delta^+\right)^2 - \frac{\theta^2 c^2\,n(1+2\kappa)\left(1+\frac{2}{c_0}\,\delta^+\right)^2}{2(1-c\theta)}. \tag{3.1.9}$$

Proof. From (1.4.5), we get

$$\mathbf{x}^p(\alpha) = \frac{\mathbf{x}^+}{\mathbf{v}^c}(\mathbf{v}^c + \alpha\,\theta\,\mathbf{d}_x^p), \quad \mathbf{x}^p(\alpha) = \frac{\mathbf{s}^+}{\mathbf{v}^c}(\mathbf{v}^c + \alpha\,\theta\,\mathbf{d}_s^p).$$

Based on (5.17), page 2643 in [25] and considering the second equation of the system (3.1.3), we obtain

$$\begin{aligned} \mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha) &= \mu\Big((\mathbf{v}^c)^2 + \alpha\theta\mathbf{v}^c(-c\,\mathbf{v}^c) + \alpha^2\,\theta^2\,\mathbf{d}_x^p\mathbf{d}_s^p\Big) \\ &= \mu\Big((1-c\alpha\theta)\,(\mathbf{v}^c)^2 + \alpha^2\,\theta^2\,\mathbf{d}_x^p\mathbf{d}_s^p\Big). \end{aligned} \tag{3.1.10}$$

Considering (3.1.10) and the ideas from [25], we have

$$\begin{aligned} \min\left(\frac{\mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha)}{\mu\,(1-c\alpha\theta)}\right) &= \min\left((\mathbf{v}^c)^2 + \frac{\alpha^2\theta^2}{1-c\alpha\theta}\mathbf{d}_x^p\mathbf{d}_s^p\right) \\ &\geq \min\left((\mathbf{v}^c)^2 - \frac{\alpha^2\theta^2}{1-c\alpha\theta}\|\mathbf{d}_x^p\mathbf{d}_s^p\|_\infty\mathbf{e}\right) \\ &\geq \min\left((\mathbf{v}^c)^2 - \frac{\theta^2}{1-c\theta}\|\mathbf{d}_x^p\mathbf{d}_s^p\|_\infty\mathbf{e}\right). \end{aligned} \tag{3.1.11}$$

It can be seen that

$$h(\alpha) = \frac{\alpha^2\theta^2}{1-c\alpha\theta}$$

is a strictly monotone increasing function for $0 \leq \alpha \leq 1$ and for each fixed update parameter $\theta \in \left(0, \frac{1}{c_4}\right)$, where $c_4 = \max\{1, c\}$. This shows that the last inequality holds. Using

$$|1 - v_i^+| \leq \|\mathbf{e} - \mathbf{v}^c\|, \;\; \forall i = 1, \ldots, n$$

we obtain

$$1 - \sigma^+ \leq v_i^+ \leq 1 + \sigma^+, \;\; \forall i = 1, \ldots, n.$$

Considering (3.1.7) it can be seen that

$$\min\,(\mathbf{v}^c)^2 \geq (1-\sigma^+)^2 \geq \left(1 - \frac{2}{c_0}\,\delta^+\right)^2. \tag{3.1.12}$$

According to Lemma 3.1.6, (3.1.12), and by applying $\|\cdot\|_\infty \leq \|\cdot\|$, we get

$$\begin{aligned} \min\left(\frac{\mathbf{x}^p(\alpha)\,\mathbf{s}^p(\alpha)}{\mu\,(1-c\,\alpha\,\theta)}\right) &\geq \left(1 - \frac{2}{c_0}\delta^+\right)^2 - \frac{\theta^2 c^2\, n(1+2\kappa)\left(1+\frac{2}{c_0}\,\delta^+\right)^2}{2(1-c\theta)} \\ &=: u(\delta^+, \theta, n). \end{aligned} \tag{3.1.13}$$

If $u(\delta^+, \theta, n) > 0$, we have $\mathbf{x}^p(\alpha)\,\mathbf{s}^p(\alpha) > 0$ for $0 \leq \alpha \leq 1$. This means that $\mathbf{x}^p(\alpha)$ and $\mathbf{s}^p(\alpha)$ do not change sign on $0 \leq \alpha \leq 1$. Therefore, we know that $\mathbf{x}^p(0) = \mathbf{x}^c > \mathbf{0}$ and $\mathbf{s}^p(0) = \mathbf{s}^c > \mathbf{0}$, so $\mathbf{x}^p(1) = \mathbf{x}^p > \mathbf{0}$ and $\mathbf{s}^p(1) = \mathbf{s}^p > \mathbf{0}$ as well. This proves the lemma.

$\square$

Let

$$\mathbf{v}^p = \sqrt{\frac{\mathbf{x}^p \mathbf{s}^p}{\mu^p}}, \tag{3.1.14}$$

where $\mu^p = (1 - c\theta)\,\mu$. Considering $\alpha = 1$ and (3.1.13), we obtain

$$\min\,(\mathbf{v}^p)^2 \geq u(\delta^+, \theta, n) > 0. \tag{3.1.15}$$

Now, we show the quadratic convergence of the predictor step.

Lemma 3.1.8. *(Lemma 3.7. in [60]) Consider a function φ belonging to the new class of AET functions given in Definition 3.1.1. Furthermore, let $\delta^+ := \delta(\mathbf{x}^c, \mathbf{s}^c; \mu) < \frac{c_0}{2}$, $u(\delta^+, \theta, n) > \xi^2$, $\mathbf{v} > \xi \mathbf{e}$, $\mu^p = (1 - c\theta)\,\mu$, where $\theta \in \left(0, \frac{1}{c_4}\right)$, with $c_4 = \max\{1, c\}$, and let $\mathbf{x}^p$ and $\mathbf{s}^p$ denote the iterates after a predictor step. Then, $\mathbf{v}^p > \xi \mathbf{e}$ and*

$$\delta^p := \delta(\mathbf{x}^p, \mathbf{s}^p; \mu^p) \leq c_1((2 + c_3) + 4\kappa)\delta^2 + \frac{c_1\, c^2\, n\, \theta^2}{2(1 - c\theta)}(1 + 2\kappa)\left(1 + \frac{2}{c_0}\delta^+\right)^2,$$

where $\delta := \delta(\mathbf{x}, \mathbf{s}; \mu)$ and $c_3 = \max\{1, c_2\}$.

Proof. We have $u(\delta^+, \theta, n) > \xi^2$. Therefore, using Lemma 3.1.7 we obtain $\mathbf{x}^p, \mathbf{s}^p > \mathbf{0}$. From (3.1.15), we get

$$\min\,(\mathbf{v}^p) \geq \sqrt{u(\delta^+, \theta, n)} > \xi.$$

With this we proved the first part of the lemma. From (1.4.8) we get

$$\delta^p \quad := \quad \frac{\|f(\mathbf{v}^p)\|}{2} \leq c_1 \left\|\mathbf{e} - (\mathbf{v}^p)^2\right\|, \tag{3.1.16}$$

where we used (2.1.1) and the function φ that satisfies condition AETa from Proposition 3.1.1. It is important to mention that considering Lemma 3.1.5, it can be seen that the last inequality of (3.1.16) holds.

By using (3.1.10) we get

$$(\mathbf{v}^p)^2 = (\mathbf{v}^c)^2 + \frac{\theta^2}{1 - c\theta}\mathbf{d}_x^p \mathbf{d}_s^p, \tag{3.1.17}$$

therefore

$$\begin{aligned}\delta^p &\leq c_1 \left\| \mathbf{e} - (\mathbf{v}^c)^2 - \frac{\theta^2}{1-c\theta} \mathbf{d}_x^p \mathbf{d}_s^p \right\| \\ &\leq c_1 \left(\left\| \mathbf{e} - (\mathbf{v}^c)^2 \right\| + \frac{\theta^2}{1-c\theta} \left\| \mathbf{d}_x^p \mathbf{d}_s^p \right\| \right). \end{aligned} \tag{3.1.18}$$

From (3.1.18) and using (5.25) from [25] we have

$$\left\| \mathbf{e} - (\mathbf{v}^c)^2 \right\| \leq \| \mathbf{e} - \mathbf{v}^2 - \mathbf{v}\mathbf{p}_\varphi \| + \left\| \frac{\mathbf{p}_\varphi^2}{4} \right\| + \left\| \frac{\mathbf{q}_\varphi^2}{4} \right\|, \tag{3.1.19}$$

where $\mathbf{q}_\varphi = \mathbf{d}_x^c - \mathbf{d}_s^c$ and $\mathbf{p}_\varphi = f(\mathbf{v})$. After further computations we get

$$\left\| \mathbf{e} - \mathbf{v}^2 - \mathbf{v}\mathbf{p}_\varphi \right\| = \left\| \mathbf{e} - \mathbf{v}^2 - \mathbf{v} f(\mathbf{v}) \right\| \leq c_3 \left\| \frac{\mathbf{p}_\varphi^2}{4} \right\|, \tag{3.1.20}$$

where $c_3 = \max\{1, c_2\}$. Considering AETb of Proposition 3.1.1 and a modification of Lemma 3.1.5, it can be seen that the last inequality holds. According to Lemma 5.4 from [25], (3.1.19) and (3.1.20) we derive

$$\begin{aligned}\left\| \mathbf{e} - (\mathbf{v}^c)^2 \right\| &\leq (c_3+1) \left\| \frac{\mathbf{p}_\varphi^2}{4} \right\| + \left\| \frac{\mathbf{q}_\varphi^2}{4} \right\| \leq (1+c_3) \left(\frac{\|\mathbf{p}_\varphi\|}{2} \right)^2 + \left(\frac{\|\mathbf{q}_\varphi\|}{2} \right)^2 \\ &\leq ((2+c_3)+4\kappa)\delta^2. \end{aligned} \tag{3.1.21}$$

Finally, applying Lemma 3.1.6, (3.1.18) and (3.1.21) we obtain

$$\delta^p \leq c_1((2+c_3)+4\kappa)\delta^2 + \frac{c_1 c^2 n \theta^2}{2(1-c\theta)} (1+2\kappa) \left(1 + \frac{2}{c_0} \delta^+ \right)^2.$$

This proves the lemma. □

At the end of this subsubsection we give an upper bound on the duality gap after a main iteration.

Lemma 3.1.9. *(Lemma 3.8. in [60]) Let φ be member of the new class of functions proposed in Definition 3.1.1 and let $\theta \in \left(0, \frac{1}{c_4}\right)$, where $c_4 = \max\{1, c\}$. If $\mathbf{x}^p$ and $\mathbf{s}^p$ are the iterates obtained* *after the predictor step of Algorithm 2, then*

$$\left(\mathbf{x}^p\right)^T\mathbf{s}^p \le \left(1-c\theta+\frac{c^2\theta^2}{2}\right)\left(\mathbf{x}^c\right)^T\mathbf{s}^c < \frac{\left(n+(c_2+1)\delta^2\right)\mu^p}{1-c\theta}.$$

Proof. Consider $\alpha = 1$ and (3.1.10). Then, from (3.1.14), we have

$$\begin{aligned}\left(\mathbf{x}^p\right)^T\mathbf{s}^p &= \mu^p\mathbf{e}^T\left(\mathbf{v}^p\right)^2 = \mu\mathbf{e}^T\left((1-c\theta)\left(\mathbf{v}^c\right)^2+\theta^2\mathbf{d}_x^p\mathbf{d}_s^p\right)\\ &= (1-c\theta)\left(\mathbf{x}^c\right)^T\mathbf{s}^c+\mu\theta^2\left(\mathbf{d}_x^p\right)^T\mathbf{d}_s^p.\end{aligned} \tag{3.1.22}$$

Furthermore, by using the proof of Lemma 5.7 from [25], we obtain

$$\left(\mathbf{x}^p\right)^T\mathbf{s}^p \le \left(1-c\theta+\frac{c^2\theta^2}{2}\right)\left(\mathbf{x}^c\right)^T\mathbf{s}^c. \tag{3.1.23}$$

If $\theta \in \left(0, \frac{1}{c_4}\right)$, where $c_4 = \max\{1, c\}$, then we have

$$1-c\theta+\frac{c^2\theta^2}{2} < 1. \tag{3.1.24}$$

Finally, based on (3.1.23), and (3.1.24), $\mu^p = (1-c\theta)\mu$, and by Lemma 3.1.4 we obtain

$$\begin{aligned}\left(\mathbf{x}^p\right)^T\mathbf{s}^p &\le \left(1-c\theta+\frac{c^2\theta^2}{2}\right)\left(\mathbf{x}^c\right)^T\mathbf{s}^c\\ &< \left(\mathbf{x}^c\right)^T\mathbf{s}^c \le \mu\left(n+(c_2+1)\delta^2\right) = \frac{\left(n+(c_2+1)\delta^2\right)\mu^p}{1-c\theta}.\end{aligned}$$

This proves the lemma. □

3.1.4 Iteration bound

In this subsection we prove that the algorithm is well defined. First, we have to show that $u(\delta^+, \theta, n) > \xi^2$, where $u(\delta^+, \theta, n)$ is defined in (3.1.9). For this, let us use the following notation:

$$Q(\tau) = \frac{\left(1-\frac{1}{c_0}\tau\right)^2-\xi^2}{\left(1+\frac{1}{c_0}\tau\right)^2}. \tag{3.1.25}$$

It can be seen if $\xi = 0$, then $Q(\tau) > 0$. On the other hand, if $\xi > 0$, it is important to show that $\tau < (1-\xi)c_0$ to have $Q(\tau) > 0$.

Let us denote by

$$Q_L = \frac{\theta^2 c^2\, n(1+2\kappa)}{2(1-c\theta)}. \tag{3.1.26}$$

Now, we prove that our PC IPA is well defined with the appropriate parameters τ and θ.

Lemma 3.1.10. *(Lemma 4.1. in [60]) Let φ be a function from the new class of AET functions given in Definition 3.1.1. Let $0 \le \xi \le \frac{1}{2}$, $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, where $r \ge 2$, and $0 < \theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \ge 1$, $c > 0$, $c_5 = \max\left\{\frac{1}{4}, c_1\right\}$ and $c_3 = \max\{1, c_2\}$. If (i) $r \le 2q$ and (ii) $Q_L < Q(\tau)$, then the PC IPA proposed in Algorithm 2 is well defined.*

Proof. We prove that if $0 < \theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, then $\theta \in \left(0, \frac{1}{c_4}\right)$ is satisfied, where $c_4 = \max\{1, c\}$. From $c_5 \ge \frac{1}{4}$, $c_3 \ge 1$, $0 < \theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, $n \ge 1$, $q \ge 1$, and $\kappa \ge 0$, we see that

$$\theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}} < \frac{1}{c_4}. \tag{3.1.27}$$

Moreover, we have to prove that $u(\delta^+, \theta, n) > 0$. For this, we consider Lemma 3.1.7. First, let us consider an upper bound for Q_L. By using the upper bound $\theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, $c_1, c_3 \ge 1$, $c_5 \ge c_1$ and $\kappa \ge 0$ we obtain

$$\frac{1}{2(1-c\theta)} \le \frac{6q}{12q-1}. \tag{3.1.28}$$

Moreover, it can be seen that

$$\theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}} \le \frac{1}{16qcc_5(1+2\kappa)\sqrt{n}}. \tag{3.1.29}$$

According to (3.1.28) and (3.1.29), we have

$$\begin{aligned} Q_L &= \frac{\theta^2 c^2\, n(1+2\kappa)}{2(1-c\theta)} \\ &\le c^2 n(1+2\kappa)\, \frac{6q}{12q-1}\, \frac{1}{16qcc_5(1+2\kappa)\sqrt{n}}\, \frac{\sqrt{1-\xi^2}}{16qc_5c(c_3+2+4\kappa)\sqrt{n}} \\ &< \frac{3}{4(12q-1)}\frac{r}{q}\tau. \end{aligned} \tag{3.1.30}$$

Finally, it can be seen that from (3.1.30), after some calculations we get $c_5 Q_L < \frac{3}{4(12q-1)}\frac{r}{q}\tau$.

Now, consider $(\mathbf{x},\mathbf{s}) \in \mathcal{F}^+$ such that $\mathbf{v} > \xi\mathbf{e}$ and $\delta(\mathbf{x},\mathbf{s};\mu) \le \tau$. In this case Lemma 3.1.3 can be considered since from $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, where $r \ge 2$ and using $c_5 \ge \frac{1}{4}$, $c_3 \ge 1$ we have $\tau < \sqrt{\frac{1-\xi^2}{1+4\kappa}}$. Using Lemma 3.1.3 we obtain

$$\delta^+ := \delta(\mathbf{x}^c,\mathbf{s}^c;\mu) \le c_1(c_3+2+4\kappa)\delta(\mathbf{x},\mathbf{s};\mu)^2,$$

where $c_3 = \max\{1,c_2\} \in \mathbb{R}_+$. Based on $\delta(\mathbf{x},\mathbf{s};\mu) \le \tau$, $r \ge 2$ and $c_1 \le c_5$, we have

$$\delta^+ \le c_5(c_3+2+4\kappa)\tau^2 = \frac{\sqrt{1-\xi^2}}{8r}\tau \le \frac{1}{16}\tau. \tag{3.1.31}$$

By using $c_5 \ge \frac{1}{4}$, $c_3 \ge 1$ and $\kappa \ge 0$ we get

$$\delta^+ \le c_5(c_3+2+4\kappa)\tau^2 \le \frac{1-\xi^2}{64r^2c_5(c_3+2+4\kappa)} \le \frac{1}{48r^2}. \tag{3.1.32}$$

Furthermore, according to $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $c_1,c_3 \ge 1$, $c_5 \ge \frac{1}{4}$, $c_0 \ge \frac{1}{4}$, $r \ge 2$ and (3.1.31), we derive

$$\delta^+ \le \frac{1}{16}\tau < \frac{1}{2}\tau \le \frac{1}{4r(3+4\kappa)} < \frac{1}{24} < \frac{c_0}{2}. \tag{3.1.33}$$

This shows that $\tau < \frac{c_0}{2}$. Considering $\xi \le \frac{1}{2}$, we get

$$\tau < \frac{c_0}{2} \le (1-\xi)c_0.$$

Therefore, $Q(\tau) > 0$. Now, we will prove $u(\delta^+,\theta,n) > 0$. First, let us reconsider the function

$$u(\delta^+,\theta,n) = \left(1-\frac{2}{c_0}\delta^+\right)^2 - \frac{\theta^2c^2\,n(1+2\kappa)\left(1+\frac{2}{c_0}\delta^+\right)^2}{2(1-c\theta)}.$$

Following from (3.1.30) we get $Q_L < 1$. According to $Q_L < 1$ and (3.1.33) we obtain

$$\delta^+ \le \frac{1}{16}\tau < \frac{c_0}{2} < \frac{c_0}{2}\frac{1+Q_L}{1-Q_L}. \tag{3.1.34}$$

It is easy to see that this function is monotone decreasing with respect to δ^+ in the interval

$\left(0, \frac{c_0}{2}\frac{1+Q_L}{1-Q_L}\right)$. Considering condition (ii) given in the lemma and (3.1.33) we have

$$u(\delta^+,\theta,n) > u\left(\frac{1}{2}\tau,\theta,n\right) = \left(1-\frac{1}{c_0}\tau\right)^2 - Q_L\left(1+\frac{1}{c_0}\tau\right)^2 > \xi^2 > 0. \tag{3.1.35}$$

Therefore, we satisfied the condition $u(\delta^c,\theta,n) > 0$ from Lemma 3.1.7 and $u(\delta^c,\theta,n) > \xi^2$ from Lemma 3.1.8.

Furthermore, using Lemma 3.1.8 and $c_1 \le c_5$, we have

$$\delta^p \le c_5((2+c_3)+4\kappa)\delta^2 + \frac{c_5\, c^2\, n\,\theta^2}{2(1-c\theta)}(1+2\kappa)\left(1+\frac{2}{c_0}\delta^+\right)^2, \tag{3.1.36}$$

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

Based on (3.1.30), (3.1.32) and $c_0 \ge \frac{1}{4}$ and $c_5 Q_L < \frac{3}{4(12q-1)}\frac{r}{q}\tau$ we obtain

$$c_5 Q_L\left(1+\frac{2}{c_0}\delta^+\right)^2 \quad < \quad \frac{3}{4(12q-1)}\frac{r}{q}\left(1+\frac{1}{6r^2}\right)^2\tau. \tag{3.1.37}$$

By using $r \ge 2$, $q \ge 1$ and $r \le 2q$ we get

$$\frac{3}{4(12q-1)}\frac{r}{q}\left(1+\frac{1}{6r^2}\right)^2 \le \frac{3}{44}\cdot 2\cdot\frac{25^2}{24^2} < \frac{15}{16}. \tag{3.1.38}$$

Finally, according to (3.1.31), (3.1.36), (3.1.37) and (3.1.38) we obtain

$$\delta^p < \left(\frac{1}{16}+\frac{15}{16}\right)\tau < \tau. \tag{3.1.39}$$

This means, that for a feasible solution for which $\delta(\mathbf{x},\mathbf{s};\mu) \le \tau$, after a corrector and a predictor step we obtain the predictor iteration $(\mathbf{x}^p,\mathbf{s}^p)$ such as $\delta^p(\mathbf{x}^p,\mathbf{s}^p,\mu^p) < \tau$. □

In the next lemma we consider a sufficient condition for satisfying condition (ii) from Lemma 3.1.10.

Lemma 3.1.11. *(Lemma 4.2. in [60]) Let φ be a function belonging to the AET class defined in Definition 3.1.1. Consider $0 \le \xi \le \frac{1}{2}$, $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $r \ge 2$, and $0 < \theta \le \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \ge \frac{1}{2}r$, $c_5 = \max\{\frac{1}{4}, c_1\}$, $c_3 = \max\{1, c_2\}$, and $c_4 = \max\{1, c\}$ with $c > 0$. Consider $Q(\tau)$ given in (3.1.25). If*

$$\frac{1}{16q^2} < Q\left(\frac{1}{6r}\right), \tag{3.1.40}$$

then condition (ii) of Lemma 3.1.10 is satisfied.

Proof. Let us consider (3.1.29), we get

$$\frac{1}{2(1-c\theta)} < 1. \tag{3.1.41}$$

By using the properties of the function φ, $c_5 \geq \frac{1}{4}$, and $\kappa \geq 0$, we get

$$Q_L = \frac{\theta^2 c^2\, n(1+2\kappa)}{2(1-c\theta)} \quad < \quad c^2 n\,(1+2\kappa)\, \frac{1}{256q^2c^2c_5^2(1+2\kappa)^2 n} \leq \frac{1}{16q^2}. \tag{3.1.42}$$

Moreover, using $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $c_3 \geq 1$, $c_5 \geq \frac{1}{4}$, and $\kappa \geq 0$ we have

$$\tau \leq \frac{1}{6r}. \tag{3.1.43}$$

It is important to note that the function $Q(\tau)$ is strictly decreasing with respect to τ on $\left(0, \frac{c_0(2-\xi^2)}{2}\right)$. On the other hand, if $r \geq 2$ and $c_0 \geq \frac{1}{4}$, then we have $\frac{1}{6r} < \frac{c_0(2-\xi^2)}{2}$. This means that $Q(\tau)$ is also strictly decreasing with respect to τ on $\left(0, \frac{1}{6r}\right]$. Therefore, according to (3.1.43), we derive

$$Q(\tau) \geq Q\left(\frac{1}{6r}\right). \tag{3.1.44}$$

Finally, based on (3.1.40), (3.1.41), (3.1.42) and (3.1.44) we see that

$$Q_L = \frac{\theta^2 c^2\, n(1+2\kappa)}{2(1-c\theta)} < \frac{1}{16q^2} < Q\left(\frac{1}{6r}\right) \leq Q(\tau).$$

This proves the lemma. □

With the next lemma we show that if condition (i) from Lemma 3.1.10 holds it implies that condition (ii) from Lemma 3.1.10 is also fulfilled.

Lemma 3.1.12. *(Lemma 4.3. in [60]) Let φ be a function belonging to the AET class defined in Definition 3.1.1. Consider $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $0 \leq \xi \leq \frac{1}{2}$, $r \geq 2$, and $0 < \theta \leq \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \geq 1$, $c_5 = \max\{\frac{1}{4}, c_1\}$, and $c_3 = \max\{1, c_2\}$. Consider Q given in (3.1.25). If condition (i) from Lemma 3.1.10 is satisfied, then condition (ii) from Lemma 3.1.10 also holds.*

Proof. Considering the following

$$\tilde{g}(r) = \frac{5}{4r}\frac{\left(1+\frac{1}{6rc_0}\right)}{\sqrt{\left(1-\frac{1}{6rc_0}\right)^2 - \xi^2}},$$

it can be seen that this function is decreasing with respect to r. Hence, considering that $c_0 \geq \frac{1}{4}$ and $\xi \leq \frac{1}{2}$, for $r \geq 2$ we get

$$\tilde{g}(r) \leq \tilde{g}(2) < 2. \tag{3.1.45}$$

Using (3.1.45) and condition (i) of Lemma 3.1.10 we obtain

$$\frac{1}{\sqrt{Q\left(\frac{1}{6r}\right)}} < \frac{8}{5}r < 4q.$$

Therefore, when condition (i) of Lemma 3.1.10 is satisfied, that means that (3.1.40) holds. Based on Lemma 3.1.11, we derived that condition (ii) from Lemma 3.1.10 is also satisfied. □

From Lemma 3.1.12 we can see that Algorithm 2 is well defined as well.

Corollary 3.1.1. *(Corollary 4.4. in [60]) Let φ be a function belonging to the AET class defined in Definition 3.1.1. Consider $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $r \geq 2$, and $0 < \theta \leq \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \geq 1$, $c_5 = \max\{\frac{1}{4}, c_1\}$, and $c_3 = \max\{1, c_2\}$. If $q \geq \frac{1}{2}r$, then the PC IPA proposed in Algorithm 2 is well defined.*

Finally, in the last lemma of this section we show an upper bound on the iteration number.

Lemma 3.1.13. *(Lemma 4.5. in [60]) Let φ be a function belonging to the AET class defined in Definition 3.1.1. Consider $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $r \geq 2$, and $0 < \theta \leq \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \geq \frac{1}{2}r$, $c_5 = \max\{\frac{1}{4}, c_1\}$, and $c_3 = \max\{1, c_2\}$. Furthermore, let $\mu^0 = \frac{\left(\mathbf{x}^0\right)^T\mathbf{s}^0}{n}$ and $\delta(\mathbf{x}^0, \mathbf{s}^0; \mu^0) \leq \tau$. Moreover, we denote by $\mathbf{x}^k$ and $\mathbf{s}^k$ the iterates obtained after k iterations of Algorithm 2. Then, $\left(\mathbf{x}^k\right)^T\mathbf{s}^k \leq \epsilon$ when*

$$k \geq 1 + \left\lceil \frac{1}{c\theta}\log\frac{(37+c_2)\left(\mathbf{x}^0\right)^T\mathbf{s}^0}{36\epsilon}\right\rceil.$$

Proof. Using $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $r \geq 2$, $c_5 \geq \frac{1}{4}$, $c_3 \geq 1$, $n \geq 1$, and $\kappa \geq 0$, we obtain

$$\delta^2 < \frac{1}{4(3+4\kappa)^2} < \frac{n}{36}. \tag{3.1.46}$$

Based on Lemma 3.1.9, we get

$$\left(\mathbf{x}^k\right)^T \mathbf{s}^k < \frac{\left(n+(c_2+1)\delta^2\right)\mu^k}{1-c\theta} < \frac{(37+c_2)n\mu^k}{36(1-c\theta)} = \frac{37+c_2}{36}(1-c\theta)^{k-1}\left(\mathbf{x}^0\right)^T \mathbf{s}^0.$$

It can be seen that if $\frac{37+c_2}{36}(1-c\theta)^{k-1}\left(\mathbf{x}^0\right)^T \mathbf{s}^0 < \epsilon$ holds, then the inequality $\left(\mathbf{x}^k\right)^T \mathbf{s}^k \leq \epsilon$ is satisfied as well. Let us take the logarithms of both sides, then we have

$$(k-1)\log(1-c\theta) + \log\frac{(37+c_2)\left(\mathbf{x}^0\right)^T \mathbf{s}^0}{36} < \log\epsilon.$$

Based on the well-known inequality $\log(1+\theta) \leq \theta$, $\theta \geq -1$, we obtain the following inequality

$$-c\theta(k-1) + \log\frac{(37+c_2)\left(\mathbf{x}^0\right)^T \mathbf{s}^0}{36} \leq \log\epsilon,$$

which proves the lemma. □

Finally, we summarize our analysis.

Theorem 3.1.1. *(Theorem 4.6. in [60]) Let φ be a function belonging to the AET class defined in Definition 3.1.1. Consider $\tau = \frac{\sqrt{1-\xi^2}}{8rc_5(c_3+2+4\kappa)}$, $r \geq 2$, and $0 < \theta \leq \frac{\sqrt{1-\xi^2}}{16qc_5c_4(c_3+2+4\kappa)\sqrt{n}}$, where $q \geq \frac{1}{2}r$, $c_5 = \max\{\frac{1}{4}, c_1\}$, and $c_3 = \max\{1, c_2\}$. Then, the PC IPA proposed in Algorithm 2 is well defined and it performs at most*

$$\mathcal{O}\left((c_3+2+4\kappa)\sqrt{n}\log\frac{(37+c_2)n\mu^0}{36\epsilon}\right)$$

iterations. The output is a pair $(\mathbf{x}, \mathbf{s})$ satisfying $\mathbf{x}^T\mathbf{s} \leq \epsilon$.

Proof. Considering Corollary 3.1.1 and Lemma 3.1.13 we get the result. □

In the next section we consider a special case of this new class of AET functions, namely $\varphi(t) = t^2 - t + \sqrt{t}$, which has an inflection point.

3.2 Predictor-corrector interior-point algorithm based on an AET function having an inflection point

As we have already mentioned, in the literature, there are many IPAs that start with a corrector step and then a predictor step because after a corrector step we reach a proper neighborhood of the central path. In our case our algorithm performs similarly. It is important to mention that the analysis of this section is presented in [59].

Our chosen AET function is $\varphi(t) = t^2 - t + \sqrt{t}$, which is a member of the class of functions defined in Definition 3.1.1. In this case, we have

$$\mathbf{a}_\varphi = \mu \frac{\mathbf{e} - \frac{\mathbf{x}^2\mathbf{s}^2}{\mu^2} + \frac{\mathbf{xs}}{\mu} - \sqrt{\frac{\mathbf{xs}}{\mu}}}{2\frac{\mathbf{xs}}{\mu} - \mathbf{e} + \frac{1}{2}\sqrt{\frac{\mu}{\mathbf{xs}}}} = \frac{4\mu^2\sqrt{\mathbf{xs}} + 2\mu\mathbf{xs}\sqrt{\mathbf{xs}} - 3\mu\mathbf{xs}\sqrt{\mu}}{2(4\mathbf{xs}\sqrt{\mathbf{xs}} - 2\mu\sqrt{\mathbf{xs}} + \mu\sqrt{\mu})} - \frac{\mathbf{xs}}{2}. \tag{3.2.1}$$

In the corrector step, we have the following $\mathbf{p}_\varphi$

$$\mathbf{p}_\varphi = \frac{\mathbf{e} - \mathbf{v}^4 + \mathbf{v}^2 - \mathbf{v}}{2\mathbf{v}^3 - \mathbf{v} + \frac{1}{2}\mathbf{e}} = \frac{2(\mathbf{e} - \mathbf{v})(\mathbf{e} + \mathbf{v}^2 + \mathbf{v}^3)}{4\mathbf{v}^3 - 2\mathbf{v} + \mathbf{e}}. \tag{3.2.2}$$

For system (1.4.7) with (3.2.2) we obtain the following solution:

$$\mathbf{d}_x^c = (I + \bar{M})^{-1}\mathbf{p}_\varphi, \quad \mathbf{d}_s^c = \bar{M}(I + \bar{M})^{-1}\mathbf{p}_\varphi.$$

Using (1.4.5) it is easy to calculate the search directions $\Delta^c\mathbf{x}$ and $\Delta^c\mathbf{s}$. The new corrector iterates are

$$\mathbf{x}^+ = \mathbf{x} + \Delta^c\mathbf{x}, \quad \mathbf{s}^+ = \mathbf{s} + \Delta^c\mathbf{s}.$$

In case of the predictor step, we introduced the scaled predictor system in (3.1.3) in Section 3.1. Taking into consideration the special case of $\varphi(t) = t^2 - t + \sqrt{t}$ we get the system

$$\begin{aligned} -\bar{M}_+\mathbf{d}_x^p + \mathbf{d}_s^p &= \mathbf{0}, \\ \mathbf{d}_x^p + \mathbf{d}_s^p &= -\frac{1}{2}\mathbf{v}^+, \end{aligned} \tag{3.2.3}$$

where $\mathbf{v}^+ = \sqrt{\frac{\mathbf{x}^+\mathbf{s}^+}{\mu}}$, $\mathbf{d}^+ = \sqrt{\frac{\mathbf{x}^+}{\mathbf{s}^+}}$, $D^+ = \mathrm{diag}(\mathbf{d}^+)$, $\bar{M}_+ = D^+MD^+$. The solution of the above system is:

$$\mathbf{d}_x^p = -\frac{1}{2}(I + \bar{M}_+)^{-1}\mathbf{v}^+ \quad \text{and} \quad \mathbf{d}_s^p = -\frac{1}{2}\bar{M}_+(I + \bar{M}_+)^{-1}\mathbf{v}^+.$$

From this solution using (1.4.5) it is easy to get the predictor search directions $\Delta^p \mathbf{x}$ and $\Delta^p \mathbf{s}$. The new predictor iterates are:

$$\mathbf{x}^p = \mathbf{x}^+ + \theta \Delta^p \mathbf{x}, \quad \mathbf{s}^p = \mathbf{s}^+ + \theta \Delta^p \mathbf{s}, \quad \mu^p = \left(1 - \frac{\theta}{2}\right) \mu,$$

where $\theta \in (0,1)$ is the update parameter.

The proximity measure is similar to the one that we introduced in 2.2.1 from Section 2.2, but with considering the special AET function

$$\delta(\mathbf{x}, \mathbf{s}, \mu) = \frac{\|\mathbf{p}_\varphi\|}{2} = \left\| \frac{(\mathbf{e} - \mathbf{v})(\mathbf{e} + \mathbf{v}^2 + \mathbf{v}^3)}{4\mathbf{v}^3 - 2\mathbf{v} + \mathbf{e}} \right\|. \tag{3.2.4}$$

The neighborhood is similar to the one given in 2.2.2.

The PC IPA for solving $P_*(\kappa)$-LCPs is given in Algorithm 3.

Now, let us present the complexity analysis of the proposed Algorithm 3.

3.2.1 Analysis of Algorithm 3

Let us consider the complexity analysis of the proposed PC IPA shown in Algorithm 3. First, we examine the corrector step. Since this is a full-Newton step, this part is very similar to the to the analysis of the IPA given in Algorithm 2, but in the special case of $\varphi(t) = t^2 - t + \sqrt{t}$. We list the lemmas and corollaries used in this case without proofs.

First, using Lemma 2.3.1 from Chapter 2 in the special case of $\varphi(t) = t^2 - t + \sqrt{t}$, we get the result regarding the strict feasibility of the corrector step.

Lemma 3.2.1. *(Lemma 4.1 in [59]) Let $(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{F}^+$ be given, such that $\delta(\boldsymbol{x}, \boldsymbol{s}; \mu) < \frac{1}{\sqrt{1+4\kappa}}$. In case of $\varphi(t) = t^2 - t + \sqrt{t}$ we have that $(\mathbf{x}^+, \mathbf{s}^+) \in \mathcal{F}^+$.*

Considering Lemma 3.1.3 in the special case of $\varphi(t) = t^2 - t + \sqrt{t}$, we obtain the quadratic convergence of the corrector step.

Lemma 3.2.2. *(Lemma 4.2 in [59]) Let $(\boldsymbol{x}, \boldsymbol{s}) \in \mathcal{F}^+$ be given, such that $\delta(\boldsymbol{x}, \boldsymbol{s}; \mu) \le \frac{1}{\sqrt{1+4\kappa}}$. In case of $\varphi(t) = t^2 - t + \sqrt{t}$ we have that*

$$\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) \le 2(10 + 4\kappa)\delta(\boldsymbol{x}, \boldsymbol{s}; \mu)^2.$$

Algorithm 3 New PC IPA for $P_*(\kappa)$-LCPs [59]

Input:

Let $\epsilon > 0$ *be the accuracy parameter;*
$0 < \theta < 1$ *the update parameter;*
τ *the proximity parameter;*
A known upper bound κ *of the handicap* $\hat{\kappa}(M)$ *is given.*
Assume that for $(\mathbf{x}^0, \mathbf{s}^0)$ *the* $\left(\mathbf{x}^0\right)^T \mathbf{s}^0 = n\mu^0$, $\mu^0 > 0$ *holds such that* $\delta(\mathbf{x}^0, \mathbf{s}^0; \mu^0) \leq \tau$.

Iteration:

begin
  $k := 0$;
  while $\left(\mathbf{x}^k\right)^T \mathbf{s}^k > \epsilon$ **do begin**

   (corrector step)
   compute $(\Delta^c x^k, \Delta^c s^k)$ *from system (1.4.7) using (1.4.5);*
   update $\left(\mathbf{x}^+\right)^k := \mathbf{x}^k + \Delta^c x^k$ *and* $\left(\mathbf{s}^+\right)^k := \mathbf{s}^k + \Delta^c s^k$;

   (predictor step)
   compute $(\Delta^p x^k, \Delta^p s^k)$ *from system (3.2.3) using (1.4.5);*
   update $(\mathbf{x}^p)^k := \left(\mathbf{x}^+\right)^k + \theta \Delta^p \mathbf{x}^k$ *and* $(\mathbf{s}^p)^k := \left(\mathbf{s}^+\right)^k + \theta \Delta^p \mathbf{s}^k$;

   (update of the parameters and the iterates)
   $(\mu^p)^k = \left(1 - \frac{\theta}{2}\right) \mu^k$;
   $\mathbf{x}^{k+1} := (\mathbf{x}^p)^k, \quad \mathbf{s}^{k+1} := (\mathbf{s}^p)^k, \quad \mu^{k+1} := (\mu^p)^k$;
   $k := k+1$;

  end
end.

From Lemma 3.1.4 considering $\varphi(t) = t^2 - t + \sqrt{t}$, we get an upper bound on the duality gap.

Lemma 3.2.3. *(Lemma 4.3 in [59]) Let* $\delta := \delta(\boldsymbol{x}, \boldsymbol{s}; \mu)$ *and suppose that the vectors* $\boldsymbol{x}^+$ *and* $\boldsymbol{s}^+$ *are obtained using a full-Newton step, thus,* $\boldsymbol{x}^+ = \boldsymbol{x} + \Delta \boldsymbol{x}$ *and* $\boldsymbol{s}^+ = \boldsymbol{s} + \Delta \boldsymbol{s}$. *In case of* $\varphi(t) = t^2 - t + \sqrt{t}$ *we have*

$$\left(\boldsymbol{x}^+\right)^T \boldsymbol{s}^+ \leq \mu(n + 9\delta^2).$$

Lemma 3.2.4. *(Lemma 4.6 in [59]) Let* $\varphi(t) = t^2 - t + \sqrt{t}$. *Then,*

$$\|\mathbf{d}_x^p \mathbf{d}_s^p\| \leq \frac{n\left(1 + \frac{\kappa}{2}\right)(1 + 4\delta^+)^2}{2},$$

where $\delta^+ = \delta(\mathbf{x}^+, \mathbf{s}^+; \mu) = \left\| \dfrac{(\mathbf{e} - \mathbf{v}^+)(\mathbf{e} + (\mathbf{v}^+)^2 + (\mathbf{v}^+)^3)}{4(\mathbf{v}^+)^3 - 2\mathbf{v}^+ + \mathbf{e}} \right\|$.

We show that after a predictor step we have strictly feasible iterates.

Lemma 3.2.5. *(Lemma 4.7 in [59]) Let $(\mathbf{x}^+, \mathbf{s}^+) > \mathbf{0}$ and $\mu > 0$ such that $\delta^+ := \delta(\mathbf{x}^+, \mathbf{s}^+; \mu) < \frac{1}{4}$. Furthermore, let $0 < \theta < 1$. Let $\mathbf{x}^p = \mathbf{x}^+ + \theta \Delta^p \mathbf{x}$, $\mathbf{s}^p = \mathbf{s}^+ + \theta \Delta^p \mathbf{s}$ be the iterates after a predictor step. Then, in case of $\varphi(t) = t^2 - t + \sqrt{t}$, we have $\mathbf{x}^p, \mathbf{s}^p > \mathbf{0}$ if $u(\delta^+, \theta, n) > 0$, where*

$$u(\delta^+, \theta, n) := (1 - 4\,\delta^+)^2 - \frac{n\left(1 + \frac{\kappa}{2}\right)(1 + 4\,\delta^+)^2\,\theta^2}{2 - \theta}.$$

Let

$$\mathbf{v}^p = \sqrt{\frac{\mathbf{x}^p \mathbf{s}^p}{\mu^p}}, \tag{3.2.5}$$

where $\mu^p = \left(1 - \frac{\theta}{2}\right)\mu$. In the following lemma the effect of a predictor step and the update of μ on the proximity measure is analysed.

Lemma 3.2.6. *(Lemma 4.8 in [59]) Let $\varphi(t) = t^2 - t + \sqrt{t}$, $\delta^+ := \delta(\mathbf{x}^+, \mathbf{s}^+; \mu) < \frac{1}{4}$, $\mu^p = \left(1 - \frac{\theta}{2}\right)\mu$, where $0 < \theta < 1$, and let $\mathbf{x}^p$ and $\mathbf{s}^p$ denote the iterates after a predictor step. Then,*

$$\delta^p := \delta(\mathbf{x}^p, \mathbf{s}^p; \mu^p) \leq 2(10 + 4\kappa)\delta^2 + \frac{2\theta^2}{2 - \theta} n \left(1 + \frac{\kappa}{2}\right)(1 + 4\delta^+)^2,$$

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

Next, let us consider the upper bound of the duality gap after a predictor step.

Lemma 3.2.7. *(Lemma 4.9 in [59]) Let $\varphi(t) = t^2 - t + \sqrt{t}$ and $0 < \theta < 1$. If $\mathbf{x}^p$ and $\mathbf{s}^p$ are the iterates obtained after the predictor step of the algorithm, then*

$$(\mathbf{x}^p)^T \mathbf{s}^p \leq \left(1 - \frac{\theta}{2} + \frac{\theta^2}{8}\right)\left(\mathbf{x}^+\right)^T \mathbf{s}^+ < \frac{2(n + 9\delta^2)\mu^p}{2 - \theta}.$$

Finally, in the last lemma we provide a bound to the iteration number of the PC IPA.

Lemma 3.2.8. *(Lemma 5.5 in [59]) Let $\varphi(t) = t^2 - t + \sqrt{t}$, $(\mathbf{x}^0, \mathbf{s}^0) \in \mathcal{F}^+$, $\tau = \frac{1}{2r(10+4\kappa)}$, where $r \geq 2$ and $0 < \theta \leq \frac{1}{q(10+4\kappa)\sqrt{n}}$, where $q \geq \frac{5}{2}r$. Furthermore, let $\mu^0 = \frac{\left(\mathbf{x}^0\right)^T \mathbf{s}^0}{n}$ and $\delta(\mathbf{x}^0, \mathbf{s}^0; \mu^0) \leq \tau$. Moreover, let $\mathbf{x}^k$ and $\mathbf{s}^k$ be the iterates obtained after k iterations of Algorithm 3. Then, $\left(\mathbf{x}^k\right)^T \mathbf{s}^k \leq \epsilon$ when*

$$k \geq 1 + \left\lceil \frac{2}{\theta} \log \frac{3\left(\mathbf{x}^0\right)^T \mathbf{s}^0}{2\epsilon} \right\rceil.$$

We summarize these results and give a theoretical upper bound on the iteration complexity in the last theorem.

Theorem 3.2.1. *(Theorem 5.6 in [59]) Let* $\varphi(t) = t^2 - t + \sqrt{t}$, $\tau = \frac{1}{2r(10+4\kappa)}$, *where* $r \geq 2$ *and* $0 < \theta \leq \frac{1}{q(10+4\kappa)\sqrt{n}}$, *where* $q \geq \frac{5}{2}r$*. Then, the PC IPA proposed in Algorithm 3 is well defined and it performs at most*

$$\mathcal{O}\left((10+4\kappa)\sqrt{n}\log\frac{3n\mu^0}{2\epsilon}\right)$$

iterations. The output is a pair $(\mathbf{x},\mathbf{s})$ *satisfying* $\mathbf{x}^T\mathbf{s} \leq \epsilon$.

The numerical results obtained by using Algorithm 3 are presented in Chapter 5.

In the following chapter we present a new PC IPA acting in a wide neighborhood of the central path [57].

Chapter 4

Predictor-corrector interior-point algorithm acting in a wide neighborhood of the central path

In this chapter, we introduce a new PC IPA that works in a wide neighborhood of the central path [54, 57]. Throughout the whole chapter, we consider the AET function $\varphi(t) = \sqrt{t}$ case in the generalized wide neighborhood approach introduced in Section 4.1. In this way, we obtain the first IPA, where not only the central path system is transformed, but the definition of the neighborhood is also modified taking into consideration the AET approach. This gives a new direction in the research of IPAs. We also present the complexity analysis of the PC IPA.

4.1 Generalized wide neighborhoods

In this section, we examine some of the already existing wide neighborhoods of the central path in addition to our previously introduced generalized wide neighborhood [57]. Consider $(\mathbf{x}, \mathbf{s}) \in \mathcal{F}$. We define the normalized duality gap as follows

$$\mu(\mathbf{x}, \mathbf{s}) := \frac{\mathbf{x}^T \mathbf{s}}{n}. \tag{4.1.1}$$

Furthermore, we consider the following notation:

$$\mathbf{u} := \frac{\mathbf{x}\mathbf{s}}{\mu(\mathbf{x}, \mathbf{s})}.$$

We introduce a generalized form of the proximity measure:

$$\delta^{-}_{\infty,\varphi}(\mathbf{x},\mathbf{s}) := \left\| [\varphi(\mathbf{u}) - \varphi(\mathbf{e})]^{-} \right\|_{\infty}.$$

Using the introduced proximity measure and considering the AET approach, we have the generalized wide neighborhood of the (CCP_{φ}) as:

$$\mathcal{N}^{-}_{\infty,\varphi}(\alpha) := \{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^{+} : \delta^{-}_{\infty,\varphi}(\mathbf{x},\mathbf{s}) \leq \alpha\}. \tag{4.1.2}$$

Taking into consideration the case of $\varphi(t) = t$ we obtain the wide neighborhood used by Potra and Liu [88]:

$$\mathcal{N}^{-}_{\infty}(\alpha) := \{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^{+} : \delta^{-}_{\infty}(\mathbf{x},\mathbf{s}) \leq \alpha\}. \tag{4.1.3}$$

In [57] we presented another, generalized wide neighborhood for (CCP_{φ}):

$$\mathcal{D}_{\varphi}(\beta) := \{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^{+} : \varphi(\mathbf{u}) \geq \beta\varphi(\mathbf{e}),\ \beta \in (0,1)\}. \tag{4.1.4}$$

It can be seen that if we consider $\varphi(t) = t$, we get the wide neighborhood used by Potra and Liu [88]:

$$\mathcal{D}(\beta) := \{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^{+} : \mathbf{u} \geq \beta\mathbf{e}\}. \tag{4.1.5}$$

In Lemma 3.1. in [57] we showed that which AET function can be used to define a generalized wide neighborhood based on (4.1.4).

Lemma 4.1.1. *(Lemma 3.1. in [57]) Let $(\boldsymbol{x},\boldsymbol{s}) \in \mathcal{F}^{+}$ and $\alpha \in (0,1)$. Then, in the case of $\varphi(t) = t$ and $\varphi(t) = \sqrt{t}$ we have $\mathcal{N}^{-}_{\infty,\varphi}(\alpha) = \mathcal{D}_{\varphi}(1-\alpha)$. In the case of $\varphi(t) = t - \sqrt{t}$ we have $\mathcal{D}_{\varphi}(1-\alpha) \subseteq \mathcal{N}^{-}_{\infty,\varphi}(\alpha)$.*

Proof. Let us consider the case of $\varphi(t) = t$:

$$\begin{aligned}
(\mathbf{x},\mathbf{s}) \in \mathcal{D}(1-\alpha) \quad &\Longleftrightarrow \quad \mathbf{xs} \geq (1-\alpha)\mu(\mathbf{x},\mathbf{s})\mathbf{e} = \mu(\mathbf{x},\mathbf{s})\mathbf{e} - \alpha\mu(\mathbf{x},\mathbf{s})\mathbf{e} \\
&\Longleftrightarrow \quad \mathbf{u} - \mathbf{e} \geq -\alpha\mathbf{e} \Longleftrightarrow \left\| [\mathbf{u} - \mathbf{e}]^{-} \right\|_{\infty} \leq \alpha \\
&\Longleftrightarrow \quad (\mathbf{x},\mathbf{s}) \in \mathcal{N}^{-}_{\infty}(\alpha).
\end{aligned}$$

Now, we extend it to the general case. It can be seen that

$$
\begin{aligned}
(\mathbf{x},\mathbf{s}) \in \mathcal{N}^-_{\infty,\varphi}(\alpha) \quad &\Longleftrightarrow \quad \left\| [\varphi(\mathbf{u}) - \varphi(\mathbf{e})]^- \right\|_\infty \leq \alpha \\
&\Longleftrightarrow \quad \varphi(\mathbf{u}) - \varphi(\mathbf{e}) \geq -\alpha\,\mathbf{e} \Longleftrightarrow \varphi(\mathbf{u}) \geq \varphi(\mathbf{e}) - \alpha\mathbf{e}.
\end{aligned}
$$

Moreover, we have

$$
(\mathbf{x},\mathbf{s}) \in \mathcal{D}_\varphi(1-\alpha) \quad \Longleftrightarrow \quad \varphi(\mathbf{u}) \geq (1-\alpha)\varphi(\mathbf{e}) = \varphi(\mathbf{e}) - \alpha\varphi(\mathbf{e}).
$$

This shows that $\varphi(\mathbf{e}) = \mathbf{e}$ needs to be satisfied in order to get $\mathcal{N}^-_{\infty,\varphi}(\alpha) = \mathcal{D}_\varphi(1-\alpha)$. Taking into consideration the case of $\varphi(t) = \sqrt{t}$, we have $\varphi(\mathbf{e}) = \mathbf{e}$, therefore $\mathcal{N}^-_{\infty,\varphi}(\alpha) = \mathcal{D}_\varphi(1-\alpha)$. On the other hand, it can be seen that if $\varphi(t) = t - \sqrt{t}$ we can only prove that $\mathcal{D}_\varphi(1-\alpha) \subseteq \mathcal{N}^-_{\infty,\varphi}(\alpha)$. $\square$

Now let us introduce our PC IPA that works in the wide neighborhood of the central path.

4.2 New predictor-corrector interior-point algorithm

To obtain the Newton system, we use the function $\varphi(t) = \sqrt{t}$ and substitute it into (1.4.2).We obtain

$$
\begin{aligned}
-M\Delta\mathbf{x} + \Delta\mathbf{s} &= \mathbf{0}, \\
\mathbf{s}\Delta\mathbf{x} + \mathbf{x}\Delta\mathbf{s} &= 2\left(\sqrt{\mu\mathbf{x}\mathbf{s}} - \mathbf{x}\mathbf{s}\right).
\end{aligned} \tag{4.2.1}
$$

As we have seen in Chapter 3 and [25], the predictor step is a greedy one. Hence, we set $\mu = 0$ and we obtain

$$
\begin{aligned}
-M\Delta^p\mathbf{x} + \Delta^p\mathbf{s} &= \mathbf{0}, \\
\mathbf{s}\Delta^p\mathbf{x} + \mathbf{x}\Delta^p\mathbf{s} &= -2\mathbf{x}\mathbf{s},
\end{aligned} \tag{4.2.2}
$$

where $(\Delta^p\mathbf{x}, \Delta^p\mathbf{s})$ are the predictor search directions.

Let us take into account some of the results of Potra and Liu in [88]. Lemmas 3.2 and 3.3 from [88] play an important role in the following analysis.

Lemma 4.2.1. *(Lemma 3.2 in [88]) Assume that we have a $\mathcal{P}_*(\kappa)$-LCP and let $(\Delta\mathbf{x}, \Delta\mathbf{s})$ be* *the solution of the following linear system:*

$$-M\Delta\mathbf{x}+\Delta\mathbf{s}=\mathbf{0},$$
$$\mathbf{s}\Delta\mathbf{x}+\mathbf{x}\Delta\mathbf{s}=\mathbf{a},$$

where $(\Delta\mathbf{x},\Delta\mathbf{s})\in\mathbb{R}^{2n}_{+}$ *and* $\mathbf{a}\in\mathbb{R}^n$ *are given. Defining*

$$\mathcal{K}_{+}=\{i:\Delta x_i\Delta s_i>0\}\qquad \textit{and}\qquad \mathcal{K}_{-}=\{i:\Delta x_i\Delta s_i<0\}$$

we have

$$\frac{1}{1+4\kappa}\|\Delta\mathbf{x}\Delta\mathbf{s}\|_{\infty}\leq\sum_{i\in\mathcal{K}_{+}}\Delta x_i\Delta s_i\leq\frac{1}{4}\left\|(\mathbf{x}\mathbf{s})^{-\frac{1}{2}}\mathbf{a}\right\|_2^2. \tag{4.2.3}$$

Lemma 4.2.2. *(Lemma 3.3 in [88]) Assume that we have a* $\mathcal{P}_*(\kappa)$*-LCP and let* $(\Delta\mathbf{x},\Delta\mathbf{s})$ *be the solution of the following linear system:*

$$-M\Delta\mathbf{x}+\Delta\mathbf{s}=\mathbf{0},$$
$$\mathbf{s}\Delta\mathbf{x}+\mathbf{x}\Delta\mathbf{s}=\mathbf{a},$$

where $(\Delta\mathbf{x},\Delta\mathbf{s})\in\mathbb{R}^{2n}_{+}$ *and* $\mathbf{a}\in\mathbb{R}^n$ *are given. Then, the following inequality holds:*

$$\Delta\mathbf{x}^T\Delta\mathbf{s}\geq-\kappa\left\|(\mathbf{x}\mathbf{s})^{-\frac{1}{2}}\mathbf{a}\right\|_2^2. \tag{4.2.4}$$

First, we analyse the predictor step of our PC IPA that works in the wide neighborhood of the central path.

Let $(\mathbf{x},\mathbf{s})\in\mathcal{N}^{-}_{\infty,\varphi}(1-\beta)$, where $\beta\in(0,1)$. Using system (4.2.2) we can compute the predictor search directions $(\Delta^p\mathbf{x},\Delta^p\mathbf{s})$. Then, we can get the predictor iterate as

$$\mathbf{x}^p(\alpha)=\mathbf{x}+\alpha\Delta^p\mathbf{x}\quad\text{ and }\quad\mathbf{s}^p(\alpha)=\mathbf{s}+\alpha\Delta^p\mathbf{s}. \tag{4.2.5}$$

Considering (4.1.1) and (4.2.2) we obtain

$$\begin{aligned}\mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha) &= (1-2\alpha)\mathbf{x}\mathbf{s} + \alpha^2\Delta^p\mathbf{x}\Delta^p\mathbf{s},\\ \mu_p(\alpha) &= (1-2\alpha)\mu(\mathbf{x},\mathbf{s}) + \frac{\alpha^2\Delta^p\mathbf{x}^T\Delta^p\mathbf{s}}{n},\end{aligned} \tag{4.2.6}$$

where $\mu_p(\alpha) = \mu(\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha))$. Furthermore, we have $-M\mathbf{x}^p(\alpha) + \mathbf{s}^p(\alpha) = \mathbf{q}$ as well. We want the step size $\alpha > 0$ to satisfy the following

$$(\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha)) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta+\beta\gamma) = \mathcal{D}_\varphi((1-\gamma)\beta),$$

where

$$\gamma = \frac{1-\beta}{5((1+4\kappa)n+1)}. \tag{4.2.7}$$

It is important to note that Potra and Liu in [88] used the parameter $\gamma = \frac{1-\beta}{(1+4\kappa)n+1}$. However, based on our complexity analysis we chose (4.2.7) for our PC IPA based on the function $\varphi(t) = \sqrt{t}$. It is important to mention that other values for parameter γ could satisfy the complexity analysis as well.

We want to determine the largest α such that $(\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha)) \in \mathcal{F}^+$ that satisfies the following inequality

$$\sqrt{\frac{\mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha)}{\mu(\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha))}} \geq (1-\gamma)\,\beta\,\mathbf{e}. \tag{4.2.8}$$

Hence, we have

$$\frac{(\mathbf{x}+\alpha\Delta^p\mathbf{x})\,(\mathbf{s}+\alpha\Delta^p\mathbf{s})}{(\mathbf{x}+\alpha\Delta^p\mathbf{x})^T(\mathbf{s}+\alpha\Delta^p\mathbf{s})} \geq \frac{(1-\gamma)^2\beta^2}{n}\,\mathbf{e}. \tag{4.2.9}$$

In order to assure that $(\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha)) \in \mathcal{F}^+$ we define $\alpha_F = \min\{\alpha_x, \alpha_s\}$, where

$$\alpha_x = \min\left\{-\frac{x_i}{\Delta^p x_i} : \Delta^p x_i < 0\right\} \quad \text{and} \quad \alpha_s = \min\left\{-\frac{s_i}{\Delta^p s_i} : \Delta^p s_i < 0\right\}.$$

It can be seen that $\alpha_F > 0$ is the largest step that ensures the feasibility of the new predictor solution. Considering (4.2.9) we obtain further restrictions:

$$\tilde{a}_i\,\alpha^2 - \tilde{b}_i\,\alpha + \tilde{c}_i \geq 0, \qquad \forall i, \tag{4.2.10}$$

where $\tilde{a}_i = \Delta^p x_i\,\Delta^p s_i - (1-\gamma)^2\beta^2\mu(\Delta^p\mathbf{x}, \Delta^p\mathbf{s})$, $\tilde{b}_i = 2\,(x_i\,s_i - (1-\gamma)^2\beta^2\mu(\mathbf{x},\mathbf{s}))$, and $\tilde{c}_i = \frac{\tilde{b}_i}{2}, \forall i$.

Based on $(\mathbf{x},\mathbf{s}) \in \mathcal{D}_\varphi(\beta)$ we get $\tilde{b}_i \geq 0, \forall i$. This means that $\alpha = 0$ satisfy all inequalities.

After some calculations, we get that inequality (4.2.10) is satisfied if $\alpha \in (0, \alpha_{p_i}]$, where

$$\alpha_{p_i} = \begin{cases} \infty, & \text{if } \Delta_i \leq 0 \\ \frac{1}{2}, & \text{if } \tilde{a}_i = 0 \\ \zeta_i, & \text{if } \Delta_i > 0 \text{ and } \tilde{a}_i \neq 0, \end{cases} \tag{4.2.11}$$

where

$$\Delta_i = \tilde{b}_i^2 - 4\tilde{a}_i\tilde{c}_i = \tilde{b}_i^2 - 2\tilde{a}_i\tilde{b}_i$$

and

$$\zeta_i = \frac{\tilde{b}_i - \sqrt{\Delta_i}}{2\tilde{a}_i} = \frac{\tilde{b}_i^2 - \Delta_i}{2\tilde{a}_i(\tilde{b}_i + \sqrt{\Delta_i})} = \frac{\tilde{b}_i}{\tilde{b}_i + \sqrt{\Delta_i}} = \frac{1}{1 + \sqrt{1 - 2\frac{\tilde{a}_i}{\tilde{b}_i}}}.$$

Considering

$$\alpha_N = \min\{\alpha_{p_i} : \ 1, \ldots, n\} \tag{4.2.12}$$

we have a predictor step length which ensures that the iterates after the predictor step stay in the predictor neighborhood.

Finally, the step length $\alpha > 0$ has to fulfill $\mu_p(\alpha) > 0$ as well. Considering $(\mathbf{x},\mathbf{s}) \in \mathcal{D}_\varphi(\beta)$, where $\beta \in (0,1)$, we have vector $\mathbf{u}$ that is defined earlier as

$$\mathbf{u} = \frac{\mathbf{x}\mathbf{s}}{\mu(\mathbf{x},\mathbf{s})} \quad \text{and} \quad \mathbf{v} = \frac{\Delta^p\mathbf{x}\,\Delta^p\mathbf{s}}{\mu(\mathbf{x},\mathbf{s})}. \tag{4.2.13}$$

Using Lemma 4.2.1 and Lemma 4.2.2 we have the following bounds for $\mathbf{v}$

$$\|\mathbf{v}\|_\infty \leq (1+4\kappa)n \quad \text{and} \quad -4\kappa n \leq \mathbf{e}^T\mathbf{v} \leq \sum_{i \in \mathcal{I}_+} v_i \leq n. \tag{4.2.14}$$

Based on $\frac{\mu(\Delta^p\mathbf{x}, \Delta^p\mathbf{s})}{\mu(\mathbf{x},\mathbf{s})} = \frac{\mathbf{e}^T\mathbf{v}}{n}$ and (4.2.6), for the quadratic equation defined by $\mu_p(\alpha) = 0$, we have

$$\frac{\mathbf{e}^T\mathbf{v}}{n}\alpha^2 - 2\alpha + 1 = 0, \tag{4.2.15}$$

hence, the discriminant denoted by Δ_μ is

$$\Delta_\mu = 4\left(1 - \frac{\mathbf{e}^T\mathbf{v}}{n}\right).$$

Taking into account (4.2.14), it can be seen that $-4\kappa \leq \frac{\mathbf{e}^T\mathbf{v}}{n} \leq 1$, which shows that Δ_μ is always nonnegative. Furthermore, the smallest positive root is:

$$\alpha_0 = \frac{2-\sqrt{4-4\frac{\mathbf{e}^T\mathbf{v}}{n}}}{2\frac{\mathbf{e}^T\mathbf{v}}{n}} = \frac{1}{1+\sqrt{1-\frac{\mathbf{e}^T\mathbf{v}}{n}}}. \tag{4.2.16}$$

This means that we have

$$\mu_p(\alpha) > \mu_p(\alpha_0) = 0, \text{ for all } 0 \leq \alpha < \alpha_0. \tag{4.2.17}$$

Considering the previous statements, we can introduce the step length α_p that fulfills all the previously mentioned conditions, the interior point condition, the neighborhood condition, and ensures decrease in the centrality parameter, as :

$$\alpha_p = \min\{\alpha_F, \alpha_N, \alpha_0\}. \tag{4.2.18}$$

Considering α_F it can be seen that $\alpha_F \geq \alpha_N$. Moreover, based on α_p, we get

$$\sqrt{\mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha)} \geq (1-\gamma)\beta\sqrt{\mu_p(\alpha)} > (1-\gamma)\beta\sqrt{\mu_p(\alpha_p)} \geq 0, \tag{4.2.19}$$

where for all $0 \leq \alpha < \alpha_p$. Taking into consideration the standard continuity argument we obtain $\mathbf{x}^p(\alpha) > \mathbf{0}$ and $\mathbf{s}^p(\alpha) > \mathbf{0}$, $\forall \alpha \in (0, \alpha_p)$. Therefore $(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{F}^+$, where $\mathbf{x}^p = \mathbf{x}^p(\alpha_p)$ and $\mathbf{s}^p = \mathbf{s}^p(\alpha_p)$.

Previous results are summarized in Lemma 4.3. in [57].

Lemma 4.2.3. *(Lemma 4.3. [57]) Let* $(\mathbf{x}, \mathbf{s}) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta) = \mathcal{D}_\varphi(\beta)$. *Then,* $\exists \alpha_p > 0$ *predictor steplength*

$$(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta+\beta\gamma) = \mathcal{D}_\varphi((1-\gamma)\beta)$$

holds, where $\beta, \gamma \in (0,1)$ *are given parameters.*

It is important to note that computing the exact value of α_p is time consuming in each iteration. Hence, α_p can be calculated by using the following optimization problem

$$\bar{\alpha}_p = \sup\{\hat{\alpha} > 0 : (\mathbf{x}^p(\alpha), \mathbf{s}^p(\alpha)) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta+\beta\gamma), \forall \alpha \in [0, \hat{\alpha}]\}. \tag{4.2.20}$$

It is important to note that we did not prove that α_p and $\bar{\alpha}_p$ are equal. We considered only an $\hat{\alpha}$ from (4.2.20) that is a lower bound on the value α_p and is based only on the values κ, n and β. In that case we can use $\hat{\alpha}$ instead of α_p.

First, we consider the following technical lemma.

Lemma 4.2.4. *(Lemma 4.4. [57]) Let $\mathbf{u} = \frac{\mathbf{xs}}{\mu(\mathbf{xs})}$, where $(\mathbf{x},\mathbf{s}) \in \mathcal{D}_\varphi(\beta)$, $\beta \in (0,1)$, and $\gamma = \frac{1-\beta}{5((1+4\kappa)n+1)}$. Then, we have*

$$u_i - ((1-\gamma)\beta)^2 \geq \beta^2\gamma.$$

Proof. Let us consider $(\mathbf{x},\mathbf{s}) \in \mathcal{D}_\varphi(\beta)$ as an input for the predictor step. Then we have

$$u_i - ((1-\gamma)\beta)^2 = u_i - \beta^2 + 2\beta^2\gamma - \beta^2\gamma^2 \geq 2\beta^2\gamma - \beta^2\gamma^2.$$

From this we obtain

$$2\beta^2\gamma - \beta^2\gamma^2 \geq \beta^2\gamma, \tag{4.2.21}$$

therefore, we have $\gamma \geq \gamma^2$. This holds for each $\gamma < 1$. Based on (4.2.7) and $0 < \beta < 1$ the proof of the lemma follows. □

In the next lemma, we calculate the lower bound $\hat{\alpha}$ of α_p that is only based on the values κ, n and β.

Lemma 4.2.5. *(Lemma 4.5. [57]) Let $(\mathbf{x},\mathbf{s}) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta) = \mathcal{D}_\varphi(\beta)$. Then,*

$$\alpha_p > \hat{\alpha},$$

where $\hat{\alpha} = \dfrac{\beta\sqrt{1-\beta}}{5((1+4\kappa)n+2)}$.

Proof. From Lemma 4.2.4, we have

$$\frac{\tilde{b}_i}{\mu(\mathbf{x},\mathbf{s})} = 2\left(u_i - ((1-\gamma)\beta)^2\right) \geq 2\beta^2\gamma > 0.$$

Using $\frac{\mu(\Delta^p\mathbf{x},\Delta^p\mathbf{s})}{\mu(\mathbf{x},\mathbf{s})} = \frac{\mathbf{e}^T\mathbf{v}}{n}$ and (4.2.14), we obtain

$$-\frac{\tilde{a}_i}{\mu(\mathbf{x},\mathbf{s})} = -v_i + ((1-\gamma)\beta)^2\frac{\mathbf{e}^T\mathbf{v}}{n} \leq \|\mathbf{v}\|_\infty + ((1-\gamma)\beta)^2 \leq \|\mathbf{v}\|_\infty + 1.$$

Let $\tilde{a}_i \neq 0$. Taking into account $u_i - ((1-\gamma)\beta)^2 \geq \beta^2\gamma > 0$ and Lemmas 4.2.1 and 4.2.2, we get the following

$$\zeta_i = \frac{1}{1+\sqrt{1+\frac{-v_i+((1-\gamma)\beta)^2 \frac{\mathbf{e}^T\mathbf{v}}{n}}{u_i-((1-\gamma)\beta)^2}}} \geq \frac{1}{1+\sqrt{1+\frac{\|\mathbf{v}\|_\infty+1}{\beta^2\gamma}}} \geq \frac{1}{1+\sqrt{1+\frac{(1+4\kappa)n+1}{\beta^2\gamma}}},$$

if the discriminant Δ_i of the quadratic equation of (4.2.10) is positive. Based on (4.2.7) we have the lower bound as the following

$$\zeta_i \geq \frac{\beta\sqrt{1-\beta}}{\beta\sqrt{1-\beta}+\sqrt{\beta^2(1-\beta)+5((1+4\kappa)n+1)^2}}.$$

The denominator of the previous fraction has the following upper bound

$$\begin{aligned}\beta\sqrt{1-\beta}+\sqrt{\beta^2(1-\beta)+5((1+4\kappa)n+1)^2} &\leq \frac{1}{2}+\sqrt{5((1+4\kappa)n+1)^2+\frac{1}{4}} \\ &< 5((1+4\kappa)n+2),\end{aligned}$$

where we used that $\beta\sqrt{1-\beta} \leq \frac{1}{2}$. This shows that ζ_i is depending only on the parameters β, κ and the problem size n

$$\zeta_i > \hat{\alpha} := \frac{\beta\sqrt{1-\beta}}{5((1+4\kappa)n+2)}. \tag{4.2.22}$$

Taking into consideration the definition of $\mu_p(\alpha)$, the quadratic equation $\mu_p(\alpha) = 0$ can be written as (4.2.15) and the smallest root α_0 as in (4.2.16). Since (4.2.14) holds, the smallest root α_0 satisfies the following lower bound

$$\alpha_0 = \frac{1}{1+\sqrt{1-\frac{\mathbf{e}^T\mathbf{v}}{n}}} \geq \frac{1}{1+\sqrt{1+4\kappa}} > \hat{\alpha}. \tag{4.2.23}$$

This proves that for the predictor step length we have $\alpha_p > \hat{\alpha}$. □

In the next part of this subsection we deal with the corrector step. After the predictor step, from (4.2.5), (4.2.12) and Lemma 4.2.3 we obtain

$$(\mathbf{x}^p, \mathbf{s}^p) = (\mathbf{x}^p(\alpha_p), \mathbf{s}^p(\alpha_p)) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta+\beta\gamma) = \mathcal{D}_\varphi((1-\gamma)\beta). \tag{4.2.24}$$

The iterates given in (4.2.24) will be the input of the corrector step. Based on system (4.2.1) we compute the corrector direction $(\Delta^c\mathbf{x}, \Delta^c\mathbf{s})$ from system:

$$\begin{aligned} -M\Delta^c\mathbf{x} + \Delta^c\mathbf{s} &= 0, \\ \mathbf{s}^p\Delta^c\mathbf{x} + \mathbf{x}^p\Delta^c\mathbf{s} &= 2\left(\sqrt{\mu_p\,\mathbf{x}^p\,\mathbf{s}^p} - \mathbf{x}^p\,\mathbf{s}^p\right), \end{aligned} \tag{4.2.25}$$

where

$$\mu_p = \mu(\mathbf{x}^p, \mathbf{s}^p) = \frac{(\mathbf{x}^p)^T\mathbf{s}^p}{n}. \tag{4.2.26}$$

We have to calculate the corrector step length as the corrector step reaches the narrower, corrector neighborhood of the central path. For reaching this neighborhood of the central path, we have to solve the following optimization problem

$$\alpha_c := \arg\min\{\mu_c(\alpha) : (\mathbf{x}^c(\alpha), \mathbf{s}^c(\alpha)) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta) = \mathcal{D}_\varphi(\beta)\}, \tag{4.2.27}$$

where

$$\mu_c(\alpha) := \mu(\mathbf{x}^c(\alpha), \mathbf{s}^c(\alpha)) = \frac{(\mathbf{x}^c(\alpha))^T\mathbf{s}^c(\alpha)}{n} \tag{4.2.28}$$

and

$$\mathbf{x}^c(\alpha) := \mathbf{x}^p + \alpha\Delta^c\mathbf{x}, \quad \mathbf{s}^c(\alpha) := \mathbf{s}^p + \alpha\Delta^c\mathbf{s}. \tag{4.2.29}$$

However, it is enough to find such a step length $\bar{\alpha}$ which ensures that in one step we get back into the corrector neighborhood and gives enough decrease in the duality gap and that is

$$\bar{\alpha} := \frac{\beta}{2\left((1+4\kappa)\,n+1\right)}. \tag{4.2.30}$$

Algorithm 4 Predictor-corrector interior-point algorithm acting in a wide neighborhood of the central path [57]

Input:

Given $\kappa \geq \hat{\kappa}(M)$, $(\mathbf{x}^0, \mathbf{s}^0) \in \mathcal{N}^{-}_{\infty,\varphi}(1-\beta)$, $\beta \in (0,1)$

Calculate $\gamma = \frac{1-\beta}{5((1+4\kappa)n+1)}$

Let $\mu_0 = \mu(\mathbf{x}^0, \mathbf{s}^0)$ and $k = 0$

$\varepsilon > 0$ precision value.

While $n\mu \geq \varepsilon$

(Predictor step);

$\mathbf{x} := \mathbf{x}^k$, $\mathbf{s} := \mathbf{s}^k$;

Step 1. Calculate the affine direction from (4.2.2);

Step 2. Calculate the predictor steplength using (4.2.18);

Step 3. Calculate $(\mathbf{x}^p, \mathbf{s}^p)$ using (4.2.24);

If $\mu(\mathbf{x}^p, \mathbf{s}^p) = 0$

STOP; Optimal solution found;

Else if $(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{N}^{-}_{\infty,\varphi}(1-\beta)$

$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^p, \mathbf{s}^p)$, $\mu^{k+1} = \mu(\mathbf{x}^p, \mathbf{s}^p)$, $k = k+1$, RETURN;

(Corrector step);

Step 4. Calculate centering direction from (4.2.25);

Step 5. Calculate centering steplength using (4.2.27);

Step 6. Calculate $(\mathbf{x}^c, \mathbf{s}^c)$ using (4.2.39);

$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^c, \mathbf{s}^c)$, $\mu^{k+1} = \mu(\mathbf{x}^c, \mathbf{s}^c)$, $k = k+1$, RETURN;

End

Output: $(\mathbf{x}^k, \mathbf{s}^k) : (\mathbf{x}^k)^T \mathbf{s}^k \leq \varepsilon$

Now, as presented in [57], we show a procedure to compute the corrector step length α_c. Considering (4.2.25) and (4.2.28) we obtain

$$\mathbf{x}^c(\alpha)\mathbf{s}^c(\alpha) = (1-2\alpha)\mathbf{x}^p\mathbf{s}^p + 2\alpha\sqrt{\mu_p \mathbf{x}^p\mathbf{s}^p} + \alpha^2 \Delta^c\mathbf{x}\Delta^c\mathbf{s}. \tag{4.2.31}$$

Moreover, based on the Cauchy-Schwartz inequality we derive from $\mathbf{e}^T\sqrt{\mathbf{x}^p\mathbf{s}^p} \leq \sqrt{n}\sqrt{\mathbf{e}^T\mathbf{x}^p\mathbf{s}^p}$. Now using (4.2.29), we have

$$\begin{aligned} \mu_c(\alpha) &= \frac{\mathbf{x}^c(\alpha)^T\mathbf{s}^c(\alpha)}{n} = (1-2\alpha)\mu_p + 2\alpha\frac{\mathbf{e}^T\sqrt{\mu_p\mathbf{x}^p\mathbf{s}^p}}{n} + \frac{\alpha^2\Delta^c\mathbf{x}^T\Delta^c\mathbf{s}}{n} \\ &\leq (1-2\alpha)\mu_p + 2\alpha\frac{\sqrt{\mu_p}}{n}\sqrt{n}\sqrt{\mathbf{e}^T\mathbf{x}^p\mathbf{s}^p} + \frac{\alpha^2\Delta^c\mathbf{x}^T\Delta^c\mathbf{s}}{n} \\ &= (1-2\alpha)\mu_p + 2\alpha\mu_p + \frac{\alpha^2\Delta^c\mathbf{x}^T\Delta^c\mathbf{s}}{n} = \mu_p + \frac{\alpha^2\Delta^c\mathbf{x}^T\Delta^c\mathbf{s}}{n} =: \bar{\mu}_c(\alpha), \end{aligned} \tag{4.2.32}$$

as well.

From (4.2.5) and (4.2.26) we have

$$\bar{\mathbf{u}} = \frac{\mathbf{x}^p \mathbf{s}^p}{\mu_p}, \qquad \bar{\mathbf{v}} = \frac{\Delta^c \mathbf{x} \Delta^c \mathbf{s}}{\mu_p}. \tag{4.2.33}$$

Our goal is to satisfy the following:

$$\sqrt{\frac{\mathbf{x}^c(\alpha)\mathbf{s}^c(\alpha)}{\mu_c(\alpha)}} \geq \beta\,\mathbf{e}. \tag{4.2.34}$$

This can be written as

$$(\mathbf{x}^p + \alpha\Delta^c\mathbf{x})\,(\mathbf{s}^p + \alpha\Delta^c\mathbf{s}) \geq \mu_c(\alpha)\,\beta^2\,\mathbf{e}. \tag{4.2.35}$$

Considering (4.2.32) we get

$$\mu_c(\alpha) \leq \bar{\mu}_c(\alpha). \tag{4.2.36}$$

This shows that we need to satisfy only the following

$$(\mathbf{x}^p + \alpha\Delta^c\mathbf{x})\,(\mathbf{s}^p + \alpha\Delta^c\mathbf{s}) \geq \bar{\mu}_c(\alpha)\beta^2\mathbf{e} \geq \mu_c(\alpha)\beta^2\mathbf{e}. \tag{4.2.37}$$

We have to fulfill $(\mathbf{x}^c(\alpha), \mathbf{s}^c(\alpha)) \in \mathcal{F}^+$ as well. Using this we have $\alpha_F = \min\{\alpha_x, \alpha_s\}$, where

$$\alpha_x = \min\left\{-\frac{x_i^p}{\Delta^c x_i} : \Delta^c x_i < 0\right\} \quad \text{and} \quad \alpha_s = \min\left\{-\frac{s_i^p}{\Delta^c s_i} : \Delta^c s_i < 0\right\}.$$

This shows that similarly to the predictor step, we have that $\alpha_F > 0$ is the largest step for which the iterates after a corrector step are still feasible. Furthermore, it can be seen that any α that satisfies (4.2.37) gives a lower bound on α_F. Using (4.2.37), after some calculations, we get the following:

$$\hat{a}_i\,\alpha^2 + \hat{b}_i\,\alpha + \hat{c}_i \geq 0, \qquad \text{for all } \; i, \tag{4.2.38}$$

where $\hat{a}_i = \bar{v}_i - \beta^2\,\frac{\mathbf{e}^T\bar{\mathbf{v}}}{n}$, $\hat{b}_i = 2\left(\sqrt{\bar{u}_i} - \bar{u}_i\right)$ and $\hat{c}_i = \bar{u}_i - \beta^2$. The discriminant of the quadratic equation is $\Delta_i = \hat{b}_i^2 - 4\,\hat{a}_i\,\hat{c}_i$.

We have the smallest and the largest root of the quadratic equation, if $\Delta_i \geq 0$ and $\hat{a}_i \neq 0$ are satisfied, as

$$\alpha_i^- = \frac{-\hat{b}_i - \operatorname{sgn}(\hat{a}_i)\sqrt{\Delta_i}}{2\,\hat{a}_i}, \qquad \text{and} \qquad \alpha_i^+ = \frac{-\hat{b}_i + \operatorname{sgn}(\hat{a}_i)\sqrt{\Delta_i}}{2\,\hat{a}_i}.$$

Let us denote the solution set of the inequality (4.2.38) as T_i for each $1,\dots,n$. The T_i is given below

$$\mathcal{T}_i = \begin{cases} (-\infty,\infty), & \text{if } \Delta_i<0, \hat{a}_i>0 \\ (-\infty,\alpha_i^-]\cup[\alpha_i^+,\infty), & \text{if } \Delta_i\geq 0, \hat{a}_i>0 \\ [\alpha_i^-,\alpha_i^+], & \text{if } \Delta_i\geq 0, \hat{a}_i<0 \\ \left(-\infty,-\frac{\hat{c}_i}{\hat{b}_i}\right], & \text{if } \hat{a}_i=0, \hat{b}_i<0 \\ \left[-\frac{\hat{c}_i}{\hat{b}_i},\infty\right), & \text{if } \hat{a}_i=0, \hat{b}_i>0 \\ (-\infty,\infty), & \text{if } \hat{a}_i=0, \hat{b}_i=0. \end{cases}$$

It is important to note that the following holds

$$-\frac{\hat{c}_i}{\hat{b}_i} = \frac{\beta^2-\bar{u}_i}{2\left(\sqrt{\bar{u}_i}-\bar{u}_i\right)}.$$

It can be seen that the inequality (4.2.34) holds $\forall\alpha\in\mathcal{T}=\cap_{i=1}^n\mathcal{T}_i\cap\mathbb{R}^n_\oplus$, except for $\mathcal{T}=\emptyset$. We will show in Theorem 4.3.1 that $\mathcal{T}\neq\emptyset$.

After each corrector step we have the new points as

$$(\mathbf{x}^c,\mathbf{s}^c) = (\mathbf{x}^c(\alpha_c),\mathbf{s}^c(\alpha_c)) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta) = \mathcal{D}_\varphi(\beta), \tag{4.2.39}$$

where $\mathbf{x}^c(\alpha_c) = \mathbf{x}^p+\alpha_c\Delta^c\mathbf{x}$ and $\mathbf{s}^c(\alpha_c) = \mathbf{s}^p+\alpha_c\Delta^c\mathbf{s}$.

This shows that we have $(\mathbf{x}^c,\mathbf{s}^c)\in\mathcal{N}^-_{\infty,\varphi}(1-\beta)=\mathcal{D}_\varphi(\beta)$. Hence, we set $(\mathbf{x},\mathbf{s}):=(\mathbf{x}^c,\mathbf{s}^c)$ and start another predictor-corrector iteration.

Since, it is computationally very expensive to calculate the exact value of α_c, we use another $\bar{\alpha}\in\mathcal{T}$ such that $\mu_c(\alpha_c)\leq\mu_c(\bar{\alpha})$. We show in Theorem 4.3.1 that $\bar{\alpha}$ guarantees a large enough decrease in the duality gap. It is important to mention that it is not necessary to solve either optimization problem given in (4.2.20) or in (4.2.27). However, by solving those optimization problems we can ensure that our step lenght does not depend on the value κ.

Finally, in the following subsection we show the analysis of Algorithm 4.

4.3 Analysis of the algorithm

In Section 4.2 we proved the feasibility of the predictor and corrector step. In Theorem 5.1. in [57] we gave the size of the decrease of the central path parameter at each iteration.

Theorem 4.3.1. *(Theorem 5.1. in [57]) Let $n \geq 2$ and $\beta \in (0,1)$. Then, the PC IPA given in Algorithm 4 using the function $\varphi(t) = \sqrt{t}$ in the AET technique is well defined and*

$$\mu_{k+1} \leq \left(1 - \frac{(1-\beta)\beta}{20((1+4\kappa)n+2)}\right)\mu_k, \quad k = 0,1\dots$$

Proof. As an input for the k^{th} iteration of Algorithm 4 we have $(\mathbf{x}^k, \mathbf{s}^k) := (\mathbf{x}, \mathbf{s}) \in \mathcal{D}_\varphi(\beta)$ and $\mu_k := \mu(\mathbf{x}, \mathbf{s})$. After we get the Newton-directions from system (4.2.2) and calculate step length α_p, we get $(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{D}_\varphi((1-\gamma)\beta)$ by using Lemma 4.2.3. Moreover, we get a lower bound on α_p by using Lemma 4.2.5, hence, we obtain

$$\sqrt{\mathbf{x}^p(\alpha)\mathbf{s}^p(\alpha)} \geq (1-\gamma)\beta\sqrt{\mu_p(\alpha)} > (1-\gamma)\beta\sqrt{\mu_p(\alpha_p)} \geq 0,$$

where $\alpha_p > \hat{\alpha}$. Based on (4.2.14) we get $\frac{\mathbf{e}^T\mathbf{v}}{n} \leq 1$, which means that we have $(\Delta^p\mathbf{x})^T\Delta^p\mathbf{s} \leq \mu(\mathbf{x},\mathbf{s})n$. Considering (4.2.6) if $\alpha_p > \hat{\alpha}$, the following holds:

$$\mu_p = \mu(\alpha_p) < \mu(\hat{\alpha}) \leq \left((1-2\hat{\alpha}) + \hat{\alpha}^2\right)\mu(\mathbf{x},\mathbf{s}) = (1-(2-\hat{\alpha})\hat{\alpha})\,\mu(\mathbf{x},\mathbf{s}). \tag{4.3.1}$$

Let us assume that $n \geq 2$, $\kappa > 0$ and from (4.2.22), we derive

$$2 - \hat{\alpha} = 2 - \frac{\beta\sqrt{1-\beta}}{5((1+4\kappa)n+2)} \geq 2 - \frac{\beta\sqrt{1-\beta}}{20} \geq 2 - \frac{1}{40} = \frac{79}{40}.$$

Therefore, we obtain

$$\mu_p \leq \left(1 - \frac{79\beta\sqrt{1-\beta}}{200((1+4\kappa)n+2)}\right)\mu(\mathbf{x},\mathbf{s}). \tag{4.3.2}$$

Let us analyse the corrector step. First, we have $(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta+\beta\gamma) = \mathcal{D}_\varphi((1-\gamma)\beta)$, hence,

$$\sqrt{\bar{\mathbf{u}}} = \sqrt{\frac{\mathbf{x}^p\mathbf{s}^p}{\mu_p}} \geq (1-\gamma)\,\beta\,\mathbf{e} \qquad \Longleftrightarrow \qquad \mathbf{e}^T\sqrt{\bar{\mathbf{u}}} \geq (1-\gamma)\,\beta\, n.$$

Considering Lemma 4.2.1 with $\mathbf{a} = 2(\sqrt{\mu_p \mathbf{x}^p \mathbf{s}^p} - \mathbf{x}^p \mathbf{s}^p)$ we obtain

$$\frac{1}{1+4\kappa} \|\Delta^c \mathbf{x} \Delta^c \mathbf{s}\|_\infty \leq \sum_{i \in \mathcal{I}_+} \Delta^c x_i \Delta^c s_i \leq \mu_p \left\| \mathbf{e} - \sqrt{\bar{\mathbf{u}}} \right\|_2^2. \tag{4.3.3}$$

Moreover, we have

$$\left\| \mathbf{e} - \sqrt{\bar{\mathbf{u}}} \right\|_2^2 = (\mathbf{e} - \sqrt{\bar{\mathbf{u}}})^T (\mathbf{e} - \sqrt{\bar{\mathbf{u}}}) = (n - 2\mathbf{e}^T \sqrt{\bar{\mathbf{u}}} + \mathbf{e}^T \bar{\mathbf{u}}) \leq 2(1 - (1-\gamma)\beta)\, n := \bar{\xi} n,$$

where

$$\bar{\xi} = 2(1 - (1-\gamma)\beta). \tag{4.3.4}$$

Based on (4.3.3) and (4.3.4) we obtain

$$\|\Delta^c \mathbf{x} \Delta^c \mathbf{s}\|_\infty \leq (1+4\kappa)\, \bar{\xi}\, n\, \mu_p, \qquad \sum_{i \in \mathcal{I}_+} \Delta^c x_i \Delta^c s_i \leq \bar{\xi}\, n\, \mu_p. \tag{4.3.5}$$

This shows that $\|\bar{\mathbf{v}}\|_\infty \leq (1+4\kappa)\bar{\xi} n$. Considering (4.2.31) we get

$$\begin{aligned}
\frac{\mathbf{x}^c(\alpha)\mathbf{s}^c(\alpha)}{\mu_p} &= (1-2\alpha)\,\bar{\mathbf{u}} + 2\alpha\sqrt{\bar{\mathbf{u}}} + \alpha^2\,\bar{\mathbf{v}} \\
&\geq \left((1-2\alpha)\,((1-\gamma)\beta)^2 + 2\alpha\,(1-\gamma)\,\beta - \alpha^2\,(1+4\kappa)\,\bar{\xi} n \right) \mathbf{e} \\
&= \left(((1-\gamma)\beta)^2 + \alpha\,(1-\gamma)\,\beta\,\bar{\xi} - \alpha^2\,(1+4\kappa)\,\bar{\xi} n \right) \mathbf{e}.
\end{aligned} \tag{4.3.6}$$

By using (4.2.32) and (4.3.5) we get

$$\mu_c(\alpha) \leq \left(1 + \alpha^2 \frac{\mathbf{e}^T \bar{\mathbf{v}}}{n} \right) \mu_p \leq \left(1 + \alpha^2\, \bar{\xi} \right) \mu_p. \tag{4.3.7}$$

From (4.3.6) and (4.3.7) we obtain

$$\frac{\mathbf{x}^c(\alpha)\mathbf{s}^c(\alpha) - \beta^2\,\mu_c(\alpha)\,\mathbf{e}}{\mu_p} \geq \hat{g}(\alpha)\,\mathbf{e}, \tag{4.3.8}$$

where

$$\hat{g}(\alpha) := -\gamma\,\beta^2\,(2-\gamma) + \alpha\,\bar{\xi}\,((1-\gamma)\,\beta) - \bar{\xi}\,\left(\beta^2 + (1+4\kappa)\,n \right) \alpha^2. \tag{4.3.9}$$

It can be seen that (4.3.9) is equivalent to $\hat{g}(\alpha) = a_{\hat{g}}\,\alpha^2 + b_{\hat{g}}\,\alpha + c_{\hat{g}}$, where

$$a_{\hat{g}} := -\bar{\xi}\left(\beta^2 + (1+4\kappa)\,n\right) < 0, \qquad b_{\hat{g}} := \bar{\xi}\,((1-\gamma)\,\beta) > 0,$$

and

$$c_{\hat{g}} := -\gamma\,\beta^2\,(2-\gamma) < 0.$$

Based on (4.2.7) we have

$$\bar{\xi} = 2\,\frac{(1-\beta)\,(\beta + 5((1+4\kappa)n+1))}{5((1+4\kappa)n+1)}.$$

Furthermore, from (4.2.7) and using the definition of $\bar{\xi}$, the parameters $a_{\hat{g}}, b_{\hat{g}}$ and $c_{\hat{g}}$ can be calculated as

$$c_{\hat{g}} = -\frac{\beta^2\,(1-\beta)\,(10\,((1+4\,\kappa)\,n+1)+\beta-1)}{25((1+4\,\kappa)\,n+1)^2},$$

$$b_{\hat{g}} = \frac{2\,\beta\,(1-\beta)\,(\beta+5((1+4\,\kappa)\,n+1))\,(\beta-1+5((1+4\,\kappa)\,n+1))}{25((1+4\,\kappa)\,n+1)^2}$$

and

$$a_{\hat{g}} = -\frac{2\,(1-\beta)\,(\beta+5((1+4\,\kappa)\,n+1))\,(\beta^2+(1+4\,\kappa)\,n)}{5((1+4\,\kappa)\,n+1)}.$$

It can be seen that our goal is to find those parameters β, γ for which $\hat{g}(\alpha) \geq 0$. Let us simplify the search by fixing

$$\bar{\alpha} := \frac{\beta}{2\,((1+4\,\kappa)\,n+1)} \tag{4.3.10}$$

and with that we calculate the parameter β. It is easy to see that $\hat{g}(\bar{\alpha}) \geq 0$ is equivalent to

$$\hat{g}(\bar{\alpha}) := \frac{\beta^2\,(1-\beta)}{((1+4\,\kappa)\,n+1)^3}\,\bar{g}(\bar{\alpha}) \geq 0$$

for some $\beta \in (0,1)$. We get the following:

$$\begin{aligned}
\bar{g}(\bar{\alpha}) \;&=\; -\frac{1}{10}\,(\beta+5((1+4\,\kappa)\,n+1))\,(\beta^2+(1+4\,\kappa)\,n) \\
&\quad+\; \frac{1}{25}(\beta+5((1+4\,\kappa)\,n+1))\,(\beta-1+5((1+4\,\kappa)\,n+1)) \\
&\quad-\; \frac{1}{25}\,(\beta-1+10\,((1+4\,\kappa)\,n+1))\,((1+4\,\kappa)\,n+1)\,.
\end{aligned}$$

After some calculations, we obtain

$$\hat{f}(\beta) := \bar{g}(\bar{\alpha}) = \left(-\frac{\beta^3}{10} - \frac{23\beta^2}{50} + \frac{8\beta}{25} + \frac{11}{25}\right) + \left(-\frac{\beta^2}{2} + \frac{13\beta}{50} + \frac{27}{50}\right)(1+4\kappa)n + \frac{1}{10}(1+4\kappa)^2 n^2.$$

We would like to find all $\beta \in (0,1)$ that satisfy $\hat{f}(\beta) \geq 0$. We have to solve the following system of nonlinear inequalities to have a sufficient condition for $\hat{f}(\beta) \geq 0$:

$$\begin{aligned} -\frac{\beta^3}{10} - \frac{23\beta^2}{50} + \frac{8\beta}{25} + \frac{11}{25} &\geq 0, \\ -\frac{\beta^2}{2} + \frac{13\beta}{50} + \frac{27}{50} &\geq 0. \end{aligned}$$

It can be seen after some calculations that both of the inequalities hold for $\beta \in (0,1)$. This shows that the inequality $\hat{f}(\beta) \geq 0$ holds for $\beta \in (0,1)$. Moreover, this shows as well that $\bar{\alpha} \in \mathcal{T}$, hence $\mathcal{T} \neq \emptyset$. Therefore, the analysis of the corrector step is completed.

Finally, let us consider the case where $\bar{\alpha}$ is given, $\beta \in (0,1)$ and $n \geq 2$. Moreover, let us consider that at the beginning of the kth iteration we have $(\mathbf{x}^k, \mathbf{s}^k) \in \mathcal{D}_{\varphi}(\beta)$ with μ_k, and we calculate the new iterate $(\mathbf{x}^p, \mathbf{s}^p)$ with μ_p in the predictor step.

Let us assume that $n \geq 2$ and by using (4.3.7) we get

$$\begin{aligned} \mu_c &= \mu_c(\alpha_c) \leq \mu_c(\bar{\alpha}) = \mu_c\left(\frac{\beta}{2((1+4\kappa)n+1)}\right) \\ &\leq \left(1 + \frac{\beta^2(1-\beta)(5((1+4\kappa)n+1)+\beta)}{10((1+4\kappa)n+1)^3}\right)\mu_p. \end{aligned}$$

We have $\frac{5((1+4\kappa)n+1)+\beta}{10((1+4\kappa)n+1)} = \frac{1}{2} + \frac{\beta}{10((1+4\kappa)n+1)} \leq \frac{2}{3}$, hence, we obtain

$$\mu_c \leq \left(1 + \frac{2\beta^2(1-\beta)}{3((1+4\kappa)n+1)^2}\right)\mu_p < \left(1 + \frac{2\beta(1-\beta)}{3((1+4\kappa)n+1)^2}\right)\mu_p. \tag{4.3.11}$$

From (4.3.2) and (4.3.11) and using that $\frac{79}{200} - \frac{2}{3(1+4\kappa)n} \geq \frac{37}{600} > \frac{1}{20}$ holds, where $n \geq 2$, we get

$$
\begin{aligned}
\mu_c &\leq \left(1-\frac{79\beta\sqrt{1-\beta}}{200((1+4\kappa)n+2)}\right)\left(1+\frac{2\beta(1-\beta)}{3((1+4\kappa)n+1)^2}\right)\mu \\
&\leq \left(1-\frac{79\beta(1-\beta)}{200((1+4\kappa)n+2)}\right)\left(1+\frac{2\beta(1-\beta)}{3(1+4\kappa)n((1+4\kappa)n+2)}\right)\mu \\
&\leq \left(1-\frac{79\beta(1-\beta)}{200((1+4\kappa)n+2)}+\frac{2\beta(1-\beta)}{3(1+4\kappa)n((1+4\kappa)n+2)}\right)\mu \\
&\leq \left(1-\left(\frac{79}{200}-\frac{2}{3(1+4\kappa)n}\right)\frac{\beta(1-\beta)}{((1+4\kappa)n+2)}\right)\mu \\
&\leq \left(1-\frac{(1-\beta)\beta}{20((1+4\kappa)n+2)}\right)\mu.
\end{aligned} \tag{4.3.12}
$$

Finally, if we denote by $\mu = \mu(\mathbf{x}^k,\mathbf{s}^k) = \mu_k$ and $\mu_{k+1} = \mu_c$, we get the desired result.

□

Finally, we showed in Corollary 5.2. in [57] a direct consequence of Theorem 4.3.1.

Corollary 4.3.1. *(Corollary 5.2. in [57]) Let $n \geq 2$ and $\beta \in (0,1)$. Then, Algorithm 4 produces an iterate $(\mathbf{x}^k,\mathbf{s}^k) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$ with $\mathbf{x}^k\mathbf{s}^k \leq \epsilon$ in at most*

$$
\mathcal{O}\left((1+\kappa)\,n\log\left(\frac{\left(\mathbf{x}^0\right)^T\mathbf{s}^0}{\epsilon}\right)\right)
$$

iterations.

It is important to note that (4.2.7) depends on the handicap parameter $\kappa \geq \hat{\kappa}(M)$. It is computationally expensive to determine an upper bound for the handicap $\hat{\kappa}(M)$ in case of many applications, see [33, 81, 102]. In Section 4.4 we present our PC IPA that works in a wide neighborhood of the central path and does not depend on the handicap $\hat{\kappa}(M)$ of the matrix.

4.4 Modification of the predictor-corrector interior-point algorithm

First, it is important to mention that, in this case, if Algorithm 4 can not produce corrector points $(\mathbf{x}^c,\mathbf{s}^c)$ in $(k+1)$th iteration such that $(\mathbf{x}^c,\mathbf{s}^c) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta) = \mathcal{D}_\varphi(\beta)$ with $\varphi(t) = \sqrt{t}$, then we double the value of κ and restart Algorithm 4 from points $(\mathbf{x}^k,\mathbf{s}^k) \in \mathcal{D}_\varphi(\beta)$. We choose to do this because it is most likely that the value κ is too small. Finally, we note that we have to double the value of κ at most $\lceil \log_2 \hat{\kappa}(M) \rceil$ times. In Algorithm 5 a modified PC IPA is presented.

Algorithm 5 Modified predictor-corrector interior-point algorithm [57]

Input:
$(\mathbf{x}^0, \mathbf{s}^0) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$, $\beta \in (0,1)$;
Set $\kappa = 1$
Let $\mu_0 = \mu(\mathbf{x}^0, \mathbf{s}^0)$ and $k = 0$
$\varepsilon > 0$ precision value.
While $n\mu \geq \varepsilon$
(Predictor step);
$\mathbf{x} := \mathbf{x}^k$, $\mathbf{s} := \mathbf{s}^k$;
Step 1. Calculate affine direction from (4.2.2);
Step 2. Calculate the predictor steplength using (4.2.18);
Step 3. Calculate $(\mathbf{x}^p, \mathbf{s}^p)$;
If $\mu(\mathbf{x}^p, \mathbf{s}^p) = 0$
STOP; Optimal solution found;
Else if $(\mathbf{x}^p, \mathbf{s}^p) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$
$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^p, \mathbf{s}^p)$, $\mu^{k+1} = \mu(\mathbf{x}^p, \mathbf{s}^p)$, $k = k+1$, RETURN;
(Corrector step);
Step 4. Calculate centering direction from (4.2.25);
Step 5. Calculate centering steplength using (4.2.27);
Step 6. Calculate $(\mathbf{x}^c, \mathbf{s}^c)$;
If $(\mathbf{x}^c, \mathbf{s}^c) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$
$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^c, \mathbf{s}^c)$, $\mu^{k+1} = \mu(\mathbf{x}^c, \mathbf{s}^c)$, $k = k+1$, RETURN;
Else
$\kappa = 2\kappa$; $(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^k, \mathbf{s}^k)$, $\mu^{k+1} = \mu(\mathbf{x}^k, \mathbf{s}^k)$, $k = k+1$, RETURN;
End
Output: $(\mathbf{x}^k, \mathbf{s}^k) : \mathbf{x}^{kT}\mathbf{s}^k \leq \varepsilon$

In the following theorem the complexity result is provided for Algorithm 5.

Theorem 4.4.1. *(Lemma 6.1. in [57]) Algorithm 5 produces an iterate* $(\mathbf{x}^k, \mathbf{s}^k) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$ *with* $\mathbf{x}^k\mathbf{s}^k \leq \epsilon$ *in at most* $\mathcal{O}\left((1+\hat{\kappa}(M))\, n \log\left(\frac{(\mathbf{x}^0)^T\mathbf{s}^0}{\epsilon}\right)\right)$ *iterations.*

Proof. Let $\bar{\kappa}$ be the largest value of κ used in Algorithm 5. Then, it can be seen that $\bar{\kappa} < 2\hat{\kappa}(M)$. Let us take into account that at iteration k of Algorithm 5 we get $\kappa < \hat{\kappa}(M)$. If $(\mathbf{x}^c, \mathbf{s}^c) \in \mathcal{N}^-_{\infty,\varphi}(1-\beta)$, then $(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^c, \mathbf{s}^c)$. Considering that in case of $\kappa = \hat{\kappa}(M)$, Lemmas 4.2.1 and 4.2.2 are fulfilled and that the bound on the predictor step size is based on the parameter γ that is decreasing in κ, we have

$$\mu_{k+1} \leq \left(1 - \frac{(1-\beta)\beta}{20((1+4\hat{\kappa}(M))n+2)}\right)\mu_k \leq \left(1 - \frac{(1-\beta)\beta}{20((1+8\hat{\kappa}(M))n+2)}\right)\mu_k.$$

Moreover, if $\kappa \geq \hat{\kappa}(M)$, then the corrector step is never rejected. This means that based on Theorem 4.3.1 and the fact that $\kappa \leq \bar{\kappa} < 2\hat{\kappa}(M)$, we get

$$\mu_{k+1} \leq \left(1 - \frac{(1-\beta)\beta}{20((1+4\kappa)n+2)}\right)\mu_k \leq \left(1 - \frac{(1-\beta)\beta}{20((1+8\hat{\kappa}(M))n+2)}\right)\mu_k.$$

By taking into account that at most $\log_2(\bar{\kappa})$ rejections can occur, the lemma is proven. □

Finally, in the next chapter we present the numerical results obtained by using the new PD and PC IPAs presented in this thesis.

Chapter 5

Numerical results

In this chapter, we present a solver for solving sufficient LCPs. We developed the implemented versions of Algorithms 1-4. We will show the differences between the theoretical and implemented versions of the PD and PC IPAs. All computations were made on a desktop computer with an Intel quad-core 3.3 GHz processor and 32 GB RAM.

First, we present the method we used to determine the search directions in the implementation. It is important to mention that the calculation of the Newton directions, the step length, the verification of feasibility, and even the neighborhoods are different in the implementation than in the theory.

5.1 Calculating Newton directions

In this section, the calculation of the Newton directions is presented. First, it is shown how the Newton directions are calculated in theory. After that, their practical calculation method is presented. Throughout this section, let M' be a general $2n \times 2n$ matrix and $\mathbf{x}'$ be an arbitrary vector.

In theory, from Newton system (1.4.2), we get the Newton directions as

$$\begin{aligned} \Delta\mathbf{x} &= (S+XM)^{-1}\mathbf{a}_{\varphi}, \\ \Delta\mathbf{s} &= M(S+XM)^{-1}\mathbf{a}_{\varphi}, \end{aligned} \tag{5.1.1}$$

where $\mathbf{a}_{\varphi}$ is given in (1.4.3). In practice, it is very expensive to calculate the inverse of a matrix. Furthermore, it is unstable, as numerical problems frequently occur when the matrix is nearly singular.

Considering the following system,

$$M'\mathbf{x}' = \mathbf{a}',$$

where $\mathbf{a}'$ is arbitrary, one of the easiest ways to solve this problem is to decompose the matrix into a lower (L) and an upper (U) triangular matrix. In this case, we have to solve

$$M'\mathbf{x}' = (L \cdot U)\mathbf{x}' = \mathbf{a}'. \tag{5.1.2}$$

Using the notation $\mathbf{y}' = U\mathbf{x}'$, we have

$$L\mathbf{y}' = \mathbf{a}', \tag{5.1.3}$$

and

$$U\mathbf{x}' = \mathbf{y}'. \tag{5.1.4}$$

From system (5.1.3) and (5.1.4) we obtain $\mathbf{x}'$.

Several techniques are known for the decomposition of a matrix. It is well known that for positive semidefinite matrices, the best decomposition method is the Cholesky-decomposition. However, for solving LCPs, Cholesky-decomposition is out of question, since the matrices appearing in LCPs are more general. In the implementation, we tried two of these types of decompositions. We used LU and QR decompositions in our solver.

5.1.1 Algebraically equivalent Newton systems

Note that (1.4.2) can be written as

$$\begin{bmatrix} -M & I \\ S & X \end{bmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_\varphi \end{pmatrix}. \tag{5.1.5}$$

Considering (5.1.2), we have

$$M'\mathbf{x}' = LU\mathbf{x}' = \mathbf{q}',$$

where $M' = \begin{bmatrix} -M & I \\ S & X \end{bmatrix}$, $\mathbf{q}' = \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_\varphi \end{pmatrix}$ and $\mathbf{x}' = \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \end{pmatrix}$. Using LU or QR decomposition, we have the desired result.

Expressing $\Delta\mathbf{s}$ from the first equation of (1.4.2) we obtain

$$\Delta\mathbf{s} = M\Delta\mathbf{x}, \qquad S\Delta\mathbf{x} + X\Delta\mathbf{s} = \mathbf{a}_\varphi.$$

From this we have

$$(S + XM)\Delta\mathbf{x} = \mathbf{a}_\varphi. \tag{5.1.6}$$

Using (5.1.2), we have

$$M'\mathbf{x}' = LU\mathbf{x}' = \mathbf{q}',$$

where $M' = (S + XM)$, $\mathbf{q}' = \mathbf{a}_\varphi$ and $\mathbf{x}' = \Delta\mathbf{x}$. With LU or QR decomposition we get $\Delta\mathbf{x}$. Finally, we obtain $\Delta\mathbf{s}$ by $\Delta\mathbf{s} = M\Delta\mathbf{x}$.

Multiplying (5.1.6) by X^{-1} from the right, we obtain

$$(X^{-1}S + M)\Delta\mathbf{x} = X^{-1}\mathbf{a}_\varphi. \tag{5.1.7}$$

It is important to mention that in this case, after some iterations we do not multiply the matrix M by small numbers that came from the matrix X.
Considering (5.1.2), we have

$$M'\mathbf{x}' = LU\mathbf{x}' = \mathbf{q}',$$

where $M' = (X^{-1}S + M)$, $\mathbf{q}' = X^{-1}\mathbf{a}_\varphi$ and $\mathbf{x}' = \Delta\mathbf{x}$. We get $\Delta\mathbf{x}$ by using LU or QR decomposition. Finally, we get $\Delta\mathbf{s}$ from $\Delta\mathbf{s} = M\Delta\mathbf{x}$.
We used system (5.1.5) for solving LCPs, but in some cases (5.1.7) showed better results. Because of this, in the general case we solved the system which has $X^{-1}S + M$ as coefficient matrix.

In the case of the kernel functions, we worked with the scaled Newton systems, so we mention the implementation of that case as well in this section.
The scaled $2n \times 2n$ system is based on (1.4.5) and it is the following:

$$\begin{bmatrix} -\bar{M} & I \\ I & I \end{bmatrix} \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_s \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{p}_\varphi \end{pmatrix}, \tag{5.1.8}$$

where $\bar{M}$ is given in (1.4.7), $\mathbf{d}_x, \mathbf{d}_s$ in (1.4.5) and $\mathbf{p}_\varphi$ in (1.4.8). If we consider (5.1.2), we have

$$M'\mathbf{x}' = LU\mathbf{x}' = \mathbf{q}',$$

where $M' = \begin{bmatrix} -\bar{M} & I \\ I & I \end{bmatrix}$, $\mathbf{q}' = \begin{pmatrix} \mathbf{0} \\ \mathbf{p}_\varphi \end{pmatrix}$ and $\mathbf{x}' = \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_s \end{pmatrix}$. Using LU or QR decomposition, we get $\begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_s \end{pmatrix}$, and then it is an easy computation to get $\Delta\mathbf{x}$ and $\Delta\mathbf{s}$.

In the case of PD IPAs, we solve one of the four Newton systems (5.1.5), (5.1.6), (5.1.7) or (5.1.8). After calculating the Newton directions we use them for calculating the new points in each iteration. We calculate the Newton directions similarly in the case of PC IPAs, but for both of the predictor and the corrector steps in each iteration. For the predictor step, we set $\mu = 0$ in the corresponding Newton system.

We used the ALGLIB library of C++ for getting solutions of these systems. The functions we used from this library are rmatrixlu(), rmatrixqr() and rmatrixsolve(). We chose to use this library because it calculates LU, QR and solves linear equation systems really fast in large cases as well. Furthermore, we tested its precision with Frobenius-norm and maximum norm as well, and it had a 10^{-17} precision in both cases.

5.2 Step length

In this section, two ways of calculation of the step length are discussed.
It is important to mention that in theory, the step length is $\alpha = 1$ in case of Algorithm 1 and 2. However, in practice we calculate the step length $\alpha \in (0,1)$ in a different way. Note that in the case of LCPs the calculation of the handicap might be exponential, and we will show that there is a strong connection between the value of κ and the step length. The detailed explanation can be found in Paragraph 5.5.1.1.1.

In practice we have $\alpha < 1$ in most of the cases. We calculate α by using ratio test in practice in the following way:

$$\alpha_x = \max_{\Delta\mathbf{x}_i < 0} -\frac{x_i}{\Delta\mathbf{x}_i}, \qquad \alpha_s = \max_{\Delta\mathbf{s}_i < 0} -\frac{s_i}{\Delta\mathbf{s}_i}.$$

With this method, we get α_x and α_s which are the maximum values without violating the non-negativity criteria of $\mathbf{x}$ and $\mathbf{s}$.

Taking the minimum of α_x and α_s guarantees that the new points have nonnegative coordinates:

$$\alpha_q = \min\{\alpha_x, \alpha_s\}.$$

However, in some cases, $\alpha_q > 1$. In this case, the points might get far from the central path, although in some cases choosing a threshold where we truncate α_q gives a faster convergence to the optimal solution.

Finally, it is important to mention that in each problem if the step length is α_q, after some iterations the new points can get stuck at one of the corners of the polyhedron. We multiply α_q by the decrease parameter $\sigma \in (0,1)$. Based on our experience, in case of the problems that have the matrix with size of at most 200, $\sigma = 0.9999$ is a satisfying choice.

In the case of the PC IPA that works in a wide neighborhood of the central path, we made a more conservative step length calculation. In the predictor step we used (4.2.11), while in the corrector step we used (4.2.27) to determine the step length. It is important to mention that in the case of the corrector step we used the following method for getting the satisfactory step length. We started with $\alpha = 0.1$ and we increased α by 0.1 in each step if the new point with the current value of α would be in the neighborhood $\mathcal{D}_\varphi(\beta)$ defined in (5.3.2). If the new point was outside of neighborhood $\mathcal{D}_\varphi(\beta)$, we set $\alpha = \alpha - 0.1$ and we used it for calculating the points after the corrector steps. We tried to refine α by increasing it by 0.01 in each step after getting $\alpha = \alpha - 0.1$, however we got the same iteration number for our PC IPA, while the CPU time increased a lot.

5.3 Neighborhood, feasibility

In this section, the types of neighborhoods and the verification of the feasibility of the points is discussed.

5.3.1 Neighborhood

We considered two types of neighborhoods of the central path. The first one is the small neighborhood mentioned in Chapter 2 and Chapter 3. The other one is the generalized Potra-type wide neighborhood showed in Chapter 4.

The implemented small neighborhood for the PD IPA [26] and for the PC IPA [25] is the following:

$$\mathcal{N}_2(\tau,\mu) := \left\{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^+ : \left\| \frac{\mathbf{a}_\varphi}{2\sqrt{\mu \mathbf{x}\mathbf{s}}} \right\| \le \tau \right\}, \tag{5.3.1}$$

where $\tau \in (0,1)$ and φ is a function satisfying conditions that were given in Definitions 2.1.1 or 3.1.1.

The implemented generalized Potra-type wide neighborhood for the PC IPA [57] is the following:

$$\mathcal{D}_\varphi(\beta) := \{(\mathbf{x},\mathbf{s}) \in \mathcal{F}^+ : \varphi(\mathbf{u}) \ge \beta\varphi(\mathbf{e})\}. \tag{5.3.2}$$

Both τ and β are parameter options for the implemented solver.

If, with the implemented step length, we do not remain in the neighborhood, then we use a halving method to calculate a good enough α for which we remain in the corresponding neighborhood. The halving method has similarities to the method mentioned earlier in this chapter, however, in this case, we have a starting lower bound and upper bound for α and we calculate the point α_τ between the lower and upper bound. If the new points calculated with α_τ are in the τ neighborhood, we increase the lower bound to α_τ. If they are not in the corresponding neighborhood, we decrease the upper bound to α_τ. After that, we calculate the new α_τ again. We continue this method until the difference between the upper bound and the lower bound is less than 10^{-3}. We tried to set smaller difference than 10^{-3}, though the CPU time significantly increased.

5.3.2 Feasibility

It is important to mention that in theory, in each iteration the feasibility condition, that is $-M\mathbf{x}+\mathbf{s}=\mathbf{q}$, should be satisfied. However, in practice we calculate the residual vector $-M\mathbf{x}+\mathbf{s}-\mathbf{q}=\mathbf{r}_q$. This vector $\mathbf{r}_q$ has really small values. In the implementation we use this vector in the right hand side of the Newton system (5.1.5):

$$\begin{bmatrix} -M & I \\ S & X \end{bmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_q \\ \mathbf{a}_\varphi \end{pmatrix}. \tag{5.3.3}$$

From system (5.1.7) we get

$$(X^{-1}S+M)\Delta\mathbf{x} = X^{-1}\mathbf{a}_\varphi + \mathbf{r}_q. \tag{5.3.4}$$

Finally, using system (5.1.8) we have

$$\begin{bmatrix} -\bar{M} & I \\ I & I \end{bmatrix} \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_s \end{pmatrix} = \begin{pmatrix} -\frac{1}{\sqrt{\mu}} D\mathbf{r}_q \\ \mathbf{p}_\varphi \end{pmatrix}, \tag{5.3.5}$$

where $\bar{M}$ is given in (1.4.7), $\mathbf{d}_x, \mathbf{d}_s$ in (1.4.5), D in (1.4.7) and $\mathbf{p}_\varphi$ in (1.4.8). With this method, we achieve more precision in the calculations, although we mostly do not use it, since in terms of iteration numbers we did not get better results. However, we have a parameter to switch between using $\mathbf{r}_q$ in the corresponding part of the right hand side vector or not. Note that we accounted the feasibility of the iterates by calculating $\|\mathbf{r}_q\|_\infty$.

5.4 Differences between theoretical and implemented IPAs

In this section, we show the three types of IPAs that were implemented. First of all, it is important to mention again that calculating the real handicap κ would be NP-hard [16], this is why we calculate the local κ in each iteration in the following way:

$$\kappa(\Delta\mathbf{x}) = -\frac{1}{4} \frac{(\Delta\mathbf{x})^T M \Delta\mathbf{x}}{\sum_{i \in \mathcal{I}_+(\Delta\mathbf{x})} \Delta\mathbf{x}_i (M\Delta\mathbf{x})_i}, \tag{5.4.1}$$

where $\mathcal{I}_+(\Delta\mathbf{x}) = \{1 \leq i \leq n : x_i(M\mathbf{x})_i > 0\}$. This means we calculate κ with respect to the Newton direction $\Delta\mathbf{x}$ in each step of the algorithm.

Table 5.4.1 shows the implemented AET functions.

φ	$\mathbf{a}_\varphi$	$\mathbf{p}_\varphi$
$\varphi_1(t) = t$	$\mu\mathbf{e} - \mathbf{xs}$	$\mathbf{v}^{-1} - \mathbf{v}$
$\varphi_2(t) = \sqrt{t}$	$2(\sqrt{\mu\mathbf{xs}} - \mathbf{xs})$	$2(\mathbf{e} - \mathbf{v})$
$\varphi_3(t) = t^2 - t + \sqrt{t}$	$\frac{4\mu^2\sqrt{\mathbf{xs}} + 2\mu\mathbf{xs}\sqrt{\mathbf{xs}} - 3\mu\mathbf{xs}\sqrt{\mu}}{2(4\mathbf{xs}\sqrt{\mathbf{xs}} - 2\mu\sqrt{\mathbf{xs}} + \mu\sqrt{\mu})} - \frac{\mathbf{xs}}{2}$	$\frac{2(\mathbf{e} - \mathbf{v}^4 + \mathbf{v}^2 - \mathbf{v})}{4\mathbf{v}^3 - 2\mathbf{v} + \mathbf{e}}$

Table 5.4.1: The effect of different AETs on the right hand sides of the Newton- and scaled Newton-systems.

Note that these are members of the classes of AET functions defined in Definitions 2.1.1 and 3.1.1.

However, in our implementation we used a kernel function as well, to have a comparison. We used the following kernel function

$$\psi_{p,\sigma}(t) = \frac{t^{p+1}-1}{p+1} + \frac{\mathrm{e}^{\sigma(1-t)-1}}{\sigma}, \quad p \in [0,1],\ \sigma \geq 1. \tag{5.4.2}$$

From this, we have

$$f(t) = -\psi'_{p,q}(t) = \mathrm{e}^{\sigma(1-t)} - t^p. \tag{5.4.3}$$

Finally, after substitution we consider:

$$\mathbf{p}_\varphi = f(\mathbf{v}) = \mathrm{e}^{\sigma(\mathbf{e}-\mathbf{v})} - \mathbf{v}^p. \tag{5.4.4}$$

In the case of the parametric kernel function given in (5.4.2) we have

$$\mathbf{a}_\varphi = \mu\sqrt{\frac{\mathbf{xs}}{\mu}}e^{\sigma\left(\mathbf{e}-\sqrt{\frac{\mathbf{xs}}{\mu}}\right)} - \mu\left(\sqrt{\frac{\mathbf{xs}}{\mu}}\right)^{p+1}. \tag{5.4.5}$$

It should be mentioned that we can get PC algorithm only in the case when $p=1$. In this case, we have $c=1$, so we have $-\mathbf{xs}$ as the right-hand side of system (3.1.3).

First, consider the implementation of Algorithm 1. In this case, we present the obtained numerical results.

Algorithm 6 : Implemented PD IPA based on a new class of AETs

Input:
a function φ that satisfies the conditions given in Definition 2.1.1;
accuracy parameter: $\varepsilon = 10^{-5}$;
update parameter: $\theta = 0.999$;
proximity parameter: $\tau = 1$;
reducing parameter: $\sigma = 0.9999$;
strictly feasible starting points $(\boldsymbol{x}^0, \boldsymbol{s}^0)$, $\mu^0 > 0$, such that $\delta(\boldsymbol{x}^0, \boldsymbol{s}^0; \mu^0) \leq \tau$;
sufficient matrix $M \in \mathbb{R}^{n\times n}$ and vector $\boldsymbol{q} \in \mathbb{R}^n$.

Iteration:
begin
$\boldsymbol{x} := \boldsymbol{x}^0$, $\boldsymbol{s} := \boldsymbol{s}^0$, $\mu := \mu^0$;
while $n\mu \geq \varepsilon$
$\mu := (1-\theta)\mu$;
Calculate $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{s}$ from (5.1.7), (5.1.8) using (3.1.3) and Table 5.4.1
with LU-decomposition or QR decompositon (rmatrixlu(),rmatrixqr(),rmatrixsolve());
Calculate $\kappa(\Delta\boldsymbol{x})$ with (5.4.1);
$\alpha_{x_h} := \{-\frac{x_i}{\Delta x_i} : \Delta x_i < 0\}$ *and* $\alpha_{s_h} := \{-\frac{s_i}{\Delta s_i} : \Delta s_i < 0\}$;
$\alpha_h := \sigma \cdot \min\{\alpha_{x_h}, \alpha_{s_h}\}$;
$\alpha := \min\{\alpha_h, 3.0\}$;
$\boldsymbol{x}^+ := \boldsymbol{x} + \alpha\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^+ := \boldsymbol{s} + \alpha\Delta\boldsymbol{s}$;
$\|\boldsymbol{r}_q\|_\infty := \left\|-M\boldsymbol{x}^+ + \boldsymbol{s}^+ - \boldsymbol{q}\right\|_\infty$;
$\boldsymbol{v}^+ := \sqrt{\frac{\boldsymbol{x}^+\boldsymbol{s}^+}{\mu}}$ *and* $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) := \left\|\boldsymbol{p}_{\varphi^+}\right\|_2$;
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) > \tau$
"Not in the τ neighborhood."
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu^+) > 2\cdot\tau$
"Not in its own τ neighborhood."
$\mu := \mu^+ := \frac{(\boldsymbol{x}^+)^T\boldsymbol{s}^+}{n}$;
end
end.
Output $(\boldsymbol{x}, \boldsymbol{s})$: $n\mu < \varepsilon$

After presenting the algorithm for the implemented PD IPA, we present the implemented version of Algorithm 2.

Algorithm 7 : Implemented PC IPA based on a new class of AETs

Input:

a function φ that satisfies the conditions given in Definition 3.1.1;
accuracy parameter: $\varepsilon = 10^{-5}$;
update parameter: $\theta = 0.999$;
proximity parameter: $\tau = 1$;
reducing parameter: $\sigma = 0.9999$;
strictly feasible starting points $(\boldsymbol{x}^0, \boldsymbol{s}^0)$, $\mu^0 > 0$, such that $\delta(\boldsymbol{x}^0, \boldsymbol{s}^0; \mu^0) \leq \tau$;
sufficient matrix $M \in \mathbb{R}^{n \times n}$ and vector $\boldsymbol{q} \in \mathbb{R}^n$.

Input:

begin
$\boldsymbol{x} := \boldsymbol{x}^0$, $\boldsymbol{s} := \boldsymbol{s}^0$, $\mu := \mu^0$;
while $n\mu \geq \varepsilon$
$\mu := (1-\theta)\mu$;
(predictor step)
Calculate $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{s}$ from (5.1.7), (5.1.8) using (3.1.3) and Table 5.4.1
with LU-decomposition or QR decompositon (rmatrixlu(),rmatrixqr(),rmatrixsolve());
Calculate $\kappa(\Delta\boldsymbol{x})$ with (5.4.1);
$\alpha_{x_h} := \{-\frac{x_i}{\Delta x_i} : \Delta x_i < 0\}$ *and* $\alpha_{s_h} := \{-\frac{s_i}{\Delta s_i} : \Delta s_i < 0\}$;
$\alpha_h := \sigma \cdot \min\{\alpha_{x_h}, \alpha_{s_h}\}$;
$\alpha := \min\{\alpha_h, 3.0\}$;
$\boldsymbol{x}^+ := \boldsymbol{x} + \alpha\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^+ := \boldsymbol{s} + \alpha\Delta\boldsymbol{s}$;
$\|\boldsymbol{r}_q\|_\infty := \|-M\boldsymbol{x}^+ + \boldsymbol{s}^+ - \boldsymbol{q}\|_\infty$;
$\boldsymbol{v}^+ := \sqrt{\frac{\boldsymbol{x}^+\boldsymbol{s}^+}{\mu}}$;
$\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) := \|\boldsymbol{p}_{\varphi^+}\|_2$;
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) > \tau$
"Not in the τ neighborhood."
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu^+) > 2 \cdot \tau$
"Not in its own τ neighborhood."
$\mu := \mu^+ := \frac{(\boldsymbol{x}^+)^T \boldsymbol{s}^+}{n}$;
(corrector step)
Calculate $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{s}$ from (5.1.7), (5.1.8) with
LU-decomposition or QR decompositon (rmatrixlu(),rmatrixqr(),rmatrixsolve());
Calculate the $\kappa(\Delta\boldsymbol{x})$ with (5.4.1);
$\alpha_{x_h} := \{-\frac{x_i}{\Delta x_i} : \Delta x_i < 0\}$ *and* $\alpha_{s_h} := \{-\frac{s_i}{\Delta s_i} : \Delta s_i < 0\}$;
$\alpha_h := \sigma \cdot \min\{\alpha_{x_h}, \alpha_{s_h}\}$;
$\alpha := \min\{\alpha_h, 3.0\}$;
$\boldsymbol{x}^+ := \boldsymbol{x} + \alpha\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^+ := \boldsymbol{s} + \alpha\Delta\boldsymbol{s}$;
$\|\boldsymbol{r}_q\|_\infty := \|-M\boldsymbol{x}^+ + \boldsymbol{s}^+ - \boldsymbol{q}\|_\infty$;
$\boldsymbol{v}^+ := \sqrt{\frac{\boldsymbol{x}^+\boldsymbol{s}^+}{\mu}}$;
$\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) := \|\boldsymbol{p}_{\varphi^+}\|_2$;
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu) > \tau$
"Not in the τ neighborhood."
if $\delta(\boldsymbol{x}^+, \boldsymbol{s}^+; \mu^+) > 2 \cdot \tau$
"Not in its own τ neighborhood."
$\mu := \mu^+ := \frac{(\boldsymbol{x}^+)^T \boldsymbol{s}^+}{n}$;
end
end.

Output $(\boldsymbol{x}, \boldsymbol{s})$: $n\mu < \varepsilon$

Finally, consider the implementation of Algorithm 4.

Algorithm 8 : Implemented PC IPA acting in a wide neighborhood of the central path

Input:

$(\boldsymbol{x}^0, \boldsymbol{s}^0) \in \mathcal{D}_\varphi(\beta)$, $\beta \in (0,1)$;
set $\kappa = 1$ *and* $\mu_0 = \mu(\boldsymbol{x}^0, \boldsymbol{s}^0)$;
set precision value $\varepsilon > 0$;
sufficient matrix $M \in \mathbb{R}^{n \times n}$ *and vector* $\boldsymbol{q} \in \mathbb{R}^n$.

Iteration:

begin
$\boldsymbol{x} := \boldsymbol{x}^0$, $\boldsymbol{s} := \boldsymbol{s}^0$, $\mu := \mu^0$ *and* $k = 0$;
while $n\mu \geq \varepsilon$
(predictor step)
Calculate affine direction from (4.2.2) using (5.1.7)
with LU-decomposition or QR decompositon (rmatrixlu(),rmatrixqr(),rmatrixsolve());
Calculate $\kappa(\Delta \boldsymbol{x})$ *with (5.4.1);*
Calculate the predictor steplength using (4.2.11);
Calculate $(\boldsymbol{x}^p, \boldsymbol{s}^p)$;
if $\mu(\boldsymbol{x}^p, \boldsymbol{s}^p) = 0$
STOP; Optimal solution found;
end
if $(\boldsymbol{x}^p, \boldsymbol{s}^p) \in \mathcal{D}_\varphi(\beta)$
$(\boldsymbol{x}^{k+1}, \boldsymbol{s}^{k+1}) = (\boldsymbol{x}^p, \boldsymbol{s}^p)$, $\mu^{k+1} = \mu(\boldsymbol{x}^p, \boldsymbol{s}^p)$, $k = k+1$, **go to (predictor step)**;
end
(corrector step)
Calculate centering direction from (4.2.25) using (5.1.7)
with LU-decomposition or QR decompositon (rmatrixlu(),rmatrixqr(),rmatrixsolve());
Calculate $\kappa(\Delta \boldsymbol{x})$ *with (5.4.1);*
Calculate centering steplenght using (4.2.27);
Calculate $(\boldsymbol{x}^c, \boldsymbol{s}^c)$;
if $(\boldsymbol{x}^c, \boldsymbol{s}^c) \in \mathcal{D}_\varphi(\beta)$
$(\boldsymbol{x}^{k+1}, \boldsymbol{s}^{k+1}) = (\boldsymbol{x}^c, \boldsymbol{s}^c)$, $\mu^{k+1} = \mu(\boldsymbol{x}^c, \boldsymbol{s}^c)$, $k = k+1$;
end
else
$\kappa = 2\kappa$;
$(\boldsymbol{x}^{k+1}, \boldsymbol{s}^{k+1}) = (\boldsymbol{x}^k, \boldsymbol{s}^k)$, $\mu^{k+1} = \mu^k$;
end
end
end.

Output $(\boldsymbol{x}^k, \boldsymbol{s}^k) : \boldsymbol{x}^{kT} \boldsymbol{s}^k \leq \varepsilon$

In the following section, we present the obtained numerical results.

5.5 Numerical results

We implemented Algorithms 6, 7, and 8 in the C++ programming language. Due to the fact that in many cases we do not have information about the value of κ, in our implementation we used algorithms that do not depend on the handicap. We set the values $\theta = 0.999$ and $\epsilon = 10^{-5}$.

Moreover, it should be mentioned that most of the numerical results related to $P_*(\kappa)$-LCPs are related to problems where the value of κ is zero, which lead to LO problems. Gurtuna et al. [44] and Asadi et al. [11] provided numerical results related to $P_*(\kappa)$-LCPs having positive handicap, by considering 2×2 or 3×3 matrices. They also analysed block diagonal matrices formed from the aforementioned ones. Darvay et al. [25] presented numerical results where they solved $P_*(\kappa)$-LCPs with matrices having positive κ generated by Illés and Morapitiye [52].

5.5.1 Numerical results based on the Csizmadia matrix

Let us consider the following matrix that is called the Csizmadia matrix

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ -1 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & 1 \end{pmatrix}, \tag{5.5.1}$$

where $M \in \mathbb{R}^{n\times n}$. In [33], the authors, E.-Nagy and de Klerk proved that the handicap of this matrix is exponential in the size of the problem. Furthermore, E.-Nagy et al. [81] showed that in this case $\hat{\kappa}(M) = 2^{2n-8} - 0.25$. However, we obtained promising numerical results in this case, as well.

We generated 10 problems of each size of 10, 30, 50, 70, 100 with random starting point pairs $(\mathbf{x}, \mathbf{s})$ based on the matrix (5.5.1). The right-hand side vector of these problems is computed as $\mathbf{q} = -M\mathbf{x} + \mathbf{s}$. Based on the starting points, the number of the problems that were generated can be found in Table 5.5.1.

We present promising results in the case of IPAs for solving LCPs with the matrix (5.5.1) that can be found on the website [100].

In the case of Algorithms 6 and 7, we have the same interpretation for most of the results, so the following explanations hold for both the PD IPAs and PC IPAs working in small neighborhood of the central path. However, there are some differences that will be explained in detail in Subsubsection 5.5.1.1 and 5.5.1.2.

values of $\mathbf{s}$ \ values of $\mathbf{x}$	$[0,1]^n$	$[0,10]^n$	$[0,100]^n$	$[9,11]^n$	$[9900,11000]^n$
$[0,1]^n$	50	0	0	50	0
$[0,10]^n$	0	50	0	0	0
$[0,100]^n$	0	0	50	0	50
$[9,11]^n$	50	0	0	0	0
$[9900,11000]^n$	0	0	50	0	0

Table 5.5.1: Testing data generated from (5.5.1).

We defined Algorithms 6 and 7 by using member functions of our classes of AET functions from Chapter 2 and Chapter 3. Furthermore, we compared our results with Algorithms 6 and 7, that were based on a kernel function. Our results are based on three AET functions and a kernel function for both type of IPAs. We compared Algorithms 6 and 7 based on the AET functions φ_1, φ_2 and φ_3 from Table 5.4.1 with Algorithms 6 and 7 based on kernel function $\psi_{p,\sigma}(t) = \frac{t^{p+1}-1}{p+1} + \frac{e^{\sigma(1-t)}-1}{\sigma}$, $p \in [0,1]$, $\sigma \geq 1$, see [43, 107], respectively. In our implementation, we used $\psi_{1,5}$.

5.5.1.1 Primal-dual interior-point algorithm

It is important to highlight that the results we got with Algorithm 6 show exactly the same experiences for the problems as we mentioned in the case of Algorithm 7. These experiences are based on the starting points of the problem. If the values of $\mathbf{x}$ are similar or significantly larger than the valuse of $\mathbf{s}$, the iteration numbers significantly increase with respect to the difference between the two vectors, while in the opposite case the iteration numbers are almost constant or slightly increase in the size of the matrix. In the case of $\psi_{p,\sigma}(t)$ with $p = 1.0$ and $\sigma = 5$, we obtained similar results with the PD IPA as with Algorithm 6 based on the functions φ_1, φ_2, φ_3. However, by using $\psi_{p,\sigma}$ with $p = 0.2$ and $\sigma = 5$, we obtained significantly worse results than with Algorithm 6 using the mentioned AET functions. It is important to mention that in the case of Tables 5.5.2 and 5.5.4, we had to decrease the reducing parameter to $\sigma = 0.95$, since otherwise the algorithm failed to produce $\mathbf{x}^T\mathbf{s} < \varepsilon$. Furthermore, it can be seen in practice that Algorithm 6 perform worse than Algorithm 7 in all cases. Points may overlap in figures.

Problems / IPA	$\mathbf{x}, \mathbf{s} \in [0,1]^{100}$	$\mathbf{x}, \mathbf{s} \in [0,10]^{100}$	$\mathbf{x}, \mathbf{s} \in [0,100]^{100}$
$\varphi_1(t)$	59.4	63.1	61.5
$\varphi_2(t)$	61.4	65.8	64.4
$\varphi_3(t)$	61.2	65.2	63.4
$\psi_{0.2,5}(t)$	652.7	997.4	1254.8

Table 5.5.2: Average iteration numbers of Algorithm 6 on problems of size 100×100.

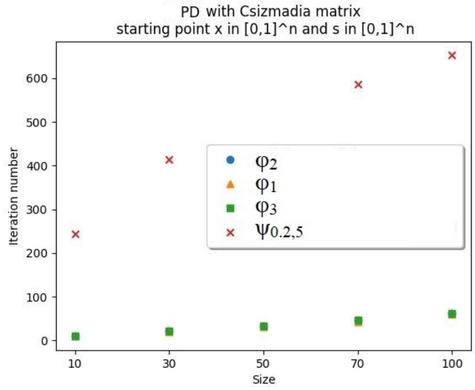


Figure 5.1: Iteration numbers for all different variants of Algorithm 6 are far from the theoretical ones.

Problems / IPA	$\mathbf{x} \in [0,1]^{100}, \mathbf{s} \in [9,11]^{100}$	$\mathbf{x} \in [0,100]^{100}, \mathbf{s} \in [9900,11000]^{100}$	$\mathbf{x} \in [0,1]^{200}, \mathbf{s} \in [9,11]^{200}$	$\mathbf{x} \in [0,100]^{200}, \mathbf{s} \in [9900,11000]^{200}$
$\varphi_1(t)$	7.2	6.0	10.0	6.0
$\varphi_2(t)$	8.0	7.0	10.8	8.0
$\varphi_3(t)$	7.0	6.0	9.9	6.0
$\psi_{0.2,5}(t)$	73.8	24.2	145.3	37.3

Table 5.5.3: Average iteration numbers of Algorithm 6 on problems of size 100×100 and 200×200.

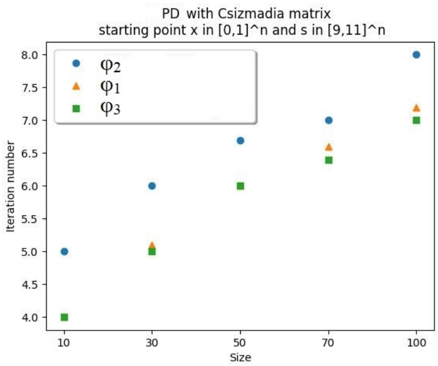


Figure 5.2: Iteration numbers for all different variants of Algorithm 6 are slightly increasing.

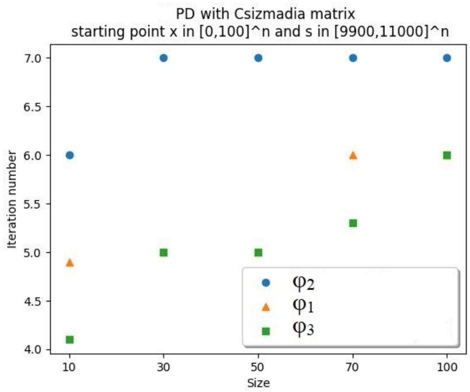


Figure 5.3: Iteration numbers for all different variants of Algorithm 6 are almost constant.

Problems / IPA	$\mathbf{x} \in [9,11]^{100}, \mathbf{s} \in [0,1]^{100}$	$\mathbf{x} \in [9900,11000]^{100}, \mathbf{s} \in [0,100]^{100}$
$\varphi_1(t)$	239.1	423.9
$\varphi_2(t)$	246.8	478.7
$\varphi_3(t)$	245.0	470.7
$\psi_{0.2,5}(t)$	992.0	1946.4

Table 5.5.4: Average iteration numbers of Algorithm 6 on problems of size 100×100.

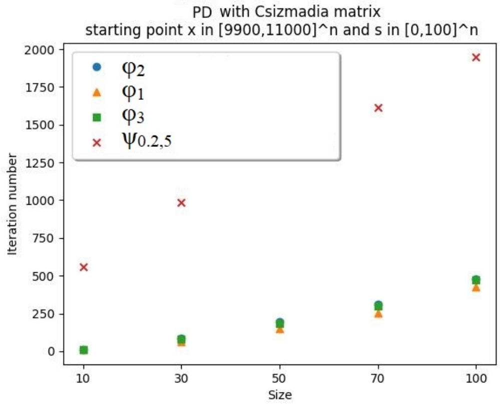


Figure 5.4: Iteration numbers of Algorithm 6 are larger than in the case shown in Figure 5.5.5

5.5.1.1.1 Connection between local κ, $\|\Delta x \Delta s\|$, duality gap and step length based on PD IPA on Csizmadia matrix

In this section, we show some results that were presented in [53]. We found important connections between the local κ value defined in 5.4.1, $\|\Delta x \Delta s\|$, the duality gap and the step length.

Taking into consideration the Csizmadia matrix defined in (5.5.1) of size 100×100, we generated an LCP with all one vectors as starting points, where $\mathbf{q} = -M\mathbf{e} + \mathbf{e}$. We tested the connections mentioned before, the local kappa, the iteration numbers and the step length. It is important to mention that in Figure 5.5 that while the value $\kappa(\Delta \mathbf{x})$ is also exponentially decreasing in each step, the step length is exponentially increasing.

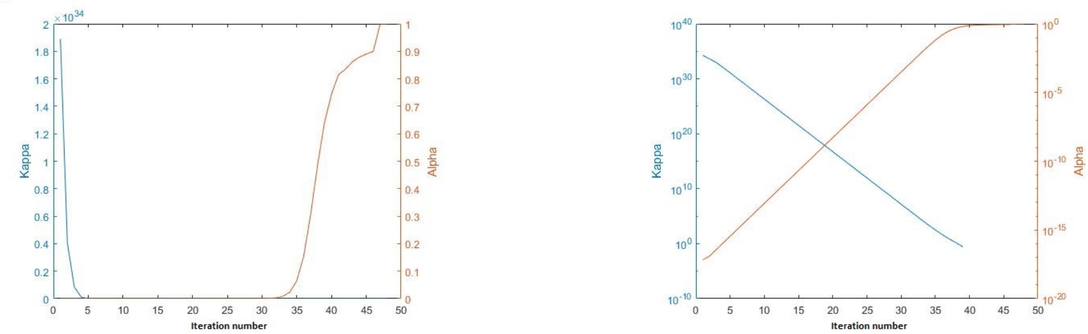


Figure 5.5: Connection between the local κ and the step length (on the right-hand side plotted with logarithmic scale).

Furthermore, from Figure 5.6 with logarithmic scale we see that until the value $\kappa(\Delta \mathbf{x})$ is significantly larger than 0, the duality gap is not significantly decreasing. However, as the value $\kappa(\Delta \mathbf{x})$ approaches 0, the duality gap starts to decrease exponentially.

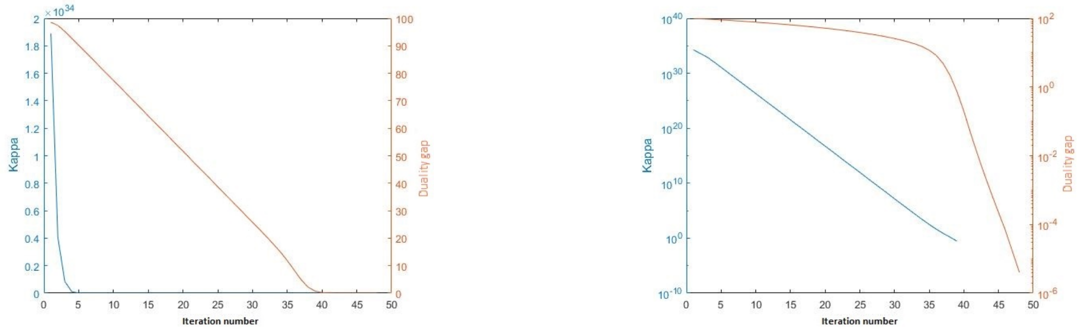


Figure 5.6: Connection between the local κ and the duality gap (on the right-hand side plotted with logarithmic scale).

Finally, the value $\kappa(\Delta \mathbf{x})$ has a similar exponential decrease as $\|\Delta x \Delta s\|$, which can be seen in Figure 5.7 with logarithmic scale.

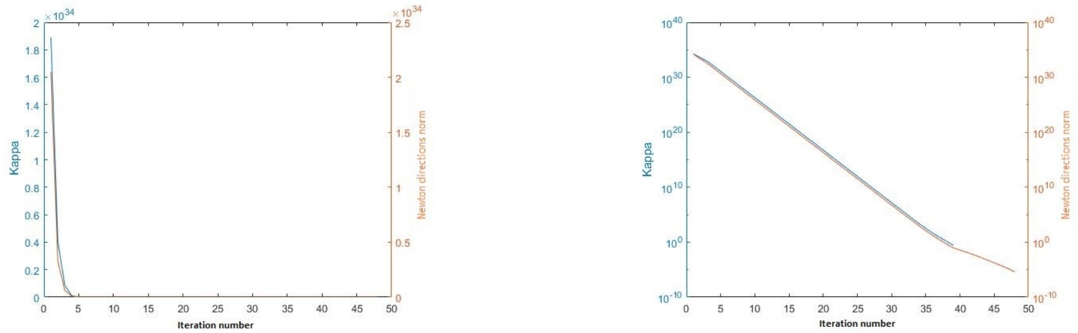


Figure 5.7: Connection between the local κ and $\|\Delta x \Delta s\|$ (on the right-hand side plotted with logarithmic scale).

In the following subsection, we present the numerical results obtained by using the PC IPA based on the new class of AET functions.

5.5.1.2 Predictor-corrector interior-point algorithm

In the case of Algorithm 7, we show the average iteration numbers for problems of size 100×100 and with the starting points $\mathbf{x}$ and $\mathbf{s}$ from similar interval as in Table 5.5.5. We can see that the number of iterations does not change significantly. In Figure 5.8, one can see that even if the handicap was exponential, the number of iterations slightly increases with the size of the problem. Points may overlap in figures.

IPA \ Problems	$\mathbf{x}, \mathbf{s} \in [0,1]^{100}$	$\mathbf{x}, \mathbf{s} \in [0,10]^{100}$	$\mathbf{x}, \mathbf{s} \in [0,100]^{100}$
$\varphi_1(t)$	51.7	55.7	55.0
$\varphi_2(t)$	52.15	56.2	55.45
$\varphi_3(t)$	51.85	56.1	55.15
$\psi_{1,5}(t)$	62.2	67.0	65.75

Table 5.5.5: Average iteration numbers of Algorithm 7 on problems of size 100×100.

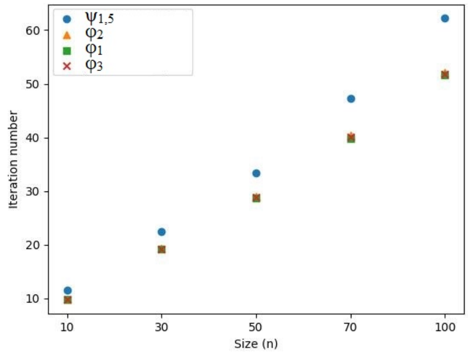


Figure 5.8: Iteration numbers for all different variants of Algorithm 7 are far from the theoretical ones.

We present our results in the case of Algorithm 7 on test cases where the values of $\mathbf{x}$ are significantly smaller than the values of $\mathbf{s}$, see in Table 5.5.6 and in Figures 5.9 and 5.10. It can be seen that the iteration numbers are almost constant.

IPA \ Problems	$\mathbf{x} \in [0,1]^{100}, \mathbf{s} \in [9,11]^{100}$	$\mathbf{x} \in [0,100]^{100}, \mathbf{s} \in [9900,11000]^{100}$	$\mathbf{x} \in [0,1]^{200}, \mathbf{s} \in [9,11]^{200}$	$\mathbf{x} \in [0,100]^{200}, \mathbf{s} \in [9900,11000]^{200}$
$\varphi_1(t)$	6.0	5.05	7.0	4.8
$\varphi_2(t)$	6.5	5.55	8.1	6.8
$\varphi_3(t)$	6.0	4.45	7.0	4.0
$\psi_{1,5}(t)$	7.0	5.05	9.7	6.0

Table 5.5.6: Average iteration numbers for Algorithm 7 on problems of size 100×100 and 200×200.

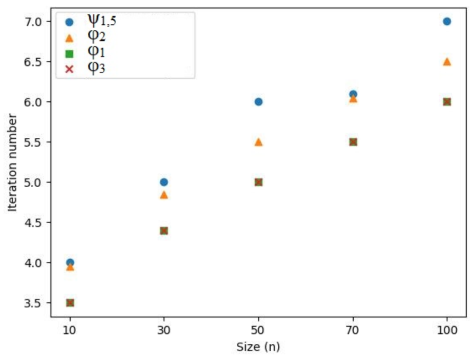


Figure 5.9: Iteration numbers for all different variants of Algorithm 7 are slightly increasing.

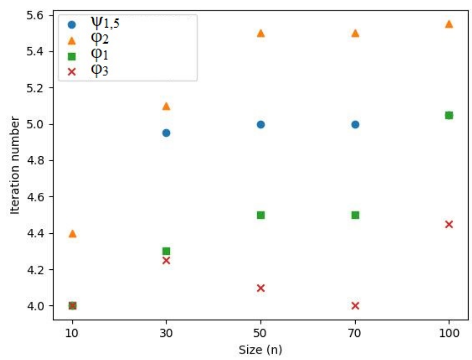


Figure 5.10: Iteration numbers for all different variants of Algorithm 7 are almost constant.

We showed results in case of Algorithm 7 on problems of size of 100×100 and with starting points where the values of $\mathbf{x}$ are significantly larger than the values of $\mathbf{s}$ in Table 5.5.7 and Figure 5.11. These results show more iterations than in the case of problems with starting points where the values of $\mathbf{x}$ are significantly smaller than the values of $\mathbf{s}$.

IPA \ Problems	$\mathbf{x} \in [9,11]^{100}, \mathbf{s} \in [0,1]^{100}$	$\mathbf{x} \in [9900, 11000]^{100}, \mathbf{s} \in [0,100]^{100}$
$\varphi_1(t)$	200.65	370.3
$\varphi_2(t)$	201.4	379.8
$\varphi_3(t)$	201.1	377.7
$\psi_{1,5}(t)$	249.5	490.4

Table 5.5.7: Average iteration numbers of Algorithm 7 on problems of size 100×100.

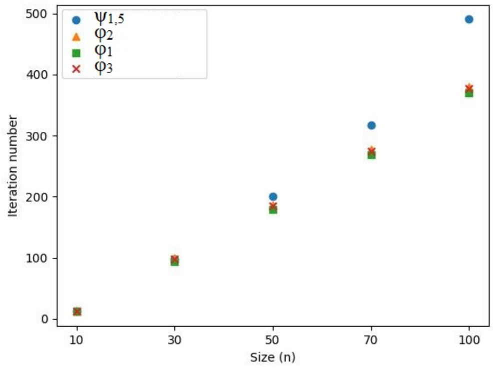


Figure 5.11: Iteration numbers of Algorithm 7 are larger than in the case shown in Figure 5.5.5

Finally, it can be seen that the Algorithm 7 based on the kernel function shows slightly worse iteration numbers than the algorithms based on the three AET functions belonging to the class of AET functions presented in Section 3.1. Furthermore, Algorithm 7 based on the function φ_3 show slightly better results than φ_2, and similar results as φ_1.

5.5.2 Numerical results based on positive semidefinite and bisymmetric matrices

Numerical results in the case of Algorithm 6 based on functions belonging to the class defined in Definition 2.1.1 can be found in [26]. Furthermore, results in case of Algorithm 7 based on functions that are members of the class of AET functions defined in Definition 3.1.1 can be found in [25, 59]. Finally, other numerical results regarding a Algorithm 8 that is working in a wide neighborhood of the central path are presented in [57].

We present our numerical results using the matrices from the website [100]. Consider matrix A, which is a constraint matrix of an LO problem from the NETLIB test collection [83]. With A, we generated PSD matrices by $A^T A$ and AA^T. The reason for this is to use the structure of those LO problems. We called those matrices "TR_PSD", and with these matrices we also considered bisymmetric matrices, that we called "TR_BS" matrices. The matrices called

"TR_BS" were generated as follows:

$$M = \begin{bmatrix} P & B \\ -B^T & Q \end{bmatrix}, \tag{5.5.2}$$

where P and Q are PSD matrices of size $n \times n$ and $m \times m$ and $B \in \mathbb{R}^{n \times m}$ is an arbitrary matrix. We used the "TR_PSD" matrices as the PSD matrices P and Q in (5.5.2). The bisymmetric matrices with structure shown in (5.5.2) first occured in a paper of Klafszky and Terlaky [98]. They showed theoretical results related to symmetric, linearly constrained convex QP problems. Finally, our goal was to acquire matrices that are $P_*(\kappa)$-matrices with specific values κ. We wanted to analyse the connection between the size of κ and the practical iteration numbers. We generated these matrices from "TR_PSD" and "TR_BS" with the scaling technique shown in [69] from Theorem 3.5. It is important to mention that the scaling technique was mentioned in other articles as well, like in [81]. In the scaling technique, we defined matrices U and V as identity matrices, except for one element at the same index, which was $\sqrt{1+4\bar{\kappa}}$ in the case of matrix U and $\dfrac{1}{\sqrt{1+4\bar{\kappa}}}$ for matrix V, where $\bar{\kappa}$ is the desired value of κ for the new modified matrix. Finally, the $P_*(\bar{\kappa})$ matrix was obtained by computing $\bar{M}' = UMV$.

We generated 39 test matrices for each of the following values of $\bar{\kappa} = 0, 10, 100, 1000, 10000$ with the techniques mentioned above. However, since we obtained similar results in most of the cases, we present only the results for problems with $\bar{\kappa} = 0$ and $\bar{\kappa} = 10000$. We chose $\mathbf{x} = \mathbf{s} = \mathbf{e}$ as starting points, where $\mathbf{e}$ is the vector of all ones. We calculated the right-hand side as $\mathbf{q} = -M\mathbf{e} + \mathbf{e}$. With this, we ensured strict feasibility of the test LCPs.

The three AET functions and the parametric kernel function used here are similar to the ones that were presented in Subsection 5.5.1.

The results can be found in Tables 5.5.8, 5.5.9, and 5.5.10. In Tables 5.5.8, 5.5.9, and 5.5.10, the results can be found for the PSD and bisymmetric matrices. We also present results obtained by using the matrices generated by the technique that was mentioned in Theorem 3.5 in [70] and with $\bar{\kappa} = 10000$ in Tables 5.5.9, 5.5.8 and 5.5.10.

In all tables, the first column contains the name of the tested matrices. The second column is the size of the square matrices. Finally, in the last eight columns (in the case of Table 5.5.10, just the last four columns) the numbers of iterations can be found for the different search directions.

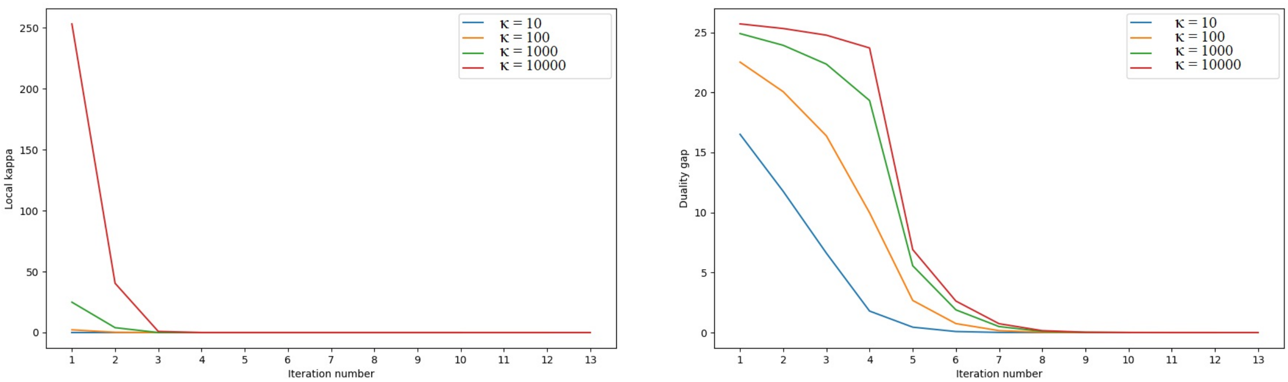
We show the changes of the local κ defined in (5.4.1) at each iteration of the algorithm based on the function φ_3. The chosen matrix for this purpose is "TR_PSD_1_2", since the maximum local κ was the largest for this one. It is easy to see that even in the case where the value κ of the matrix is 10000, the maximum of the local κ was less than 250. Furthermore, the local κ is the largest in the first iteration, and then it reaches 0 after some iterations.

Finally, due to the fact that the local κ is far smaller than the handicap of the matrices, the value κ has less effect on the iteration numbers than it should have based on the theoretical results.

5.5.2.1 Numerical results for primal-dual interior-point algorithms

In the case of Algorithm 6, the kernel function-based algorithm has similar results as the ones that were produced by the algorithms based on the three AET functions belonging to the introduced class in Definition 2.1.1.

In Figure 5.12 (a), we show the changes in the local κ in the case of the Algorithm 6, and in Figure 5.13 (b), the decrease of the duality gap. It is easy to see that the duality gap starts to decrease drastically after the local κ reaches zero.



(a) The changes of $\kappa_{\Delta x}$ for TR_PSD_1_2 (b) The changes of the duality gap for TR_PSD_1_2

Figure 5.12: It is important to mention that the iteration numbers are different for each test cases in Subfigure (a) and (b). This is obtained by using the Algorithm 6 based on φ_3.

In Table 5.5.8 we present the iteration numbers of Algorithm 6 using different search directions on matrices with $\kappa = 0$ and $\kappa = 10000$. Note that the boldface letter means that it represents the smallest number of iterations for a given matrix.

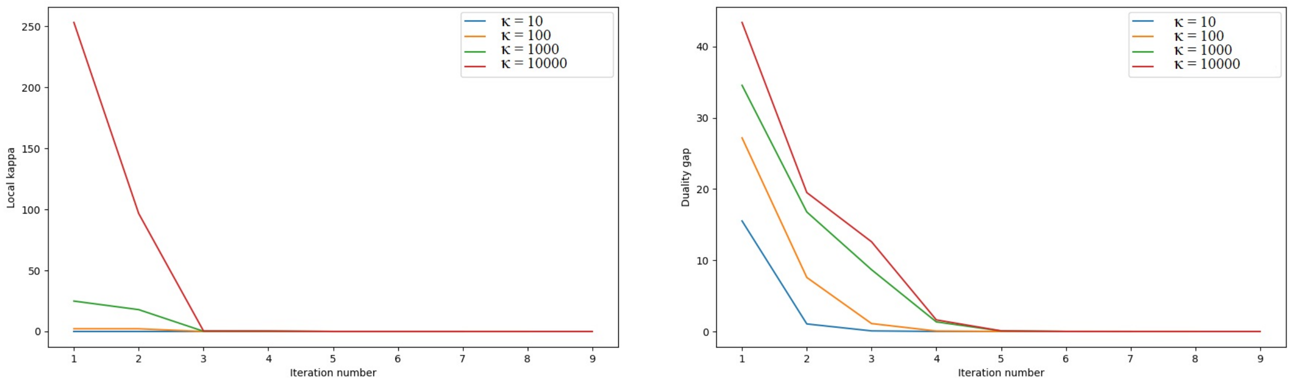
		$\bar{\kappa}=0$				$\bar{\kappa}=10000$			
Name of matrix	Size	φ_1	φ_2	φ_3	$\psi_{1,5}$	φ_1	φ_2	φ_3	$\psi_{1,5}$
TR_PSD_1_2	27	**10**	**10**	**10**	**10**	14	14	**13**	**13**
TR_PSD_2_2	32	**12**	**12**	**12**	**12**	17	17	**16**	**16**
TR_PSD_3_2	41	**11**	**11**	**11**	**11**	**15**	**15**	16	16
TR_PSD_4_2	43	**12**	**12**	**12**	**12**	**16**	**16**	**16**	**16**
TR_PSD_5_2	46	**11**	**11**	**11**	**11**	**15**	**15**	**15**	**15**
TR_PSD_6_2	47	**11**	**11**	**11**	**11**	**15**	**15**	**15**	**15**
TR_PSD_7_2	48	12	**11**	12	12	16	**15**	16	16
TR_PSD_8_2	48	**11**	**11**	**11**	**11**	**15**	**15**	**15**	**15**
TR_PSD_9_2	48	**11**	**11**	**11**	**11**	**15**	**15**	**15**	**15**
TR_PSD_10_2	49	12	**11**	12	12	**19**	**19**	20	**19**
TR_PSD_11_2	50	12	**11**	12	12	**19**	**19**	20	**19**
TR_PSD_12_2	50	12	**11**	12	12	16	**15**	16	16
TR_PSD_13_2	56	**9**	**9**	**9**	**9**	14	14	**13**	**13**
TR_BS_1_2	59	12	**11**	12	12	**14**	**14**	**14**	15
TR_BS_2_2	68	**10**	**10**	**10**	**10**	**13**	**13**	**13**	**13**
TR_BS_3_2	73	**10**	**10**	**10**	**10**	**15**	**15**	**15**	**15**
TR_BS_4_2	74	**10**	**10**	**10**	**10**	**15**	**15**	**15**	**15**
TR_BS_5_2	75	12	**11**	12	12	**20**	24	22	**20**
TR_BS_6_2	80	**12**	**12**	**12**	**12**	**14**	**14**	**14**	15
TR_BS_7_2	81	13	**12**	13	13	**15**	**15**	**15**	16
TR_BS_8_2	84	**12**	**12**	**12**	13	21	26	24	**20**
TR_BS_9_2	88	**11**	**11**	**11**	12	16	**15**	16	16
TR_BS_10_2	89	**12**	**12**	**12**	**12**	**16**	**16**	**16**	**16**
TR_BS_11_2	89	**11**	**11**	**11**	**11**	**16**	**16**	**16**	**16**
TR_BS_12_2	90	13	**12**	13	13	14	**13**	14	14
TR_BS_13_2	90	13	**12**	13	13	**16**	17	17	17
TR_BS_14_2	94	**11**	**11**	**11**	12	**15**	16	16	16
TR_BS_15_2	95	11	**10**	11	11	**15**	**15**	**15**	**15**
TR_BS_16_2	95	11	**10**	11	11	**16**	**16**	**16**	**16**
TR_BS_17_2	96	**11**	**11**	**11**	**11**	**15**	16	**15**	**15**
TR_BS_18_2	97	**11**	**11**	**11**	**11**	**17**	**17**	**17**	**17**
TR_PSD_14_2	97	**13**	**13**	**13**	14	**15**	**15**	**15**	**15**
TR_BS_19_2	102	**10**	**10**	**10**	**10**	**14**	15	**14**	**14**
TR_BS_20_2	103	11	**10**	11	11	**15**	**15**	**15**	**15**
TR_BS_21_2	124	15	**14**	15	15	19	19	**18**	19
TR_BS_22_2	140	**14**	**14**	**14**	**14**	**18**	**18**	**18**	**18**
TR_BS_23_2	145	16	**15**	16	16	21	**20**	**20**	**20**
TR_BS_24_2	146	15	**14**	15	15	20	**19**	**19**	**19**
TR_BS_25_2	153	**14**	**14**	**14**	**14**	19	19	**18**	**18**

Table 5.5.8: Iteration numbers of Algorithm 6 on matrices with $\bar{\kappa}=0$ and $\bar{\kappa}=10000$

5.5.2.2 Numerical results obtained for predictor-corrector interior-point algorithms

In the case of Algorithm 7, it is observed that the algorithm based on the chosen kernel function gives slightly worse results than the algorithms based on the three AET functions belonging to the introduced class in Definition 3.1.1. We can see that in practice, Algorithm 7 usually perform better than Algorithm 6.

We show the changes in the local κ in the case of the Algorithm 7 in Figure 5.13 (a). Furthermore, we show the decrease of the duality gap in Figure 5.13 (b).



(a) The changes of $\kappa_{\Delta x}$ for TR_PSD_1_2

(b) The changes of the duality gap for TR_PSD_1_2

Figure 5.13: It is important to mention that the iteration numbers are different for each test cases in Subfigure (a) and (b). This is obtained by using the Algorithm 7 based on φ_3.

In Table 5.5.9 we present the iteration numbers of Algorithm 7 using different search directions on matrices with $\kappa = 0$ and $\kappa = 10000$. Note that the boldface letter means that it represents the smallest number of iterations for a given matrix.

		$\bar{\kappa} = 0$				$\bar{\kappa} = 10000$			
Name of matrix	Size	φ_1	φ_2	φ_3	$\psi_{1,5}$	φ_1	φ_2	φ_3	$\psi_{1,5}$
TR_PSD_1_2	27	**7**	**7**	8	10	11	11	**9**	13
TR_PSD_2_2	32	9	**8**	**8**	12	13	12	**11**	15
TR_PSD_3_2	41	8	8	**7**	11	12	12	**11**	15
TR_PSD_4_2	43	9	**8**	**8**	12	12	12	**10**	16
TR_PSD_5_2	46	8	8	**7**	11	12	12	**11**	15
TR_PSD_6_2	47	**7**	**7**	8	11	**11**	**11**	**11**	15
TR_PSD_7_2	48	**8**	**8**	**8**	12	12	12	**11**	15
TR_PSD_8_2	48	**7**	**7**	8	11	**11**	**11**	**11**	15
TR_PSD_9_2	48	8	8	**7**	11	12	12	**10**	16
TR_PSD_10_2	49	**8**	**8**	**8**	11	12	14	**10**	17
TR_PSD_11_2	50	9	**8**	**8**	11	12	12	**10**	16
TR_PSD_12_2	50	**8**	**8**	**8**	12	12	14	**10**	17
TR_PSD_13_2	56	7	7	**6**	8	11	11	**9**	11
TR_BS_1_2	59	8	8	**7**	11	12	12	**9**	14
TR_BS_2_2	68	8	8	**7**	10	12	12	**10**	12
TR_BS_3_2	73	8	8	**7**	9	13	13	**11**	14
TR_BS_4_2	74	8	8	**7**	10	12	12	**11**	15
TR_BS_5_2	75	8	8	**7**	11	**14**	15	17	19
TR_BS_6_2	80	**8**	**8**	**8**	11	11	11	**10**	14
TR_BS_7_2	81	**8**	**8**	**8**	12	12	12	**10**	15
TR_BS_8_2	84	**8**	9	**8**	12	14	**13**	14	19
TR_BS_9_2	88	8	8	**7**	11	11	12	**10**	15
TR_BS_10_2	89	**8**	**8**	**8**	12	12	12	**11**	15
TR_BS_11_2	89	**8**	**8**	**8**	10	13	**12**	**12**	16
TR_BS_12_2	90	**8**	**8**	**8**	12	12	12	**9**	13
TR_BS_13_2	90	9	9	**8**	12	13	13	**12**	16
TR_BS_14_2	94	**8**	**8**	**8**	11	**13**	**13**	**13**	15
TR_BS_15_2	95	8	8	**7**	10	**12**	**12**	**12**	14
TR_BS_16_2	95	8	8	**7**	10	13	13	**12**	15
TR_BS_17_2	96	8	8	**7**	10	13	13	**12**	15
TR_BS_18_2	97	**8**	**8**	**8**	11	13	12	**11**	15
TR_PSD_14_2	97	9	9	**8**	13	14	13	**12**	15
TR_BS_19_2	102	8	8	**7**	10	13	13	**12**	14
TR_BS_20_2	103	8	8	**7**	10	13	13	**11**	14
TR_BS_21_2	124	**9**	10	**9**	14	15	15	**14**	17
TR_BS_22_2	140	9	9	**8**	14	**14**	**14**	15	16
TR_BS_23_2	145	11	10	**9**	16	15	15	**14**	19
TR_BS_24_2	146	10	10	**9**	14	**15**	**15**	**15**	18
TR_BS_25_2	153	**9**	10	**9**	14	15	**14**	**14**	17

Table 5.5.9: Iteration numbers of Algorithm 7 on matrices with $\bar{\kappa} = 0$ and $\bar{\kappa} = 10000$

5.5.2.3 Numerical results obtained for interior-point algorithm acting in a wide neighborhood of the central path

Finally, the results in the case of Algorithm 8 are similar to the results that we obtained with Algorithms 6 and Algorithm 7. While it has slightly worse results than Algorithm 7 previously presented, it shows better results than the proposed Algorithm 6. Finally, we show results in case of Algorithm 8 in the case of problems based on the matrix (5.5.1), with starting point $\mathbf{x} = \mathbf{s} = \mathbf{e}$, where $\mathbf{e}$ is the vector of all ones. This illustrates that LCPs with the Csizmadia matrix can be solved easily by using these algorithms. Furthermore, it is worth noting that in the case of these problems, Algorithm 8 provides better results than Algorithm 7 in this thesis.

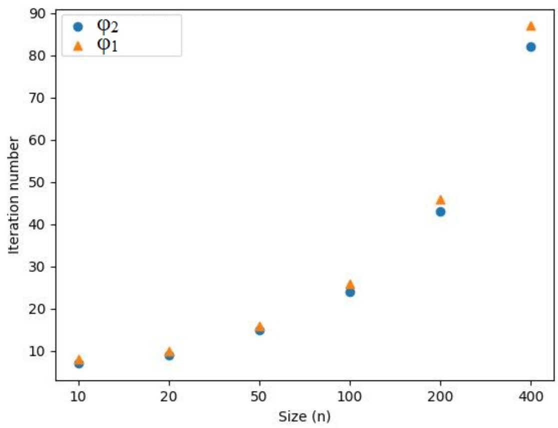


Figure 5.14: TR LCP solver for Algorithm 8 for sufficient LCPs with Csizmadia matrix, and with starting points $\mathbf{x} = \mathbf{s} = \mathbf{e}$.

In Table 5.5.10 we present the iteration numbers of the Algorithm 8 on matrices with $\kappa = 0$ and $\kappa = 10000$. Note that the boldface letter means that it represents the smallest number of iterations for a given matrix.

		$\bar{\kappa} = 0$		$\bar{\kappa} = 10000$	
Name of matrix	Size	φ_1	φ_2	φ_1	φ_2
TR_PSD_1_2	27	**8**	**8**	**13**	**13**
TR_PSD_2_2	32	**9**	**9**	**15**	**15**
TR_PSD_3_2	41	**8**	**8**	**14**	**14**
TR_PSD_4_2	43	10	**9**	**14**	**14**
TR_PSD_5_2	46	9	**8**	14	**13**
TR_PSD_6_2	47	**8**	**8**	**13**	**13**
TR_PSD_9_2	48	9	**8**	14	**13**
TR_PSD_7_2	48	**9**	**9**	**13**	**13**
TR_PSD_8_2	48	**8**	**8**	**14**	**14**
TR_PSD_10_2	49	**9**	**9**	**15**	**15**
TR_PSD_12_2	50	**9**	**9**	**14**	**14**
TR_PSD_11_2	50	**9**	**9**	**15**	**15**
TR_PSD_13_2	56	8	**7**	**12**	13
TR_BS_1_2	59	**9**	**9**	14	**13**
TR_BS_2_2	68	**8**	**8**	**13**	**13**
TR_BS_3_2	73	**8**	**8**	**15**	**15**
TR_BS_4_2	74	**8**	**8**	**14**	**14**
TR_BS_5_2	75	**9**	**9**	**17**	18
TR_BS_6_2	80	**9**	**9**	13	**12**
TR_BS_7_2	81	**10**	**10**	14	**13**
TR_BS_8_2	84	**10**	**10**	**18**	19
TR_BS_9_2	88	**9**	**9**	**13**	**13**
TR_BS_11_2	89	9	**8**	15	**14**
TR_BS_10_2	89	10	**9**	**14**	**14**
TR_BS_13_2	90	10	**9**	15	**14**
TR_BS_12_2	90	**9**	**9**	**13**	**13**
TR_BS_14_2	94	**9**	**9**	15	**14**
TR_BS_15_2	95	**8**	**8**	**14**	**14**
TR_BS_16_2	95	**9**	**9**	**15**	**15**
TR_BS_17_2	96	**9**	**9**	15	**14**
TR_BS_18_2	97	9	**8**	**16**	**16**
TR_PSD_14_2	97	**10**	**10**	**14**	**14**
TR_BS_19_2	102	9	**8**	15	**14**
TR_BS_20_2	103	9	**8**	**14**	**14**
TR_BS_21_2	124	**11**	**11**	**17**	**17**
TR_BS_22_2	140	**10**	**10**	17	**16**
TR_BS_23_2	145	**11**	**11**	**18**	**18**
TR_BS_24_2	146	**11**	**11**	**17**	**17**
TR_BS_25_2	153	11	**10**	**17**	**17**

Table 5.5.10: Iteration numbers of Algorithm 8 on matrices with $\bar{\kappa} = 0$ and $\bar{\kappa} = 10000$

To conclude, we showed the implemented versions of the IPAs and we pointed out the differences between them and the theoretical IPAs. We presented our numerical results based on two test sets that we generated during our research. Based on our experiments we observed that in case of the matrices with larger κ obtained by rescaling the test set problems, there was no significant increase in the number of steps. This indicates that in some cases the value κ is not as significantly influential as expected from theoretical analyses. It would be interesting to generate other test set of problems, where the matrices cannot be scaled to ones with $\bar{\kappa} = 0$. As further research, it would be interesting to test our IPAs on $P_*(\kappa)$-LCPs with these types of matrices, as well. In the following chapter we present some concluding remarks and further research topics.

Chapter 6

Conclusions and further research topics

In this chapter, we summarize the thesis and suggest further research topics in the field of interior-point algorithms (IPAs) for solving linear complementarity problems (LCPs).

In this thesis, we introduced new classes of algebraically equivalent transformation (AET) functions for determining search directions for both primal-dual and predictor-corrector IPAs. We also showed that the kernel functions corresponding to some of these AET functions do not belong to the known classes of kernel functions from the literature. Furthermore, we also gave an example of such an eligible kernel function, for which the corresponding AET function is not a member of our classes of AET functions. We introduced three different IPAs, a primal-dual, and two predictor-corrector IPAs for solving $P_*(\kappa)$-LCPs. These algorithms are Algorithm 1, 2, and 5. We also presented the complexity analyses of the proposed IPAs. Furthermore, numerical results were also provided.

First, in Chapter 1, we proposed a literature review about previous research results in the topic of LCPs and IPAs. We presented the AET technique in the case of LCPs. In Section 1.5, we also presented other ways for determining search directions in the case of IPAs. In Section 1.6, we showed several applications of LCPs.

We presented our theoretical results in Chapters 2, 3 and 4. First, in Section 2.1, we proposed a new class of AET functions. After that, in Section 2.2, we introduced the primal-dual IPA, Algorithm 1, that is based on this class of AET functions. Finally, to conclude Chapter 2, we presented the complexity analysis of Algorithm 1 in Section 2.3.

In Chapter 3, we defined the new class of AET functions that defines a predictor-corrector IPA. In Section 3.1, we introduced this class of AET functions and presented the complexity analysis of the predictor-corrector IPA, Algorithm 2, that uses a function from this class of AET functions as an input. In Section 3.2, we presented a special function that is a member of this class of AET functions and has an inflection point. We also showed the main steps of the complexity analysis of the IPA in this special case.

Finally, to conclude our theoretical results, we defined a new type of predictor-corrector IPA in Chapter 4. In Section 4.1 we introduced a generalized wide neighborhood based on the results of Potra and Liu [88]. Based on the generalized wide neighborhood, we introduced a new type of predictor-corrector IPA, Algorithm 4 that uses $\varphi(t) = \sqrt{t}$ to define the wide neighborhood of the central path in Section 4.2. After that, in Section 4.3 we showed the complexity analysis of Algorithm 4.

In Chapter 5, we proposed an implemented version for each type of IPA presented in Chapters 2, 3 and 4. Furthermore, we showed numerical methods for solving LCPs with sufficient matrices. In Section 5.1, we proposed algebraically equivalent Newton systems that are equivalent in theory, although in practice they work differently. We also suggested a method to calculate the Newton directions from the Newton systems. After calculating the directions, we suggested a method for calculating the step length in each IPA which is different than the method used in theory. In Section 5.3, we proposed a method to check the feasibility of the new iterates. Moreover, we also presented the applied neighborhoods. Furthermore, we showed some connections between the decrease of $\kappa(\Delta\mathbf{x})$ and the change in the step length α, duality gap and $\|\Delta\mathbf{x}\Delta\mathbf{s}\|$ in each iteration. This connection can be important in theory, since in all of the cases the theoretical iteration number depends on κ. This leads to new interesting research topics. Finally, in Section 5.5, we presented our numerical results. We generated test cases for which we have a large κ. Furthermore, we also considered a special matrix known from the literature which has exponential handicap. We obtained promising results in all cases. This indicates that in some cases the value of κ is not as significantly influential as expected from theoretical analyses.

Finally, we also discuss plans for further research. It is important to mention that, in theory, we would like to extend the predictor-corrector IPA that works in a wide neighborhood of the central path that is based on other AET functions as well. We would like to extend our class of AET functions in the case of the primal-dual IPA and the predictor-corrector IPA for solving general LCPs like Illés et al. did in [50, 51]. Furthermore, as we have already mentioned, we would like to further investigate the effect of the handicap on the numerical results. Finally, our aim is to further develop our solver for more general matrices.

Bibliography

[1] M. Achache. A new primal-dual path-following method for convex quadratic programming. *Computational and Applied Mathematics*, 25(1), (2006), 97–110.

[2] M. Achache. Complexity analysis and numerical implementation of a short-step primal-dual algorithm for linear complementarity problems. *Applied Mathematics and Computation*, 216(7), (2010), 1889–1895.

[3] W. Ai and S. Zhang. An O($\sqrt{n}L$) Iteration Primal-dual Path-following Method, Based on Wide Neighborhoods and Large Updates, for Monotone LCP. *SIAM Journal on Optimization*, 16(2), (2005), 400-417.

[4] A. Akkeleş, L. Balogh, and T. Illés. New variants of the criss-cross method for linearly constrained convex quadratic programming. *European Journal of Operational Research*, 157(1), (2004), 74-86.

[5] K. Arrow and G. Debreu. Existence of a competitive equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, (1954).

[6] S. Asadi and H. Mansouri. A path-following algorithm for $P_*(\kappa)$-horizontal linear complementarity problem based on Darvay's directions. In *Proceeding of the 43rd Annual Iranian Mathematics Conference, Tabriz University, Tabriz, Iran*, (2012), 861–864.

[7] S. Asadi and H. Mansouri. Polynomial interior-point algorithm for $P_*(\kappa)$ horizontal linear complementarity problems. *Numerical Algorithms*, 63(2), (2013), 385–398.

[8] S. Asadi, M. Zangiabadi, and H. Mansouri. An infeasible interior-point algorithm with full-newton steps for $P_*(\kappa)$ horizontal linear complementarity problems based on a kernel function. *Journal of Applied Mathematics and Computing*, 50(1), (2016), 15-37.

[9] S. Asadi, H. Mansouri, Zs. Darvay, and M. Zangiabadi. On the $P_*(\kappa)$ horizontal linear complementarity problems over Cartesian product of symmetric cones. *Optimization Methods and Software*, 31, (2016), 233–257.

[10] S. Asadi, H. Mansouri, and Zs. Darvay. An infeasible full-NT step IPM for $P_*(\kappa)$ horizontal linear complementarity problem over Cartesian product of symmetric cones. *Optimization*, 66(2), (2017), 225–250.

[11] S. Asadi, H. Mansouri, G. Lesaja, and M. Zangiabadi. A long-step interior-point algorithm for symmetric cone Cartesian $P_*(\kappa)$-HLCP. *Optimization*, 67(11), (2018), 2031–2060.

[12] S. Asadi, N. Mahdavi-Amiri, Zs. Darvay, and P.R. Rigó. Full-NT step feasible interior-point algorithm for symmetric cone horizontal linear complementarity problem based on a positive-asymptotic barrier function. *Optimization Methods and Software*, 37, (2022), 192-213.

[13] Y.Q. Bai, M. El Ghami, and C. Roos. A new efficient large-update primal-dual interior-point method based on a finite barrier. *SIAM Journal on Optimization*, 13, (2003), 766-782.

[14] 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 Journal on Optimization*, 15(1), (2004), 101-128.

[15] C. Brás, G. Eichfelder, and J. Júdice. Copositivity tests based on the linear complementarity problem. *Computational Optimization and Applications*, 63, (2016), 461-493.

[16] S.J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60(3), (1989), 393–399.

[17] B. Codenotti, A. Saberi, K. Varadarajan, and Y. Ye. The complexity of equilibria: Hardness results for economies via a correspondence with games. *Theoretical Computer Science*, 408(2-3), (2008) 188–198.

[18] R.W. Cottle, J.S. Pang, and V. Venkateswaran. Sufficient matrices and the linear complementarity problem. *Linear Algebra and Its Applications*, 114, (1989), 231–249.

[19] R.W. Cottle, J.S. Pang, and R.E. Stone. The Linear Complementarity Problem. *Computer Science and Scientific Computing*, Academic Press, Boston, (1992).

[20] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13, (1951), 339-347.

[21] Zs. Darvay. A new algorithm for solving self-dual linear optimization problems. *Studia Universitatis Babes-Bolyai, Series Informatica*, 47, (2002), 15-26.

[22] Zs. Darvay. New interior point algorithms in linear programming. *Advanced Modeling and Optimization*, 5(1), (2003), 51–92.

[23] Zs. Darvay. A new predictor-corrector algorithm for linear programming. *Alkalmazott Matematikai Lapok*, 22, (2005), 135-161. in Hungarian.

[24] Zs. Darvay, I.-M. Papp, and P.-R. Takács. Complexity analysis of a full-Newton step interior-point method for linear optimization. *Periodica Mathematica Hungarica*, 73(1), (2016), 27-42.

[25] Zs. Darvay, T. Illés, J. Povh, and P. R. Rigó. Feasible corrector-predictor interior-point algorithm for $P_*(\kappa)$-linear complementarity problems based on a new search direction. *SIAM Journal on Optimization*, 30(3), (2020), 2628-2658.

[26] Zs. Darvay, T. Illés, and Cs. Majoros. Interior-point algorithm for sufficient LCPs based on the technique of algebraically equivalent transformation. *Optimization Letters*, 15, (2021), 357-376.

[27] Zs. Darvay, T. Illés, and P. R. Rigó. Predictor-corrector interior-point algorithm for $P_*(\kappa)$-linear complementarity problems based on a new type of algebraic equivalent transformation technique. *European Journal of Operational Research*, 298(1), (2022), 25-35.

[28] Zs. Darvay and P. R. Rigó. New predictor-corrector algorithm for symmetric cone horizontal linear complementarity problems. *Journal of Optimization Theory and Applications*, 202(1), (2024), 50–75.

[29] Zs. Darvay and P. R. Rigó. Interior-point algorithms for symmetric cone horizontal linear complementarity problems based on a new class of algebraically equivalent transformations. *Optimization Letters*, 18(2), (2024), 615-634.

[30] Zs. Darvay, M.E. -Nagy, G. Lešaja, P.R. Rigó, and A. Varga. Comprehensive Analysis of Kernel-Based Interior-Point Methods for $P_*(\kappa)$-LCP. *Corvinus Economics Working Papers*, 2024/03 (2024).

[31] E. de Klerk, C. Roos, and T. Terlaky. Polynomial primal-dual affine scaling algorithms. *Journal of Combinatorial Optimization*, 2(1), (1998), 51-69.

[32] E. de Klerk. Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications. *Springer Science and Business Media*, 65, (2006).

[33] E. de Klerk and M. E.-Nagy. On the complexitiy of computing the handicap of a sufficient matrix. *Mathematical Programming*, 129, (2011), 383-402.

[34] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Doklady Academii Nauk SSSR, 174:747–748, 1967. (In Russian). Translated in Soviet Mathematics Doklady*, 8, (1967), 674-675.

[35] B. C. Eaves. A finite algorithm for the linear exchange model. *Journal of Mathematical Economics*, 3, (1976), 197–203.

[36] E. Eisenberg and D. Gale. Consensus of subjective probabilities: The pari-mutuel method. *Annals Of Mathematical Statistics*, 30, (1959), 165–168.

[37] E. Eisenberg. Aggregation of utility functions. *Management science*, 7(4), (1961), 337–350.

[38] H. Fang, X. Chen, and M. Fukushima. Stochastic R_0 matrix linear complementarity problems. *SIAM Journal on Optimization*, 18(2), (2007), 482-506.

[39] M.C. Ferris and J.S. Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39(4), (1997), 669-713.

[40] I. Fisher. Mathematical Investigations in the Theory of Value and Prices. *Yale University, PhD thesis*, (1892).

[41] K. R. Frisch. The multiplex method for linear programming. *Sankhyâ: The Indian Journal of Statistics*, 18(3/4), (1957), 329–362.

[42] D. Gale. The Theory of Linear Economic Models. McGraw Hill, New York, (1960).

[43] M. El Ghami, I. Ivanov, J.B. Melissen, C. Roos, and T. Steihaug. A polynomial-time algorithm for linear optimization based on a new class of kernel functions. *Journal of Computational and Applied Mathematics*, 224, (2009), 500-513.

[44] F. Gurtuna, C. Petra, F.A. Potra, O. Shevchenko, and A. Vancea. Corrector-predictor methods for sufficient linear complementarity problems. *Computational Optimization and Applications*, 48(3), (2011), 453-485.

[45] S.M. Guu and R.W. Cottle. On a subclass of P_0. *Linear Algebra and its Applications*, 223/224, (1995), 325–335.

[46] M. Haddou, T. Migot, and J. Omer. A generalized direction in interior point method for monotone linear complementarity problems. *Optimization Letters*, 13(1), (2019), 35-53.

[47] T. Illés, J. Peng, and C. Roos A strongly polynomial rounding procedure yielding a maximally complementary solution for P*(κ) linear complementarity problems. *SIAM Journal on Optimization*, 11(2), (2000), 320–340.

[48] T. Illés and M. Nagy. A Mizuno-Todd-Ye type predictor-corrector algorithm for sufficient linear complementarity problems. *European Journal of Operational Research*, 181(3), (2007), 1097-1111.

[49] T. Illés and M. Nagy. EP theorem for dual linear complementarity problems. *Journal of Optimization Theory and Applications*, 140(2), (2009), 233–238.

[50] T. Illés, M. Nagy, and T. Terlaky. A polynomial path-following interior point algorithm for general linear complementarity problems. *Journal of Global Optimization*, 47(3), (2010), 329-342.

[51] T. Illés, M. Nagy, and T. Terlaky. Polynomial Interior Point Algorithms for General Linear Complementarity Problems. *Algorithmic Operations Research*, 5(1), (2010), 1-12.

[52] T. Illés and S. Morapitiye. Generating sufficient matrices. In F. Friedler, editor, *8th VOCAL Optimization Conference: Advanced Algorithms*, Pázmány Péter Catholic University, Budapest, Hungary, (2018), 56-61.

[53] T. Illés, P.R. Rigó, and R. Török. Implementation of primal-dual interior-point algorithm for solving sufficient linear complementarity problems. In S. Drobne, L. Zadnik Stirn,

M. Kljajić Borštar, J. Povh, and J. Žerovnik, editors, *Proceedings of the 16th International Symposium on Operational Research in Slovenia: SOR'21*, Slovenia, (2021), 86-91.

[54] T. Illés, P.R. Rigó, and R. Török. Predictor-corrector interior-point algorithm based on a new search direction working in a wide neighbourhood of the central path. *Corvinus Economics Working Papers*, 2021/02, (2021).

[55] T. Illés, P.R. Rigó, and R. Török. New predictor-corrector interior-point algorithm with AET function having inflection point. *Corvinus Economics Working Papers*, 2022/05, (2022).

[56] T. Illés, P.R. Rigó, and R. Török. Unified approach of primal-dual interior-point algorithms for a new class of AET functions. *Corvinus Economics Working Papers*, 2022/02, (2022).

[57] T. Illés, P.R. Rigó, and R. Török. Large-step predictor-corrector interior point method for sufficient linear complementarity problems based on the algebraic equivalent transformation. *EURO Journal on Computational Optimization*, 11(1), (2023).

[58] T. Illés, P.R. Rigó, and R. Török. Unified approach of interior-point algorithms for $\mathcal{P}_*(\kappa)$-LCPs using a new class of algebraically equivalent transformations. *Journal of Optimization Theory and Applications*, 202(1), (2023), 1-23.

[59] T. Illés, P.R. Rigó, and R. Török. New predictor-corrector interior-point algorithm with AET function having inflection points. *Communications in Optimization Theory*, 11, (2024), 1-21.

[60] T. Illés, P.R. Rigó, and R. Török. Predictor-corrector interior-point algorithm for sufficient linear complementarity problems based on AET functions with bounded proportional growth rate. (2025), submitted.

[61] K. Jain. A polynomial time algorithm for computing an Arrow–Debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1), (2007), 303–318.

[62] N. Karmarkar. A new polynomial-time algorithm for linear programming. *In Proceedings of the sixteenth annual ACM symposium on Theory of computing*, (1984), 302-311.

[63] W. Karush. Minima of functions of several variables with inequalities as side constraints *Master's thesis*, Department of Mathematics, University of Chicago, Illinois, (1939).

[64] L. G. Khachiyan. A polynomial algorithm in linear programming (english translation). *In Soviet Mathematics Doklady*, 20, (1979), 191-194.

[65] B. Kheirfam. A predictor-corrector interior-point algorithm for $P_*(\kappa)$-horizontal linear complementarity problem. *Numerical Algorithms*, 66(2), (2014), 349-361.

[66] B. Kheirfam and M. Haghighi. A full-Newton step feasible interior-point algorithm for $P_*(\kappa)$-LCP based on a new search direction. *Croatian Operational Research Review*, 7(2), (2016), 277-290.

[67] V. Klee and G. J. Minty. How good is the simplex algorithm? *Inequalities*, 3(3), (1972), 159-175.

[68] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44, (1989), 1-26.

[69] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems. *Springer* Verlag, Berlin, Germany, 538, (1991).

[70] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. A unified approach to interior point algorithms for linear complementarity problems: A summary. *Operations Research Letters*, 10.5, (1991), 247-254.

[71] C. E. Lemke and J. T. Howson. Equilibrium points of bimatrix games. *SIAM Journal on Applied Mathematics*, 12, (1964), 413-423.

[72] G. Lešaja and C. Roos. Unified analysis of kernel-based interior-point methods for $P_*(\kappa)$-linear complementarity problems. *SIAM Journal on Optimization*, 20(6), (2010), 3014-3039.

[73] G. Lešaja, G.Q. Wang, and D.T. Zhu. Interior-point methods for Cartesian $P_*(\kappa)$-linear complementarity problems over symmetric cones based on the eligible kernel functions. *Optimization Methods and Software*, 27, (2012), 827-843.

[74] Z.Q. Luo, N. Xiu. Path-following interior point algorithms for the Cartesian $P_*(\kappa)$-LCP over symmetric cones. *Science in China Series A-Mathematics*, 52, (2009), 1769–1784.

[75] H. Mansouri and M. Pirhaji. A polynomial interior-point algorithm for monotone linear complementarity problems. *Journal of Optimization Theory and Applications*, 157(2), (2013), 451-461.

[76] N. Megiddo. Pathways to the Optimal Set in Linear Programming. *Springer*, New York, (1989), 131–158.

[77] M. E.-Nagy. Interior point algorithms for general linear complementarity problems. *Eötvös Loránd University, PhD thesis*, (2009).

[78] M. E.-Nagy and T. Illés. Market exchange models and geometric programming. *Central European Journal of Operations Research*, 27, (2019), 415-435.

[79] M. E.-Nagy and A. Varga. A new long-step interior point algorithm for linear programming based on the algebraic equivalent transformation. *Central European Journal of Operations Research*, 31, (2022), 691–711.

[80] M. E.-Nagy and A. Varga. A new Ai–Zhang type interior point algorithm for sufficient linear complementarity problems. *Journal of Optimization Theory and Applications*, 202, (2024) 76–107.

[81] M. E.-Nagy, T. Illés, J. Povh, A. Varga, and J. Žerovnik. Sufficient matrices: properties, generating and testing. *Journal of Optimization Theory and Applications*, 202, (2024), 204–236.

[82] Y. Nesterov and A. Nemirovskii. Interior-point polynomial algorithms in convex programming. *SIAM Publications*, (1994).

[83] Netlib test collection: https://www.netlib.org/

[84] S. Pan, X. Li, and S. He. An infeasible primal-dual interior-point algorithm for linear programs based on logarithmic equivalent transformation. *Journal of Mathematical Analysis and Applications*, 314(2), (2006), 644-660.

[85] M. F. Pang. Engineering and economic applications of complementarity. *SIAM Review*, 39(4), (1997), 669-713.

[86] J. Peng, C. Roos, and T. Terlaky. Self-Regular Functions: a New Paradigm for Primal-Dual Interior-Point Methods. *Princeton University Press*, (2002).

[87] F. Potra. A superlinearly convergent predictor-corrector method for degenerate LCP in a wide neighborhood of the central path with-iteration complexity. *Mathematical Programming*, 100(2), (2004) 317-337.

[88] F. A. Potra and X. Liu. Predictor-corrector methods for sufficient linear complementarity problems in a wide neighborhood of the central path. *Optimization Methods and Software*, 20(1), (2005), 145-168.

[89] F. A. Potra. Corrector-predictor methods for monotone linear complementarity problems in a wide neighborhood of the central path. *Mathematical Programming*, 111, (2008), 243-272.

[90] F. A. Potra. Interior point methods for sufficient horizontal LCP in a wide neighborhood of the central path with best known iteration complexity. *SIAM Journal on Optimization*, 24(1), (2014) 1-28.

[91] P. R. Rigó and Zs. Darvay. Infeasible interior-point method for symmetric optimization. *Computational Optimization and Applications*, 71(2), (2018), 483-508.

[92] P. R. Rigó. New trends in algebraic equivalent transformation of the central path and its applications. *Budapest University of Technology and Economics, PhD thesis*, (2020).

[93] C. Roos, T. Terlaky, and J.-Ph. Vial. Theory and Algorithms for Linear Optimization. An Interior Approach. *John Wiley & Sons, Chichester*, UK, (1997).

[94] S.H. Schmieta and F. Alizadeh. Extension of primal-dual interior point algorithms to symmetric cones. *Mathematical Programming*, 96(3), (2003), 409–438.

[95] E. Sloan and O.N. Sloan. Quitting games and linear complementarity problems. *Mathematical Methods of Operations Research*, 45(2), (2020), 434-454.

[96] Gy. 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. *Springer* Verlag, Berlin, WestGermany, (1986), 866–876.

[97] J. Tang, L. Dong, L. Fang, and J. Zhou. A weighted-path-following interior-point algorithm for second-order cone optimization. *Applied Mathematics and Information Sciences*, 9(2), (2015), 973.

[98] T. Terlaky and E. Klafszky. Some generalizations of the Criss-Cross method for quadratic programming. *Optimization*, 24(1-2), (1992), 127-139.

[99] T. Terlaky. An easy way to teach interior-point methods. *European Journal of Operational Research*, 130(1), (2001), 1-19.

[100] R. Török. LCP test examples used for numerical results. https://sites.google.com/view/lcp-test-examples-roland-trk/

[101] H. Väliaho. P_*-matrices are just sufficient. *Linear Algebra and its Applications*, 239, (1996), 103-108.

[102] H. Väliaho. Determining the handicap of a sufficient matrix. *Linear Algebra and its Applications*, 253, (1997), 279-298.

[103] A. Varga. New Ai-Zhang type long-step interior point algorithms using the algebraically equivalent transformation technique. *Budapest University of Technology and Economics, PhD thesis*, (2022).

[104] L. Walras. Elements of Pure Economics, or the Theory of Social Wealth *(1899, 4th ed.; 1926, rev ed., 1954, Engl. Transl.)*, (1874).

[105] G. Q. Wang and Y. -Q. Bai. A primal-dual interior-point algorithm for second-order cone optimization with full Nesterov–Todd step. *Applied Mathematics and Computation*, 215(3), (2009), 1047-1061.

[106] W. L. Winston. Operations Research. Applications and Algorithms. *Duxbury Press, Belmont*, (1987).

[107] S.J. Wright. Primal-Dual Interior-Point Methods. *SIAM*, Philadelphia, USA, (1997).

[108] Y. Ye and E. Tse. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, 44(1), (1989), 157-179.

[109] Y. Ye. Interior Point Algorithms, Theory and Analysis. *John Wiley & Sons*, Chichester, UK, (1997).

[110] Y. Ye. Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. *Theoretical Computer Science*, 378.2, (2007) 134-142.

[111] Y. Ye. A path to the Arrow-Debreu competitive market equilibrium. *Mathematical Programming*, 111(1-2), (2008), 315-348.

[112] D. B. Yudin and A. S. Nemirovskii. Information complexity and efficient methods for solving convex extremal problems. *Nauka*, Moscow, USSR, (1972).