

## Improved Full-Newton-Step Infeasible Interior-Point Method for Linear Complementarity Problems\*

Goran Lešaja<sup>1,†</sup> and Mustafa Ozen<sup>1</sup>

<sup>1</sup> *Department of Mathematical Sciences, Georgia Southern University  
Statesboro, GA 30460, USA*

*E-mail: {goran, mo01520}@georgiasouthern.edu*

**Abstract.** We present an Infeasible Interior-Point Method for monotone Linear Complementarity Problem (*LCP*) which is an improved version of the algorithm given in [13]. In the earlier version, each iteration consisted of one feasibility step and few centering steps. The improved version guarantees that after one feasibility step, the new iterate is feasible and close enough to the central path thanks to the much tighter proximity estimate which is based on the new lemma introduced in [18]. Thus, the centering steps are eliminated. Another advantage of this method is the use of full-Newton-steps, that is, no calculation of the step size is required. The preliminary implementation and numerical results demonstrate the advantage of the improved version of the method in comparison with the old one.

**Key words:** linear complementarity problems, interior-point method, infeasible interior-point method, full-Newton-step

Invited paper, received: March 01, 2016; accepted: March 31, 2016; available online:  
April 04, 2016

DOI:10.17535/crorr.2016.0001

---

### 1. Introduction

In this paper, we consider a class of Linear Complementarity Problems (*LCP*) in the standard form:

$$\begin{aligned} s &= Mx + q \\ xs &= 0 \\ x &\geq 0, s \geq 0 \end{aligned} \tag{1}$$

where  $x, s \in \mathbb{R}^n$ ,  $M \in \mathbb{R}^{n \times n}$ ,  $q \in \mathbb{R}^n$  and  $xs$  denotes Hadamard (component-wise) product of vectors  $x$  and  $s$ .

*LCP* is not an optimization problem, but it has robust relationship with important optimization problems such as linear programming (*LP*) and quadratic programming (*QP*) problems. This strong relationship is based on the fact that

---

\*The paper is dedicated to Professor Luka Neralić on the occasion of his 70th Birthday.

†Corresponding author.

Karush-Kuhn-Tucker (KKT) optimality conditions for  $LP$  and  $QP$  can be converted into  $LCP$ . Furthermore, many problems from, game theory, engineering, finance, transportation, etc. can be directly formulated as  $LCP$ . Therefore, efficient methods for solving  $LCPs$  has been an important area of research in optimization both from theoretical and practical point of view. For a comprehensive treatment of  $LCP$  theory and practice we refer the reader to the monographs of Cottle et al. [4], Fachinei and Pang [6] and Kojima et al. [11].

It is well known that for general matrices  $M$ ,  $LCP$  is NP-complete [3]. Hence, we consider classes of matrices  $M$  for which the corresponding  $LCP$  can be solved in polynomial time. Most common and most studied is the class of monotone-LCPs, where the matrix  $M$  is a positive-semidefinite matrix. This is largely due to the fact that the Karush-Kuhn-Tucker conditions (KKT) of the convex  $QP$  and  $LP$  problems can be formulated as *monotone* –  $LCP$ . In addition, many practical problems that can be directly formulated as  $LCP$  are usually *monotone* –  $LCP$ . For these reasons, in this paper we consider a class of *monotone* –  $LCP$ .

Methods to solve  $LCPs$  have traditionally been connected to the methods developed for  $LP$ . The generalization of the Simplex method, the Lemke's method, was developed soon after the introduction of the Simplex Method. The trend continued with other simplex-type (pivot based) methods. The tradition continued after the development of Interior-Point Methods ( $IPMs$ ) that has revolutionized the area of both linear and nonlinear, primarily convex, optimization.

The first  $IPM$  for  $LP$  was proposed by Karmarkar [9] in 1984. The main idea of the algorithm differs from Simplex Method. It uses projective transformations and Karmarkar's potential function. Moreover,  $IPMs$  are iterative methods and iterates are calculated in the interior of the feasible region. In 1986, it was proved that the Karmarkar's algorithm is connected to barrier and Newton-type methods [8]. Soon after in 1988, Renegar [17] developed the first path following Newton-type  $IPM$  for  $LP$ . These achievements motivated the development of Newton-based  $IPMs$ . Since then, many different versions of Newton-based  $IPMs$  for  $LP$  have been proposed. Many of these  $IPMs$  for  $LP$  have been generalized for  $LCPs$ , the first one being proposed by Kojima et al. [10] in 1989. Without any attempt to being complete, see [1, 2, 13, 14, 15, 19] and the references therein for more information on the development of  $IPMs$ .

Interior-Point Methods can be classified into two groups: feasible  $IPMs$  and infeasible  $IPMs$ . Feasible  $IPM$  requires a strictly feasible starting point and feasibility of each iterates. In real life, it is not easy to find a feasible starting point all the time and finding the feasible starting point may be as difficult as solving problem itself. Hence, it is important to consider infeasible  $IPM$  which can solve problems with infeasible starting points. The algorithm we present in this paper belongs to the second group of  $IPMs$ . In addition, we consider full-Newton-step  $IPM$ , that is, the step size is always one.

In the paper, an improvement of the algorithm given in [13] is proposed. In the old version of the algorithm, each iteration requires two main steps per iteration, a feasibility step and few centering steps (at most two). By suitable choice of parameters, it is possible to eliminate centering steps altogether and just keep the feasibility step at each iteration thanks to the much tighter proximity estimate which is based

on the Lemma 3 first proposed in [18]. The algorithm is still globally convergent and iteration bound matches the best known iteration bound for these types of *IPM*.

The paper is organized as follows: The outline of the algorithm is given in Section 2 while the analysis and complexity of the algorithm are discussed in Section 3 and Section 4 respectively. In Section 5, numerical results are presented. Concluding remarks are given in Section 6.

## 2. Algorithm

In this section, an outline of the improved full-Newton-step infeasible *IPM* for monotone-*LCP* (1) is presented.

It is assumed that the *LCP* (1) has a solution  $(x^*, s^*)$  such that

$$\|x^*\|_\infty \leq \gamma_p, \quad \|s^*\|_\infty \leq \gamma_d \quad \text{and} \quad \mu^0 = \frac{(x^0)^T s^0}{n}. \quad (2)$$

The assumption basically states that the solution is in within a certain (large) box. This assumption is necessary for theoretical purposes of proving the global convergence of the method. However, in practice the assumption is not restrictive; if the solution is not found for a certain choice of constants  $\gamma_p$  and  $\gamma_d$ , they can be increased. If after few increases the solution is still not found, then we conclude that the solution does not exist or it is too big to have any practical meaning.

For a positive random starting point  $(x^0, s^0)$  and for any  $\nu$  with  $0 < \nu \leq 1$ , we consider a *perturbed LCP*, denoted by  $LCP_\nu$ .

$$\begin{aligned} s - Mx - q &= \nu r^0 \\ xs &= 0 \\ x \geq 0, s &\geq 0 \end{aligned} \quad (3)$$

where  $r^0 = s^0 - Mx^0 - q$  is called *residual*. Note that for  $\nu = 1$   $LCP_\nu$  has an obvious strictly feasible point, the initial starting point  $(x^0, s^0)$ . Often, the starting point is chosen as

$$x^0 = \gamma_p e, \quad s^0 = \gamma_d e$$

where  $e$  denotes an  $n$  dimensional vector of ones.

The following lemma, proof of which can be found in [16], shows how *LCP* and  $LCP_\nu$  are connected.

**Lemma 1.** *The original problem (1) is feasible if and only if the  $LCP_\nu$  (3) is feasible for  $0 < \nu \leq 1$ .*

Since *IPMs* are Newton-based methods, the standard procedure is to perturb the complementarity condition  $xs = 0$  and replace it by  $xs = \mu e$ , with positive parameter  $\mu$ . Then the system (3) becomes

$$\begin{aligned} s - Mx - q &= \nu r^0 \\ xs &= \mu e \\ x \geq 0, s &\geq 0. \end{aligned} \quad (4)$$

It is well known that if matrix  $M$  is positive definite, then the system (4) has a unique solution for each  $\mu > 0$ . These solutions, denoted by  $(x(\nu, \mu), s(\nu, \mu))$ , are called  $\mu$ -centers of  $LCP_\nu$  and the set of all  $\mu$ -centers is called *central path*. The general idea of *IPM* is to trace the central path by gradually reducing  $\mu$  to zero. However, tracing central path exactly would be inefficient. One of the main achievements of *IPM* is to show that it is sufficient to trace the central path approximately. As long as the iterates are in a certain neighborhood of the central path it is still possible to prove global convergence and, moreover, achieve polynomial complexity. In addition, we are not interested in solving  $LCP_\nu$ , we are interested in solving the original  $LCP$ , which is  $LCP_\nu$  with  $\nu = 0$ . Thus, the idea is to simultaneously reduce both  $\mu$  and  $\nu$  to zero, that is to work on feasibility and optimality at the same time.

The measure of proximity to the central path is given by the norm-based measure

$$\delta(x, s; \mu) = \delta(v) = \frac{1}{2} \|v - v^{-1}\|, \quad \text{where } v = \sqrt{\frac{xs}{\mu}}. \quad (5)$$

Note that  $\delta(x, s; \mu) = 0$  means  $(x, s)$  is a  $\mu$ -center. It is easy to show that when  $\nu = 1$ , the initial pair  $(x^0, s^0)$  is a  $\mu^0$ -center. Namely, the pair  $(x^0, s^0)$  is strictly feasible solution for  $LCP_{\nu=1}$  and  $\delta(x^0, s^0; \mu^0) = 0$ .

In what follows the main idea of one iteration of the algorithm is described. We assume that at the start of each iteration,  $\delta(x, s; \mu) \leq \tau$  for some threshold value  $\tau > 0$ . As we already noted above, when  $\mu = \mu^0$  and  $\nu = 1$ ,  $(x^0, s^0)$  is a  $\mu$ -center of  $LCP_\nu$ . Thus, initially we have  $\delta(x^0, s^0; \mu^0) = 0 < \tau$  which satisfies our assumption at the first iteration. We also assume that  $\mu$  and  $\nu$  are connected as follows:  $\nu = \mu/\mu^0$ .

Suppose that for  $\mu \in (0, \mu^0]$ , we have an iterate  $(x, s)$  feasible to the system (4) and  $\delta(x, s; \mu) \leq \tau$ . Then, we reduce  $\mu$  to  $\mu^+ = (1 - \theta)\mu$  and  $\nu^+ = \mu^+/\mu^0 = (1 - \theta)\nu$  using barrier parameter  $\theta \in [0, 1)$  and find the new iterate  $(x^+, s^+)$  satisfying (4) with  $\mu^+$  and  $\nu^+$ .

In the old version of the algorithm, the new iterate was found after one feasibility step followed by few centering steps to satisfy  $\delta(x^+, s^+; \mu^+) \leq \tau$ . The centering steps were necessary because it was not guaranteed that the feasibility step is in the  $\tau$ -neighborhood of the central path. In the new version, with the appropriate choice of the threshold parameter  $\tau$  and barrier parameter  $\theta$  and using Lemma 3 it is possible to show tighter proximity estimate which in turns guarantees that the feasibility step is in the  $\tau$ -neighborhood of the central path, hence, eliminating the need for centering steps. The calculation of the feasibility step is given below.

Figures 1 and 2 present graphical representations of one iteration of the old and new versions of the algorithm respectively.

The main part of one iteration of the algorithm consists of finding a new iterate after the reduction of  $\mu$  and  $\nu$ . Let  $(x, s)$  be a starting iterate in the  $\tau$ -neighborhood of the central path of  $LCP_\nu$ . Our goal is to find a strictly feasible point  $(x^f, s^f)$  that is in  $\tau$ -neighborhood of the central path of  $LCP_{\nu^+}$ . First, we need to find search directions  $\Delta^f x$  and  $\Delta^f s$ . A direct application of the Newtons method to the system (4) leads to the following Newton system for the search directions

$$\begin{aligned} M \Delta^f x - \Delta^f s &= \theta \nu r^0 \\ s \Delta^f x + x \Delta^f s &= (1 - \theta) \mu e - xs. \end{aligned} \quad (6)$$

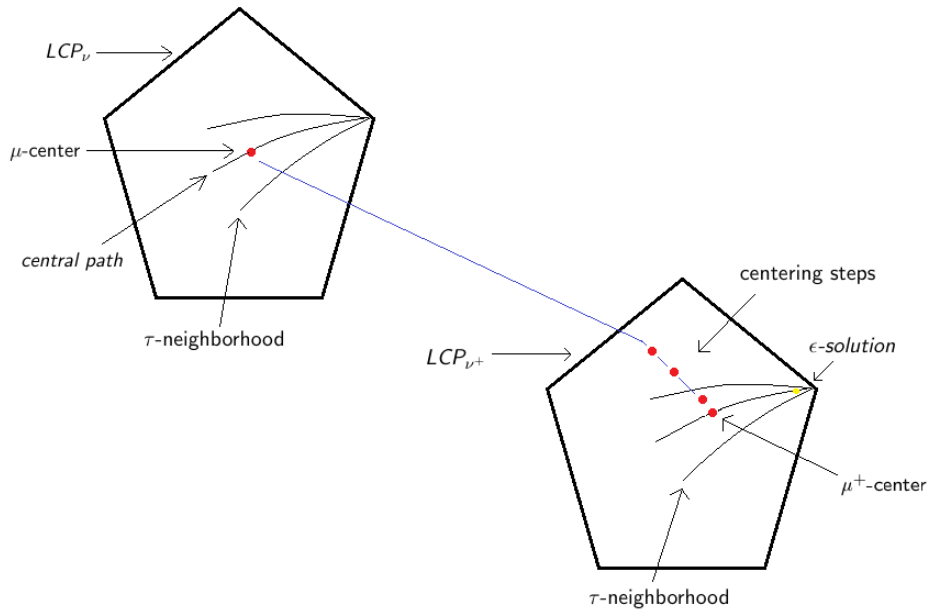


Figure 1: Graphical representation of the old version of the algorithm for  $LCP_v$

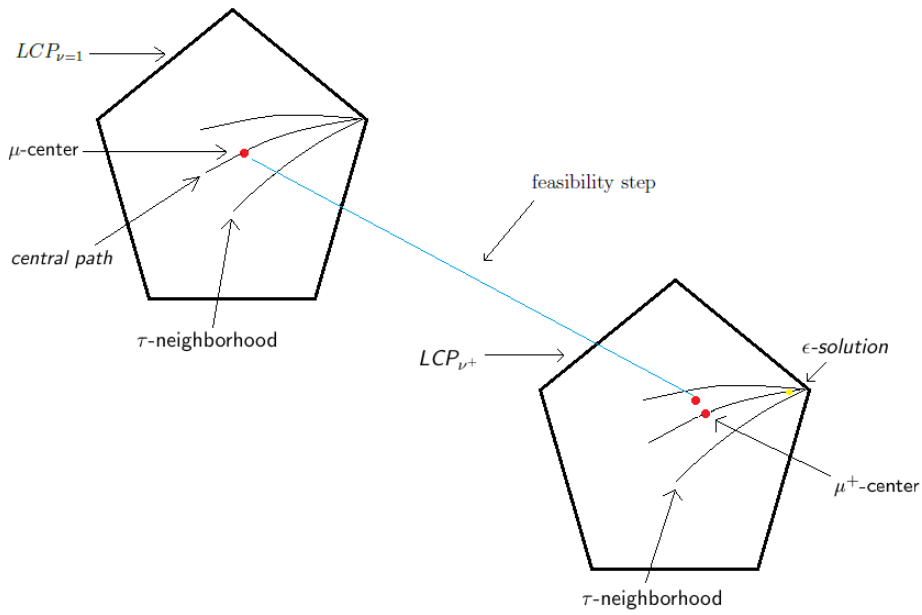


Figure 2: Graphical representation of the improved algorithm for  $LCP_v$

Given the assumptions, which include that the matrix  $M$  is positive semi-definite, the above system (6) has a unique solution for  $(x, s) > 0$ . Note that we are targeting  $\mu^+ = (1 - \theta)\mu$  center rather  $\mu$  center as it is done in majority of other algorithms which also improves the behavior of the algorithm. Taking a full Newton-step along the search direction, one constructs a new point  $(x^f, s^f)$  as follows

$$\begin{aligned} x^f &= x + \Delta^f x \\ s^f &= s + \Delta^f s. \end{aligned} \tag{7}$$

In the Section 3, Analysis of the Algorithm, it will be shown that with the appropriate choice of parameters  $\tau$  and  $\theta$ , the feasibility point  $(x^f, s^f)$  is in the  $\tau$  neighborhood of the central path, hence,  $(x^f, s^f)$  is in fact a new iterate  $(x^+, s^+) = (x^f, s^f)$ .

Previous discussion can be summarized in the following outline of the algorithm.

### Improved Full-Newton-Step Infeasible IPM for LCP

#### Input

Accuracy parameter  $\epsilon > 0$ ,  
 Barrier update parameter  $\theta$ ,  $0 < \theta < 1$ ,  
 Threshold parameter  $\tau$ ,  $0 < \tau < 1$ ,  
 Initial points:  
 $x^0 > 0, s^0 > 0$ ,  
 $\mu = \mu^0$  with  $x^0 s^0 = \mu^0 e$ ,  
 $\nu = 1$ .

#### begin

**while**  $\max(x^T s, \|s - Mx - q\|) \geq \epsilon$  **do**  
**begin**  
 Update  $\mu = (1 - \theta)\mu$ ,  $\nu = (1 - \theta)\nu$ ;  
 Calculate  $\Delta_x^f, \Delta_s^f$  by solving (6);  
 Update  $(x, s) = (x, s) + (\Delta_x^f, \Delta_s^f)$  as in (7);  
**end**  
**end**

#### end

Figure 3: Algorithm

In the sequel we will refer to the algorithm described in the Figure 3 simply as the *Algorithm*.

### 3. Analysis of the Algorithm

The main part of the analysis of the *Algorithm* consists of showing that with the appropriate choice of the threshold and barrier parameters, the point  $(x^f, s^f)$  obtained by the feasibility step (6) - (7) is both feasible and close enough to the central path thanks to the tighter proximity estimate based on the Lemma 3, i.e.,  $\delta(x^f, s^f; \mu^+) \leq \tau$ .

The analysis will require the following transformations

$$v = \sqrt{\frac{xs}{\mu}}, \quad d_x = \frac{v \triangle x}{x}, \quad d_s = \frac{v \triangle s}{s} \quad (8)$$

where the operations are component-wise operations of vectors.

Using the above scaled search directions the system (6) transforms into the following system

$$\begin{aligned} \widetilde{M}d_x - d_s &= D\mu^{-1/2}\theta\nu r^0 \\ d_x + d_s &= v^{-1}(1 - \theta) - v \end{aligned} \quad (9)$$

where  $\widetilde{M} = DMD$  and  $D = S^{-1/2}X^{1/2}$ . The matrices  $X$  and  $S$  represent diagonal matrix forms of vectors  $x$  and  $s$ , i.e.  $X = \text{diag}(x)$  and  $S = \text{diag}(s)$ .

Using (8) and the second equation of the system (6), we have

$$\begin{aligned} x^f s^f &= (x + \triangle^f x)(s + \triangle^f s) = xs + x \triangle^f s + s \triangle^f x + \triangle^f x \triangle^f s = (1 - \theta)\mu e + \triangle^f x \triangle^f s \\ &= (1 - \theta)\mu e + \frac{xs}{v^2} d_x^f d_s^f = (1 - \theta)\mu e + \mu d_x^f d_s^f = \mu[(1 - \theta)e + d_x^f d_s^f]. \end{aligned} \quad (10)$$

The following lemma then follows.

**Lemma 2.** *The iterates  $(x^f, s^f)$  are strictly feasible if and only if  $(1 - \theta)e + d_x^f d_s^f > 0$ .*

The more practical, however, just sufficient condition for strict feasibility is given in the following corollary.

**Corollary 3.1.** *The iterates  $(x^f, s^f)$  are strictly feasible if  $\|d_x^f d_s^f\|_\infty < 1 - \theta$ .*

The proofs of the above lemmas can be found in [5, 16].

Our goal is to find an upper bound for  $\delta(x^f, s^f; \mu^+)$ . The notation  $\omega(v) = \frac{1}{2}(\|d_x^f\|^2 + \|d_s^f\|^2)$  will be useful, as well as the following relationships between norms.

$$\|d_x^f d_s^f\|_\infty \leq \|d_x^f d_s^f\| \leq \|d_x^f\| \|d_s^f\| \leq \frac{1}{2}(\|d_x^f\|^2 + \|d_s^f\|^2) = \omega(v). \quad (11)$$

Then, the above corollary assumes the following form.

**Corollary 3.2.** *If  $\omega(v) < (1 - \theta)$ , then  $(x^f, s^f)$  are strictly feasible.*

The following function  $\xi$  and Lemma 3, introduced in [18], play a central role in the subsequent analysis.

$$\xi(t) = \frac{1+t}{1-\theta} + \frac{1-\theta}{1+t} - 2 = \frac{(\theta+t)^2}{(1-\theta)(1+t)} \geq 0, \quad t > -1. \quad (12)$$

**Lemma 3.** *Let  $a, b \in \mathbb{R}^n$ ,  $r \in [0, 1)$  and  $f(a, b) = \sum_{i=1}^n \xi(a_i b_i)$ . If  $\|a\|^2 + \|b\|^2 \leq 2r^2$ , then*

$$f(a, b) \leq (n-1)\xi(0) + \max\{\xi(r^2), \xi(-r^2)\}.$$

The Lemma 3 is used to prove the following lemma which gives the upper bound for  $\delta(v^f)$ .

**Lemma 4.** *If  $\omega(v) < 1 - \theta$ , then*

$$4\delta(v^f)^2 \leq (n-1)\xi(0) + \max\{\xi(\omega(v)), \xi(-\omega(v))\}.$$

**Proof.** Using (5) and (10),  $(v^f)^2$  can be written as

$$(v^f)^2 = \frac{x^f s^f}{\mu^+} = \frac{(1-\theta)e + d_x^f d_s^f \mu}{1-\theta} \frac{1}{\mu} = e + \frac{d_x^f d_s^f}{1-\theta}.$$

Since  $4\delta(v^f)^2 = \|v^f - (v^f)^{-1}\|^2$ , then

$$\begin{aligned} 4\delta(v^f)^2 &= \sum_{i=1}^n \left( v_i^f - \frac{1}{v_i^f} \right)^2 \\ &= \sum_{i=1}^n \left[ (v_i^f)^2 - 2 + \left( \frac{1}{v_i^f} \right)^2 \right] \\ &= \sum_{i=1}^n (v_i^f)^2 + \sum_{i=1}^n \left( \frac{1}{v_i^f} \right)^2 - \sum_{i=1}^n 2 \\ &= -2n + \sum_{i=1}^n \left( \frac{(1-\theta) + d_{x_i}^f d_{s_i}^f}{1-\theta} \right) + \sum_{i=1}^n \left( \frac{1-\theta}{(1-\theta) + d_{x_i}^f d_{s_i}^f} \right) \\ &= \sum_{i=1}^n \xi(d_{x_i}^f d_{s_i}^f - \theta). \end{aligned}$$

As it can be seen from the Figure 4,  $\xi(t)$  function is an increasing function when  $t > -1$ . Furthermore, since,  $\theta \in [0, 1)$ , we have

$$\sum_{i=1}^n \xi(d_{x_i}^f d_{s_i}^f - \theta) \leq \sum_{i=1}^n \xi(d_{x_i}^f d_{s_i}^f).$$

Then, using Lemma 3, the result of the lemma follows. ■

Finding the upper bound on  $\delta(v^f)$ , reduces to finding the upper bound for  $\omega(v)$  due to the Lemma 4 which essentially means finding the upper bound for  $\|d_x^f\|^2 + \|d_s^f\|^2$ . To do so we need the following lemma which was proved in [12].

**Lemma 5.** *Given the following system*

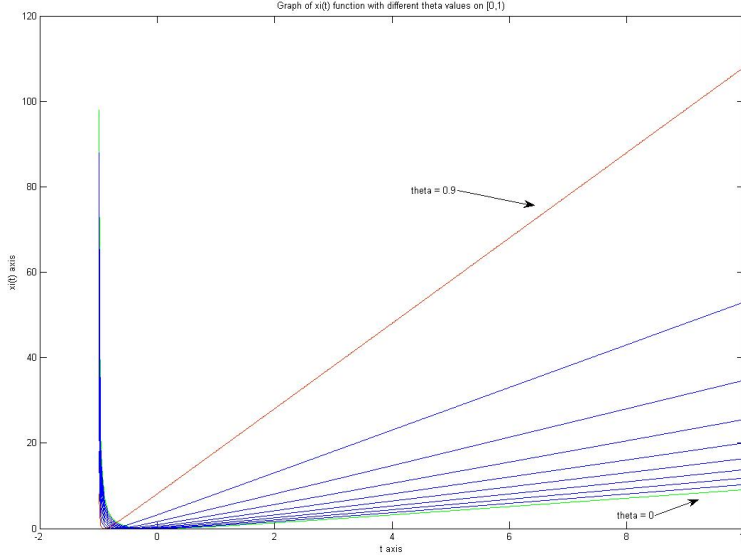
$$\begin{aligned} \widetilde{M}u - z &= \tilde{a} \\ u + z &= \tilde{b} \end{aligned} \tag{13}$$

*the following hold*

- (1)  $Du = (1 + DMD)^{-1}(a + b)$ ,  $Dz = (b - Du)$
- (2)  $\|Du\| \leq \|a + b\|$
- (3)  $\|Du\|^2 + \|Dz\|^2 \leq \|b\|^2 + 2\|a + b\| \|a\|$

where  $D = S^{-1/2}X^{1/2}$ ,  $b = D\tilde{b}$ ,  $a = D\tilde{a}$  and  $\widetilde{M} = DMD$ .



Figure 4: Graph of  $\xi(t)$  for different  $\theta \in [0, 1]$ 

It is easy to see that application of the Lemma 5 to the system (9), leads to

$$\begin{aligned} a &= D(\theta\nu Dr^0 \mu^{-1/2}) = D^2(\theta\nu r^0 \mu^{-1/2}) \\ b &= D((1-\theta)v^{-1} - v) \\ u &= d_x^f \\ z &= d_s^f. \end{aligned}$$

Then, by Lemma 5 - (3), we have

$$\begin{aligned} \|Dd_x^f\|^2 + \|Dd_s^f\|^2 &\leq \|D((1-\theta)v^{-1} - v)\|^2 + 2\|D^2(\theta\nu r^0 \mu^{-1/2})\| \\ &\quad + D((1-\theta)v^{-1} - v)\| \|D^2(\theta\nu r^0 \mu^{-1/2})\|. \end{aligned} \quad (14)$$

After applying the following norm facts to (14),

$$\begin{aligned} (i) \quad &\|Dd_x^f\| \leq \|D\| \|d_x^f\|, \quad \|Dd_s^f\| \leq \|D\| \|d_s^f\| \\ (ii) \quad &\|D^2(\theta\nu r^0 \mu^{-1/2})\| \leq \|D\|^2 \|\theta\nu r^0\| \mu^{-1/2} \\ (iii) \quad &\|D((1-\theta)v^{-1} - v)\| \leq \|D\| \|(1-\theta)v^{-1} - v\| \end{aligned}$$

where  $\|D\|$  represents a matrix norm, we obtain

$$\begin{aligned} \|d_x^f\|^2 + \|d_s^f\|^2 &\leq \|(1-\theta)v^{-1} - v\|^2 \\ &\quad + 2 \left( \|\theta\nu r^0\| \mu^{-1/2} + \|(1-\theta)v^{-1} - v\| \right) \|D(\theta\nu r^0 \mu^{-1/2})\|. \end{aligned} \quad (15)$$

To find the upper bound for  $\|d_x^f\|^2 + \|d_s^f\|^2$  we need to find upper bounds for  $\|D(\theta\nu r^0 \mu^{-1/2})\|$  and  $\|(1-\theta)v^{-1} - v\|$  respectively.

Using the definitions of  $D$ , and  $v$ , and relationship between norms, we obtain the following upper bound for  $\|D(\theta\nu r^0\mu^{-1/2})\|$

$$\begin{aligned}
\|D(\theta\nu r^0\mu^{-1/2})\| &= \frac{\theta\nu}{\sqrt{\mu}}\|Dr^0\| \\
&= \frac{\theta\nu}{\sqrt{\mu}}\|X^{1/2}S^{-1/2}r^0\| \\
&= \frac{\theta\nu}{\sqrt{\mu}}\|\sqrt{\frac{x}{s}}r^0\| \\
&\leq \frac{\theta}{\sqrt{\mu}}\frac{\mu}{\mu^0}\|\sqrt{\frac{x}{s}}r^0\|_1 \quad \longrightarrow \quad (\text{Since } \nu = \mu/\mu^0) \\
&= \frac{\theta}{\mu^0}\|\sqrt{\frac{\mu x}{s}}r^0\|_1 \\
&= \frac{\theta}{\mu^0}\|\sqrt{\frac{\mu}{xs}}xr^0\|_1 \quad \longrightarrow \quad (\text{Since } \sqrt{\frac{\mu}{xs}} = \frac{1}{v}) \\
&\leq \frac{\theta}{\mu^0}\frac{1}{v_{\min}}\|xr^0\|_1 \quad \longrightarrow \quad (\text{Since } \left|\frac{1}{v_i}x_i r_i^0\right| \leq \frac{1}{v_{\min}}|x_i r_i^0| \leq \frac{1}{v_{\min}}|x_i| |r_i^0|) \\
&\leq \frac{\theta}{\mu^0}\frac{1}{v_{\min}}\|x\|_1\|r^0\|_{\infty}.
\end{aligned}$$

Hence,

$$\|D(\theta\nu r^0\mu^{-1/2})\| \leq \frac{\theta}{\mu^0}\frac{1}{v_{\min}}\|x\|_1\|r^0\|_{\infty}. \quad (16)$$

Using initial assumption (2), we have

$$\begin{aligned}
r^0 &= s^0 - Mx^0 - q \\
&= \gamma_d e - \gamma_p M e - q \\
&= \gamma_d \left( e - \frac{\gamma_p}{\gamma_d} M e - \frac{1}{\gamma_d} q \right).
\end{aligned}$$

Thus, we can bound  $\|r^0\|_{\infty}$  as follows

$$\begin{aligned}
\|r^0\|_{\infty} &= \gamma_d \left\| e - \frac{\gamma_p}{\gamma_d} M e - \frac{1}{\gamma_d} q \right\|_{\infty} \\
&\leq \gamma_d \left( 1 - \frac{\gamma_p}{\gamma_d} \|M e\|_{\infty} - \frac{1}{\gamma_d} \|q\|_{\infty} \right).
\end{aligned}$$

By assuming  $\max\{\|M e\|_{\infty}, \|q\|_{\infty}\} \leq \gamma_d$ , we have that

$$\|r^0\|_{\infty} \leq \gamma_d(1 + 1 + 1) = 3\gamma_d. \quad (17)$$

The above assumption is in addition to the assumption (2). It is not restrictive and it is used to streamline the convergence analysis.

Substituting the upper bound for  $\|r^0\|$  into (16), we obtain the following upper bound for  $\|D(\theta\nu r^0\mu^{-1/2})\|$

$$\begin{aligned}\|D(\theta\nu r^0\mu^{-1/2})\| &\leq \frac{\theta}{\mu^0} \frac{1}{v_{\min}} \|x\|_1 3\gamma_d \\ &= \frac{\theta}{\gamma_d \gamma_p} \frac{1}{v_{\min}} 3\gamma_d \|x\|_1 \\ &= \frac{3\theta}{\gamma_p} \frac{\|x\|_1}{v_{\min}}.\end{aligned}\quad (18)$$

The upper bound for  $\|(1-\theta)v^{-1} - v\|$  is found as follows

$$\begin{aligned}\|(1-\theta)v^{-1} - v\|^2 &= \|(1-\theta)v^{-1}\|^2 - 2(1-\theta)(v^{-1})^T v + \|v\|^2 \\ &= (1-\theta)^2 \|v^{-1}\|^2 - 2(1-\theta)n + \|v\|^2 \\ &= (1-\theta)^2 \|v^{-1}\|^2 - 2n + 2\theta n + \|v\|^2 \\ &\leq \|v^{-1}\|^2 - 2n + \|v\|^2 + 2\theta n \\ &= \|v^{-1} - v\|^2 + 2\theta n \\ &= 4\delta(v)^2 + 2\theta n.\end{aligned}$$

Hence, we have

$$\|(1-\theta)v^{-1} - v\| \leq \sqrt{4\delta(v)^2 + 2\theta n}. \quad (19)$$

Substituting (18) and (19) into (15), we obtain the following upper bound for  $\|d_x^f\|^2 + \|d_s^f\|^2$

$$\|d_x^f\|^2 + \|d_s^f\|^2 \leq (4\delta(v)^2 + 2\theta n) + 2 \left( \frac{3\theta}{\gamma_p} \frac{\|x\|_1}{v_{\min}} + \sqrt{4\delta(v)^2 + 2\theta n} \right) \frac{3\theta}{\gamma_p} \frac{\|x\|_1}{v_{\min}}. \quad (20)$$

Consequently, finding the upper bound for  $\|d_x^f\|^2 + \|d_s^f\|^2$ , reduces to finding the upper bound and the lower bound for  $\|x\|_1$  and  $v_{\min}$  respectively. This is achieved by using the following lemma, proof of which can be found in [16].

**Lemma 6.** *Let*

$$q(\delta) = \delta + \sqrt{\delta^2 + 1}.$$

*Then, the following inequalities hold*

$$\begin{aligned}(i) \quad & q^{-1}(\delta) \leq v_i \leq q(\delta) \\ (ii) \quad & \|x\|_1 \leq (2 + q(\delta))n\gamma_p, \quad \|s\|_1 \leq (2 + q(\delta))n\gamma_d.\end{aligned}$$

Using Lemma 6, the upper bound obtained in (20) becomes

$$\begin{aligned}\|d_x^f\|^2 + \|d_s^f\|^2 &\leq (4\delta^2 + 2\theta n) + \frac{18\theta^2}{\gamma_p^2} \frac{\|x\|_1^2}{v_{\min}^2} + \frac{6\theta}{\gamma_p} \sqrt{4\delta^2 + 2\theta n} (2 + q(\delta))q(\delta) \\ &= (4\delta^2 + 2\theta n) + 18\theta^2 n^2 (2 + q(\delta))^2 q^2(\delta) \\ &\quad + 6\theta n \sqrt{4\delta^2 + 2\theta n} (2 + q(\delta))q(\delta).\end{aligned}\quad (21)$$

Finally, we obtain the the upper bound for  $\delta(v^f)$  in terms of  $\delta$  and  $\theta$

$$\delta(v^f) \leq \frac{1}{2} \sqrt{(n-1)\xi(0) + \max\{\xi(\omega(v)), \xi(-\omega(v))\}} \leq \tau, \quad (22)$$

where

$$\omega(v) \leq (2\delta^2 + \theta n) + 9\theta^2 n^2 (2 + q(\delta))^2 q^2(\delta) + 3\theta n \sqrt{4\delta^2 + 2\theta n} (2 + q(\delta)) q(\delta)$$

and

$$q(\delta) = \delta + \sqrt{\delta^2 + 1}.$$

Since  $\delta \leq \tau$ ,  $\delta$  is replaced with  $\tau$  in the above upper bounds (22). Now, we need to find specific values of  $\tau$  and  $\theta$  such that (22) is satisfied. Analytic estimate of these values is difficult, hence, we have performed a numerical estimate using MATLAB code to find specific  $\tau$  and  $\theta$  values satisfying the inequality (22). The results are given in the Table 1.

$\theta$	$\tau$	$\delta(v^f) : n = 2$	$n = 100$	$n = 1000$
$\frac{1}{39+n}$	$\frac{1}{5}$	0.1806	0.0751	0.0468
$\frac{1}{40+n}$	$\frac{1}{4}$	0.2483	0.1060	0.0721
$\frac{1}{53+n}$	$\frac{1}{3}$	0.3272	0.1787	0.1329
$\frac{1}{170+n}$	$\frac{1}{2}$	0.4981	0.4371	0.3708

Table 1: Proximity of new iterates to  $\mu$ -center for a certain choice of  $\tau$  and  $\theta$

It can be observed from the Table 1, that the estimate is tighter for smaller values of  $n$ , where  $n$  is a dimension of the problem, while, as  $n$  increases, the new iterates get closer to the central path. The above discussion can be summarized in the following theorem.

**Theorem 3.3.** *Let  $\delta$  and  $\tau$  be one of the pairs in the Table 1 and  $(x, s)$  be a current iterate of the Algorithm with  $\delta(x, s; \mu) \leq \tau$ . Then, after the feasibility step of the Algorithm, the point  $(x^f, s^f)$  is strictly feasible and satisfies (22), that is,  $\delta(x^f, s^f; \mu^+) \leq \tau$ . Thus,  $(x^f, s^f)$  is the new iterate,  $(x^+, s^+) = (x^f, s^f)$ .*

The above theorem essentially states that the Algorithm is well defined.

#### 4. Complexity analysis

In this section, we calculate an upper bound on the required number of iterations of the Algorithm to obtain  $\epsilon$ -approximate solution of *monotone-LCP* (1). We begin with the following lemma proof of which can be found in [16].

**Lemma 7.** *The following equation and inequalities hold:*

- (i)  $\Delta x^T \Delta s \leq \mu \delta^2$
- (ii)  $x^f s^f = \mu e + \Delta x \Delta s$
- (iii)  $(x^f)^T s^f \leq \mu(n + \delta^2)$ .

The above Lemma 7 is used in the proof of the theorem below that gives the upper bound on the required number of iteration for the *Algorithm* to find  $\epsilon$ -approximate solution of *monotone – LCP* (1).

**Theorem 4.1.** *Let  $\theta = \frac{1}{39+n}$ ,  $\tau = \frac{1}{5}$  and  $\mu^0 = \frac{(x^0)^T s^0}{n}$ . Then, the *Algorithm* requires at most  $(39+n) \log \frac{51(x^0)^T s^0}{50\epsilon}$  iterations to reach the  $\epsilon$ -approximate solution of *monotone – LCP* (1).*

**Proof.** Let  $x_k$  and  $s_k$  be the  $k$ -th iterates of the algorithm. Then,

$$\begin{aligned}
x_k^T s_k &\leq \mu_k(n + \delta^2) \\
&\leq \mu_k\left(n + \frac{1}{25}\right) \\
&= (1 - \theta)^k \mu^0 \left(n + \frac{1}{25}\right) \\
&= (1 - \theta)^k \frac{(x^0)^T s^0}{n} \left(n + \frac{1}{25}\right) \\
&\leq (1 - \theta)^k \frac{(x^0)^T s^0}{n} \left(n + \frac{1}{50}n\right) \text{ (Since } n \geq 2) \\
&= (1 - \theta)^k \frac{(x^0)^T s^0}{n} \frac{51}{50}n \\
&= (1 - \theta)^k (x^0)^T s^0 \frac{51}{50} \\
&\leq \epsilon.
\end{aligned}$$

Then by taking the logarithm of both sides, we have

$$\begin{aligned}
\log \left[ (1 - \theta)^k (x^0)^T s^0 \frac{51}{50} \right] &\leq \log \epsilon \\
\log(1 - \theta)^k + \log(x^0)^T s^0 + \log \frac{51}{50} &\leq \log \epsilon \\
\log(1 - \theta)^k &\leq \log \epsilon - \log(x^0)^T s^0 - \log \frac{51}{50} \\
k \log(1 - \theta) &\leq \log \frac{50\epsilon}{51(x^0)^T s^0} \\
-k \log(1 - \theta) &\geq -\log \frac{50\epsilon}{51(x^0)^T s^0}.
\end{aligned}$$

Since  $-\log(1 - \theta) \geq \theta$ , we obtain the desired iteration bound

$$\begin{aligned}
k\theta &\geq \log \frac{51(x^0)^T s^0}{50\epsilon} \\
k &\geq \frac{1}{\theta} \log \frac{51(x^0)^T s^0}{50\epsilon} \\
k &\geq (39+n) \log \frac{51(x^0)^T s^0}{50\epsilon}.
\end{aligned}$$

■

Similarly, we calculate the required number of iterations for other  $\theta$  and  $\tau$  values listed in the Table 1. The results are shown in the Table 2.

$\theta$	$\tau$	number of iterations
$\frac{1}{40+n}$	$\frac{1}{4}$	$(40 + n) \log \frac{33(x^0)^T s^0}{32\epsilon}$
$\frac{1}{53+n}$	$\frac{1}{3}$	$(53 + n) \log \frac{19(x^0)^T s^0}{18\epsilon}$
$\frac{1}{170+n}$	$\frac{1}{2}$	$(170 + n) \log \frac{9(x^0)^T s^0}{8\epsilon}$

Table 2: Required number of iterations for different  $\tau$  and  $\theta$  values

Hence, we have the following corollary.

**Corollary 4.2.** *Let  $\delta$  and  $\tau$  be one of the pairs in the Table 1. Then the Algorithm is globally convergent and achieves the  $\epsilon$ -approximate solution of monotone – LCP (1) in at most  $O(n \log \frac{n}{\epsilon})$  iterations.*

Recall that the iteration bound of the old version of the algorithm is

$$12n \log \frac{33(x^0)^T s^0}{32\epsilon}.$$

It is not hard to see that both iteration bounds have the same order of magnitude stated in the above Corollary 4.2; however, constant-wise the iteration bound of the new version of the method is much better than the old version for sufficiently large  $n$ .

It is worth noting that the iteration bounds of both old and new versions of the algorithm match the best known iteration bound for these type of methods as far as the order of magnitude is concerned.

## 5. Numerical Results

In this section, we present preliminary numerical results of the implementation of the new and old version of the algorithm for the set of randomly generated *monotone – LCPs*. Recall that the the old version of the algorithm, consists of one feasibility step and a few centering steps at each iteration, while the improved version of the algorithm consists of only one feasibility step. The pseudo-code of the new version of the algorithm is outlined in Figure 3 and the pseudo-code of the old version is outlined in [13]. Both versions are implemented in MATLAB and run on the desktop computer with Intel(R) core(TM) processor and 4 Gb of RAM running Windows 7 operating system.

First, the old version of the algorithm and the improved version of the method are compared . Comparison is given in the Table 3. The number of iterations (No. of it.) represents average number of iterations while the (CPU time) represents average CPU time of all instances of the problems of the given dimension. For the improved version of the algorithm, we use  $\theta = \frac{1}{40+n}$  and  $\tau = \frac{1}{4}$ , while for the old version of the algorithm the value of  $\theta$  is  $\theta = \frac{1}{12n}$  while the value of  $\tau$  is the same. The accuracy for all the cases is set at  $\epsilon = 10^{-4}$ .

Size	Old version	Old version	Improved version	Improved version
	No. of it.	CPU time	No. of it.	CPU time
$3 \times 3$	385	$1.3239 \times 10^{-2}$	439	$1.9803 \times 10^{-2}$
$5 \times 5$	716	$3.2589 \times 10^{-2}$	506	$2.5870 \times 10^{-2}$
$10 \times 10$	1626	$7.7880 \times 10^{-2}$	676	$3.7156 \times 10^{-2}$
$100 \times 100$	23917	16.955375	2697	1.894107

Table 3:  $\theta_{new} = \frac{1}{40+n}$ ,  $\theta_{old} = \frac{1}{12n}$ ,  $\tau = \frac{1}{4}$ ,  $\epsilon = 10^{-4}$ 

As expected, Table 3 shows that the improved version of the algorithm requires less iterations and less CPU time in almost all the cases. The exception is the low dimensional set of problems ( $3 \times 3$ ). However, as the dimension increases the improved version of the algorithm becomes much more efficient than the old version. The reason is twofold. The  $\theta$  value essentially controls the reduction of  $\mu$  (optimality) and  $\nu$  (feasibility); the bigger the  $\theta$ , the bigger the reduction, hence, fewer iterations are needed. For the smaller values of  $n$ , the  $\theta$  value the old version of the algorithm,  $\theta_{old} = \frac{1}{12n}$  is bigger than the  $\theta$  value of the new version,  $\theta_{new} = \frac{1}{40+n}$ . It starts to be the opposite with  $n \geq 4$  with the gap increasing as  $n$  increases. Furthermore, in the new version the centering steps are eliminated which significantly contributes to the better performance of the new version in comparison to the old version of the algorithm.

Next, we investigate the effects of different values of  $\theta$  parameter on the performance of the improved version of the algorithm. In theory, for constant values of  $\theta$  the convergence of the *Algorithm* is not guaranteed. Nevertheless, we run the MATLAB implementation of the Algorithm with  $\theta = 0.2, 0.5, 0.9$  on the set of test problems of different dimensions with accuracy being again  $\epsilon = 10^{-4}$ . Results are shown in Table 4:

Size	$\theta = \frac{1}{40+n}$	$\theta = 0.2$	$\theta = 0.5$	$\theta = 0.9$
$2 \times 2$	417	45	15	5
$5 \times 5$	488	54	17	6
$10 \times 10$	577	61	20	7
$100 \times 100$	1938	87	28	9
$1000 \times 1000$	16795	113	37	X

Table 4: Number of iterations for different  $\theta$  values

It can be observed that, although the convergence of the algorithm for the constant values of  $\theta$  is not guaranteed, the method was able to solve almost all instances of the problem for all values of  $\theta$  except the problem of the largest dimension for the largest value of  $\theta = 0.9$ . The probable reason is that the reduction taken was too aggressive. As expected, the number of iterations reduces dramatically as the value of  $\theta$  increases. Moreover, the number of iterations does not increase significantly as the dimension of the problem increases.

The preliminary implementation and numerical testing indicate that the *Algorithm* has certain computational merit even for the values of  $\theta$  that guarantee convergence and especially for the constant values of  $\theta$ , if one is willing to take a low risk

that method may not converge to the solution for some instances of the problem. Additional details on the implementation of the *Algorithm* and numerical tests can be found in [16].

More sophisticated implementation and more numerical testing are needed to come to the more definite conclusions about the practical behavior of the algorithm.

## 6. Concluding remarks

In this paper an improved version of the Full-Newton-Step Infeasible Interior-Point Method for *monotone – LCP* is considered. The method is outlined in Figure 3 and throughout the text we referred to it as simply the *Algorithm*. The old version of the algorithm was discussed in [13, 5]. In the old version of the method, each iteration consisted of one feasibility step and few centering steps (at most two) per iteration. The centering steps were necessary to bring the point obtained by feasibility step to the  $\tau$ -neighborhood of the central path again. In the improved version of the method, with the suitable choice of the threshold parameter  $\tau$  and barrier parameter  $\theta$ , it is guaranteed that after one feasibility step, the new iterate is feasible and immediately in the  $\tau$ -neighborhood of the central path thanks to the much tighter proximity estimate which is based on the new Lemma 3 introduced in [18]. Thus, the centering steps are eliminated.

The good features of the old version of the method are still preserved in the new version. The *Algorithm* does not require strictly feasible starting point (infeasible algorithm) and it uses full-Newton-steps, thus, avoiding calculations of a step-size at each iteration. Furthermore, a nice feature of both versions of the method is that they work on simultaneously reducing infeasibility and achieving optimality.

The *Algorithm* is globally convergent for the values of the threshold and barrier parameters listed in the Table 1. Furthermore, the *Algorithm* matches the best known iteration complexity for these types of methods, in order of magnitude, which is  $O(n \log \frac{n}{\epsilon})$ . Although the order of magnitude of the iteration bounds of the old and new version of the method is the same, constant-wise the iteration bound of the improved version of the method is much better than the old version for sufficiently large  $n$ .

The disadvantage is that the *Algorithm* is still a short-step method, because  $\theta = O(\frac{1}{n})$ . However, the preliminary implementation and numerical testing of the *Algorithm* indicate that the method has certain computational appeal even for the values of  $\theta$  that guarantee convergence. For the constant values of  $\theta$ , the Algorithm becomes long-step method, but in that case, the global convergence of the method is no longer guaranteed. However, the initial numerical testing shows that in most instances the *Algorithm* still converges and converges very fast, with number of iterations reducing dramatically as the value of  $\theta$  increases. Moreover, the number of iterations does not increase significantly as the dimension of the problem increases. Furthermore, numerical testing shows that the new version of the method performs much better than the old version. More sophisticated implementations and more numerical testing are needed to have a better idea about the practical behavior of the algorithm.



In addition to more numerical testing, some directions for further research include generalization of the method to more general classes of LCPs such as  $P_*(\kappa)$ -LCP and LCPs over symmetric cones.

## References

- [1] Anitescu, M., Lesaja, G. and Potra, F. A. (1997). Equivalence between different formulations of the linear complementarity problem. *Optimization Methods and Software*, 7, 265–290.
- [2] Bai, Y. Q., Lesaja, G. and Roos, C. (2008). A new class of polynomial interior-point algorithms for linear complementarity problems. *Pacific Journal of Optimization*, 4, 19–41.
- [3] Chung, S. J. (1979). A note on the complexity of *LCP*, The *LCP* is strongly NP-complete. Technical report 792, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan.
- [4] Cottle, R. W., Pang J.-S., and Stone, R. E. (1992). *The Linear Complementarity Problem*. Academic Press, Boston, MA.
- [5] Drummer, A. M. (2012). Infeasible full Newton-step interior point method for the linear complementarity problems. Master thesis, Georgia Southern University, Statesboro.
- [6] Facchinei, F. and Pang, J.-S. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, New York.
- [7] Ferris, M. C., Mangasarian, O. L. and Wright, S. J. (2007). *Linear programming with MATLAB*, SIAM, Series on Optimization.
- [8] Gill, P. E., Murray, W., Saunders, M. A., Tomlin, A. J. and Wright, H. M. (1986). On the projected Newton barrier methods for linear programming and an equivalence to Karmarkar’s projective method. *Math Program*, 36, 183–209.
- [9] Karmarkar, N. (1984). A polynomial-time algorithm for linear programming. *Combinatorica*, 4, 373–395.
- [10] Kojima, M. and Mizuno, S. and Yoshise, A. (1989). A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44, 1–26.
- [11] Kojima, M. and Mizuno, S., Noma, T. and Yoshise, A. (1991). *A Unified Approach to Interior Point Algorithms for Linear Complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, New York.
- [12] Lesaja, G. (1996). Interior-point methods for  $P^*$ -complementarity problems. Ph.D. thesis, University of Iowa, Iowa City.
- [13] Lesaja, G., Drummer, A. and Miletic, L. (2012). Infeasible full Newton-step interior-point method for linear complementarity problems. *Croatian Operational Research Review (CRORR)*, 3, 163–175.
- [14] Lesaja, G. and Roos (2010). C. Unified analysis of kernel-based interior-point methods for  $P_*(\kappa)$ -LCP. *SIAM Journal of Optimization*, 20(6), 3014–3039.
- [15] R. D. C. Monteiro and S. J. Wright (1996). A superlinear infeasible-interior-point affine scaling algorithm for LCP. *SIAM J. Optim.*, 6, 1–18.
- [16] Ozen, M. (2015). Improved infeasible full Newton-step interior point method for the linear complementarity problems. Master thesis, Georgia Southern University, Statesboro.
- [17] Renegar, J. (1988). A polynomial-time algorithm, based on Newton’s method, for linear programming. *Math Program*, 40, 59–73.
- [18] Roos, C. (2015). An improved and simplified full-Newton step  $O(n)$  infeasible interior point method for linear optimization. *SIAM J. Optim.*, 25, 102–114.

- [19] Wright, S. J. and Zhang Y. (1996). A superquadratic infeasible-interior-point method for linear complementarity problems. *Math. Programming Ser. A*, 73, 269–289.