Predictor-corrector interior-point algorithm for sufficient linear complementarity problems based on a new search direction

Zs. Darvay, T. Illés, J. Povh, and P. R. Rigó

June 2018

Eötvös Loránd University of Sciences
Department of Operations Research
Predictor-corrector interior-point algorithm for sufficient linear complementarity problems based on a new search direction

Zs. Darvay, T. Illés, J. Povh, and P. R. Rigó

Abstract

We introduce a new predictor-corrector (PC) interior-point algorithm (IPA), which is suitable for solving linear complementarity problem (LCP) with $P_\ast(\kappa)$-matrices. We use the method of algebraically equivalent transformation (AET) of the nonlinear equation of the system which defines the central path. For AET we use $\varphi(t) = t - \sqrt{t}$ to determine new search direction.

We prove that the algorithm has $O\left( (1 + 2\kappa)\sqrt{n} \log \frac{9n\mu^0}{8\varepsilon} \right)$ iteration complexity, where $\kappa$ is the handicap of the input matrix. To the best of our knowledge, this is the first PC IPA for sufficient LCPs which is based on this search direction.

We implement the proposed PC IPA in C++ programming language and demonstrate its performance on three families of LCPs. The first family consists of LCPs with $P_\ast(\kappa)$-matrices. These are the first numerical results related to $P_\ast(\kappa)$-LCPs where the matrices of the problems have positive handicap [33].

The second family of LCPs has the $P$-matrix defined by Zs. Csizmadia [24]. Eisenberg-Nagy and de Klerk showed that the handicap of these matrices should be at least $2^{2n-8} - \frac{1}{4}$. Namely, from the known complexity results for $P_\ast(\kappa)$-LCPs might follow that the computational performance of IPAs on LCPs with the matrix defined by Zs. Csizmadia could be very poor. As it was predicted, most of the IPAs with classical AET based on the function $\varphi(t) = t$ were not able to solve LCPs with the Csizmadia matrix even for example of size $n = 10$. However, the PC IPA presented in this paper, finds $\varepsilon$-optimal solution in very small number of iterations for LCPs with Csizmadia matrix.
The third family of problems consists of the LCPs related to the copositivity test of 88 matrices from [9]. For each of these matrices we create a special LCP and try to solve it using our IPA. If the LCP does not have a solution, then the related matrix is strictly copositive, otherwise it is on the boundary or outside the copositive cone. For these LCPs we do not know whether the underlying matrix is $P^*_\kappa$ or not, but we could reveal the real copositivity status of the input matrices in 83 out of 88 cases (accuracy $\geq 94\%$).

Seemingly, the implementation of our PC IPA solves efficiently the LCP test problems collected for numerical evaluation of different IPAs.

1 Introduction

1.1 Definition of linear complementarity problem

The linear complementarity problem (LCP) is a well-known problem. Given $M \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$, we want to find vectors $x, s \in \mathbb{R}^n$ that satisfy the following constraints:

$$-Mx + s = q, \quad xs = 0, \quad x, s \geq 0.$$  \hspace{1cm} (LCP)

There are several variants of LCPs, such as standard, horizontal, mixed and geometric LCPs. Anitescu et al. [3] proved the equivalences between them. Note that LCP includes linear programming (LP) and convex quadratic optimization (QO), as special cases. The most important basic results about the theory, applications and methods to solve LCPs are summarized in the monographs written by Cottle et al. [12] and Kojima et al. [41].

1.2 Motivation for our work

LCP has a wide range of applications in different fields, such as economics, optimization theory, engineering [28]. For example, the Karush-Kuhn-Tucker conditions of the (non-convex) quadratic optimization problem yield to LCP. The Arrow-Debreu competitive market equilibrium problem with linear and Leontief utility functions can be also formulated as LCP [70]. Recent work of [9] reveals that LCP can be used as the main ingredient of the NP-hard copositivity test. In general, LCP belongs to the class of NP-complete problems [11], therefore any progress in (approximately) solving LCP would straightforwardly imply a progress in (approximately) solving several NP-hard problems.

Kojima et al [41] showed that if the problem’s matrix $M$ has a special property, called $P^*_\kappa$-property (see Definition 1), IPAs for LCP have polynomial
iteration complexity in the size of the problem, the bit size of the data, the final accuracy of the solution and in the special parameter, called the handicap of the problem’s matrix $M$ (see the paragraphs after Definition 1). Furthermore, there was a gap between theory and practice in case of IPAs. This meant, that through a decade or two the small-update IPAs had better theoretical complexity, while the large-update ones seemed to perform better in practice. There were several attempts to close the gap between the theoretical complexity and practical, computational performance of large-update IPAs. Two main directions of this research were highlighted by the introduction of self-regular function by Peng et al. [51] for linear optimization problems and by proposing new analysis with special wide neighbourhood and providing new search directions by Ai and Zhang [2] for monotone LCPs. Almost at the same time, Potra [52] achieved same complexity result as Ai and Zhang by using another type of wide neighbourhoods and showed superlinear convergence of his PC IPA. Up to our best knowledge, the first large-update IPA for sufficient LCPs with the best complexity result was published by Liu and Potra [42].

Although vast amount of literature demonstrates big efforts of mathematical optimization community to make progress towards solving sufficient LCPs by IPAs, see Subsection 1.3 for literature overview. However, the main challenge in this research area remains unsolved: how to solve efficiently sufficient (or general) LCPs in theory and practice? With this paper we provide few progressive steps in this direction - we propose a new variant of PC IPA that retains the best known polynomial iteration complexity by incorporating a novel search direction based on algebraically equivalent transformation (AET), as introduced in [22]. We have implemented the new PC IPA in the C++ programming language. Our paper is one of the rare exceptions which provides numerical study with $P_*(\kappa)$-matrices having positive $\kappa$ parameter. Based on three types of test set problems, our implementation of PC IPA shows very good computational performance.

1.3 Related work

During the long history of the study of LCPs many different solution methods have been presented, among others pivot algorithms such as criss-cross algorithms [14,15,29,30]. However, the IPAs received more attention, especially after the seminal work of Kojima et al. [41].

Many researchers tried to reduce the gap between the theoretical and practical performance of IPAs. In 2005, Ai and Zhang [2] proposed a large-update IPA for monotone variant of LCP which has the same complexity as the
currently best-known short-step interior-point methods. Potra [57] generalized this method to LCP with \( P_*(\kappa) \) matrix. The IPAs for solving sufficient LCPs (i.e., LCP with sufficient matrix \( M \) - see Definition (2.3)) have been also extended to general LCP [36, 37] and to \( P_*(\kappa) \)-LCPs over symmetric cones [7, 43, 60].

The PC IPAs turned out to be efficient in practice. They do in the main iteration one predictor and one or more corrector steps. The first PC IPA for linear optimization was introduced by Mehrotra [47] and Sonnevend et al. [61]. Potra and Sheng [54, 55] proposed PC IPAs for sufficient LCPs.

Mizuno, Todd and Ye [49] gave the first PC IPA for linear programming which uses only one corrector step in the main iteration. After that, Miao [48] generalized this IPA to \( P_*(\kappa) \)-LCPs. Following his result, several Mizuno-Todd-Ye type PC IPAs have been developed among others by Illés, Nagy [34] and Kheirfam [39].

The determination of search directions is important in the case of IPAs. The most widely used technique for this is based on barrier functions. Peng, Roos and Terlaky [51] reduced the theoretical complexity of large-update IPAs by using self-regular kernel functions. Moreover, Lešaja and Roos [59] considered a unified analysis of IPAs for \( P_*(\kappa) \)-LCPs that are based on eligible kernel functions.

Illés et al. [35, 36] generalized three types of IPAs (large-update, affine scaling and PC) for solving LCPs with general matrices. The proposed IPAs either solve the problems with rational coefficient matrix in polynomial time or give a polynomial size certificate that the problem’s matrix does not belong to the set of \( P_*(\kappa) \), with arbitrary large, but apriori fixed, rational, positive \( \kappa \). Furthermore, Potra and Liu [53] proposed PC IPA for sufficient LCPs which acts in a wide neighbourhood of the central path and the algorithm does not depend on the handicap of the problem.

In 2002 Darvay [16, 17] proposed a new technique for finding search directions of IPAs for linear programming problems, namely the algebraic equivalent transformation (AET) of the system which defines the central path. This technique consists of applying a continuously differentiable, invertible, monotone increasing function \( \varphi : (\xi, \infty) \to \mathbb{R} \), where \( \xi > 0 \), on the nonlinear equation of the central path problem. After the transformation the Newton’s method is applied in order to give the new search directions. Since then, several researchers have been extended this approach to more general optimization problems, such as LCPs [1, 5, 6, 39, 44], semidefinite programming (SDP) [45, 46, 66], second-order cone programming (SOCP) [68] and symmetric optimization (SO) [38, 69].

The first PC IPAs that use the AET method for defining the search direc-
tions have been proposed by Darvay [18,19] for LP and linearly constrained convex optimization, respectively. Kheirfam [39] generalized the above mentioned algorithms for $P_*(\kappa)$-horizontal LCPs. In the literature, the most widely used function for finding the search directions using this technique is the identity map. Darvay [16,17] was the first who used the square root function in order to give the search directions. In 2016, Darvay et al. [22] proposed an IPA for LP based on the direction using a new function, namely the difference of the identity and the square root map. Recently, Kheirfam and Haghighi [40] have been introduced an IPA for $P_*(\kappa)$-LCPs which is based on a new search direction generated by using the function $\varphi(t) = \sqrt{t} - \frac{1}{2}(1+\sqrt{t})$.

An interesting question regarding the AET method is whether a general class of functions $\varphi$ can be given in order to define a polynomial-time IPA. Related to this problem, recently, Haddou et al. [32] have proposed a family of smooth concave functions which yields to IPAs with the best known iteration bound.

### 1.4 Our contribution

In this paper we give a new general framework for the determination of Newton-systems, scaled systems in case of PC IPAs for $P_*(\kappa)$-LCPs. Furthermore, we propose a new variant of IPA for LCP with sufficient matrix $M$ which has good theoretical and practical performance. The novelty of this IPA is the search direction based on algebraically equivalent transformation (AET) with function $\varphi(t) = t - \sqrt{t}$, introduced in [22]. A corrector-predictor version of this algorithm for LP was proposed in [20]. We point out that this function does not belong to the family introduced by Haddou et al. [32]. We apply Newton’s method to the transformed system to find new search directions. Naturally, due to the used function $\varphi$ the analysis of the IPA becomes more complicated, because in all iterations we should assure that the components of the $v$-vectors of the scaled space are greater than $\frac{1}{2}$.

In spite of this fact, we prove in Theorem 5.10 that our proposed IPA has $O\left((1 + 2\kappa)\sqrt{n} \log \frac{9n\mu^0}{8\varepsilon}\right)$ iteration complexity, where $\kappa$ is the handicap of matrix $M$ of order $n$, $\mu^0$ is the starting normalized complementarity gap and $\varepsilon$ is the final displacement from the complementarity gap, respectively. This is the first PC IPA for solving $P_*(\kappa)$-LCPs which uses this function in AET.

In order to show the efficiency of our algorithm we have implemented it in the C++ programming language. We demonstrate its practical performance by providing numerical results on a family of sufficient matrices with positive handicap from [33], where our IPA outperforms the other variants of IPA based on other AETs. It should be mentioned that these are the first numerical
results related to $P_*(\kappa)$-LCPs where the problem’s matrices have positive handicap.

Additionally, we consider LCPs related to matrix copositivity test from [9]. For these matrices we know the real status of copositivity, but the corresponding LCPs are not necessarily included in the class of sufficient LCPs. Due to the fact that the PC IPA provides $\varepsilon$-optimal solution, we need to propose heuristic decision rules in order to identify the copositivity property of the given matrix from a numerical result. Using the LCP approach, numerical results provided by PC IPA and the introduced heuristic decision rules we were able to decide with accuracy around 94%, whether the matrix is strictly copositive, on the boundary or outside the copositive cone. These results confirm that LCP is a promising tool to test copositivity of given matrices.

The outline of the paper is as follows. In Section 2 we give some basic concepts about the linear complementarity problems with $P_*(\kappa)$-matrices. In Section 3 we describe the AET method for determining search directions in case of IPAs and we consider a general approach for determining the Newton-systems and scaled systems in case of PC IPAs for $P_*(\kappa)$-LCPs. Section 4 contains the new proposed PC IPA for $P_*(\kappa)$-LCPs, which is based on the direction generated by applying the function $\varphi(t) = t - \sqrt{t}$. In Section 5 we present the analysis of the introduced PC IPA. In Section 6 we demonstrate the efficiency of our algorithm through two families of numerical results. Finally, we give some concluding remarks and discussions in Section 7.

2 Linear complementarity problems with $P_*(\kappa)$-matrices

2.1 $P_*(\kappa)$-matrices and sufficient matrices

The notion of $P_*(\kappa)$-matrices was introduced by Kojima et al. [41].

Definition 2.1. (Kojima et al. [41]) Let $\kappa \geq 0$ be a nonnegative number. A matrix $M \in \mathbb{R}^{n \times n}$ is called $P_*(\kappa)$-matrix if

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

where

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

We use $P_\star(\kappa)$ also to denote the set of all square real matrices $M$ satisfying (1). Note that $P_\star(0)$ is the set of positive semidefinite matrices. The handicap of $M$ is the smallest value of $\kappa \geq 0$ such that $M$ is $P_\star(\kappa)$-matrix.

**Definition 2.2.** (Kojima et al. [41]) A matrix $M \in \mathbb{R}^{n \times n}$ is called $P_\star$-matrix if it is a $P_\star(\kappa)$-matrix for some $\kappa \geq 0$. We use $P_\star$ also to denote the set of all $P_\star$-matrices, i.e.,

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

Another matrix class, the class of sufficient matrices was introduced by Cottle et al. [13].

**Definition 2.3.** (Cottle et al. [13]) A matrix $M \in \mathbb{R}^{n \times n}$ is a column sufficient matrix if for all $x \in \mathbb{R}^n$

$$X(Mx) \leq 0 \text{ implies } X(Mx) = 0,$$

and row sufficient if $M^T$ is column sufficient. The matrix $M$ is sufficient if it is both row and column sufficient.

Kojima et al. [41] proved that a $P_\star$-matrix is column sufficient and Guu and Cottle [31] proved that it is row sufficient, too. This means, that each $P_\star$-matrix is sufficient. Moreover, Väliaho [64] showed the other inclusion, so the class of $P_\star$-matrices is equivalent to the class of sufficient matrices.

As we announced in the introduction and is explicitly stated in Theorem 5.10, the worst-case iteration complexity of the IPAs for LCP depends on the handicap of the matrix $M$. Therefore, it is important to decide if a given matrix $M$ is $P_\star(\kappa)$-matrix, for some given value $\kappa$. Väliaho [63] proposed an algorithm which decides whether a matrix $M$ is sufficient or not. Furthermore, Väliaho [65] also introduced another algorithm which determines the handicap of a sufficient matrix and he conjectured that the handicap of a matrix $M$ is a continuous function of the elements of $M$. Tseng [62] showed that deciding whether a square matrix with rational entries is a column sufficient matrix is a co-NP-complete problem. This suggests that given a square matrix $M$ it can not be decided in polynomial time (unless P=NP) whether $M \in P_\star(\kappa)$. After all, it is not surprising that both of Väliaho’s algorithms have exponential running time.

Klerk and E.-Nagy [24] showed that the handicap of a $P_\star(\kappa)$-matrix may be exponential in its bit size. From their observation follows that the known complexity bounds of IPAs may not be polynomial in the input size of the LCP, if the $\kappa$ of the matrix is exponentially large in the size and bit size of

---

*Operations Research Reports No. 2018-03*
the problem. As an example, they presented a matrix which was suggested by Zsolt Csizmadia:

$$M = \begin{pmatrix}
1 & 0 & 0 & \cdots & 0 \\
-1 & 1 & 0 & \cdots & 0 \\
-1 & -1 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-1 & -1 & -1 & \cdots & 1
\end{pmatrix} \quad (2)$$

We give some numerical results related to the $P_\ast(\kappa)$-LCPs using this special matrix in Section 6.

We believe that it is important to mention here a conjecture stated by Klerk and E.-Nagy [24]: Let $A \in \mathbb{Z}^{n \times n}$ be a sufficient matrix, with $L(A)$ bit length. Then,

$$\kappa(A) \leq 2^{p(L(A))},$$

where $\kappa(A)$ denotes the handicap of matrix $A$ and $p$ is a univariate polynomial.

### 2.2 Sufficient matrices and the central path for (LCP)

We use the following notations to denote the feasible region of (LCP), its interior and the solutions set of (LCP):

$$\mathcal{F} := \{(x, s) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : -Mx + s = q\},$$

$$\mathcal{F}^+ := \{(x, s) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : -Mx + s = q\},$$

$$\mathcal{F}^* := \{(x, s) \in \mathcal{F} : xs = 0\}.$$

In the above relations we denoted by $\mathbb{R}^n_+$ the $n$-dimensional nonnegative orthant and by $\mathbb{R}^n_+$ the positive orthant, respectively.

Throughout the paper we will assume that $\mathcal{F}^+ \neq \emptyset$, there is an initial point $(x^0, s^0) \in \mathcal{F}^+$ and $M$ is a $P_\ast(\kappa)$-matrix. The central path problem for (LCP) is:

$$-Mx + s = q,$$

$$x, s > 0,$$

$$xs = \mu e,$$

where $e$ denotes the $n$-dimensional vector of ones and $\mu > 0$. Kojima et al. [41] proved the uniqueness of the central path and that the sequence $\{(x(\mu), s(\mu)) | \mu > 0\}$
μ > 0\} of solutions lying on the central path parameterised by μ > 0 approach the solution (x, s) of the (LCP).

The following theorem proves the existence and uniqueness of the central path. T. Illés, C. Roos, and T. Terlaky gave an elementary constructive proof of this theorem in an unpublished manuscript in 1997. The constructive proof is given in the PhD thesis of M. E.-Nagy [50].

**Theorem 2.1.** Let a linear complementarity problem with a $P_*(\kappa)$-matrix $M$ be given. Then, the following statements are equivalent:

1. $\mathcal{F}^+ \neq \emptyset$;
2. $\forall w \in \mathbb{R}_+^n, \exists!(x, s) \in \mathcal{F}^+ : xs = w$;
3. $\forall \mu > 0, \exists!(x, s) \in \mathcal{F}^+ : xs = \mu e$, i.e. the central path exists and it is unique.

In the rest of this subsection, we recall some important results that can be found in the book of Kojima et al. [41].

**Proposition 2.1.** (Kojima et al. [41]) If $M \in \mathbb{R}^{n \times n}$ is a $P_*(\kappa)$-matrix then the matrix

$$M' = \begin{pmatrix} -M & I \\ S & X \end{pmatrix}$$

is a nonsingular matrix for any positive diagonal matrices $X, S \in \mathbb{R}^{n \times n}$, where $I$ is the $n$-dimensional identity matrix.

The importance of the previous proposition becomes clear when we deal with the solvability of the Newton system. The next theorem is related to the scaling of $P_*$-matrices.

**Theorem 2.2.** (Kojima et al. [41]) Let $A \in \mathbb{R}^{n \times n}$, $P = \text{diag}(p_1, \ldots, p_n)$, $Q = \text{diag}(q_1, \ldots, q_n)$, where $p_i q_i > 0$ for all $i = 1, \ldots, n$, and $B = PAQ$. Then, if $A \in P_*(\kappa)$ for some $\kappa \geq 0$, then $B \in P_*(\kappa')$, where $\kappa' \geq \kappa$ is such that

$$\frac{1 + 4\kappa'}{1 + 4\kappa} = \max_i \frac{p_i}{q_i} = \min_i \frac{p_i}{q_i}.$$

These results prove that the Newton system and the scaled system have unique solution.
**Corollary 2.1.** Let $M \in \mathbb{R}^{n \times n}$ be a $P_\ast(\kappa)$-matrix, $x, s \in \mathbb{R}^n_+$. Then, for all $a_\varphi \in \mathbb{R}^n$ the system

$$
-M\Delta x + \Delta s = 0
$$

$$
S\Delta x + X\Delta s = a_\varphi
$$

has a unique solution $(\Delta x, \Delta s)$, where $X$ and $S$ are the diagonal matrices obtained from the vectors $x$ and $s$.

### 3 Determining search directions in case of PC IPAs

In this section we will determine search directions using the algebraically equivalent transformation (AET) method introduced in [17]. Let $\varphi : \left( \frac{1}{2}, \infty \right) \to \mathbb{R}_+$ be a continuously differentiable and invertible function, such that $\varphi'(t) > 0, \forall t > \frac{1}{2}$. Let us denote by $\varphi(x)$ the coordinatewise application of the function $\varphi$, namely $\varphi(x) = [\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_n)]^T$. Using this, the system which defines the central path (3) can be written in the following form:

$$
-Mx + s = q, \\
x, s > 0, \\
\varphi \left( \frac{xs}{\mu} \right) = \varphi(e),
$$

(4)

For a strictly feasible starting point $(x, s)$ we want to find search directions $\Delta x$ and $\Delta s$ such that

$$
-M(x + \Delta x) + (s + \Delta s) = q, \\
\varphi \left( \frac{xs}{\mu} + \frac{x\Delta s + s\Delta x + \Delta x\Delta s}{\mu} \right) = \varphi(e),
$$

Neglecting the quadratic term $\Delta x\Delta s$ and using Taylor’s theorem we obtain:

$$
\varphi \left( \frac{xs}{\mu} \right) + \varphi' \left( \frac{xs}{\mu} \right) \left( \frac{x\Delta s + s\Delta x}{\mu} \right) = \varphi(e),
$$

which is equivalent to the equation

$$
x\Delta s + s\Delta x = \mu \frac{\varphi(e) - \varphi \left( \frac{xs}{\mu} \right)}{\varphi' \left( \frac{xs}{\mu} \right)}.
$$
Thus, we obtain the following transformed Newton system:

\[-M\Delta x + \Delta s = 0,\]

\[S\Delta x + X\Delta s = \mu \frac{\varphi(e) - \varphi\left(\frac{xs}{\mu}\right)}{\varphi'\left(\frac{xs}{\mu}\right)},\] (5)

hence

\[a_{\varphi} = \mu \frac{\varphi(e) - \varphi\left(\frac{xs}{\mu}\right)}{\varphi'\left(\frac{xs}{\mu}\right)}.\]

We can see that depending on the used functions \(\varphi\) we can have different vectors \(a_{\varphi}\).

The following functions \(\varphi\) have been used in the literature:

- \(\varphi(t) = t\) yields \(a_{\varphi} = \mu e - xs\) introduced by Roos et al. [58], used in Mizuno-Todd-Ye PC IPA [34] and in many other IPAs.
- \(\varphi(t) = \sqrt{t}\) yields \(a_{\varphi} = 2(\sqrt{\mu xs} - xs)\) introduced by Darvay [17], used in Kheirfam’s PC IPA [39].
- \(\varphi(t) = t - \sqrt{t}\) introduced by Darvay, Papp, Takács [22].
- \(\varphi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}\) proposed by Kheirfam, Haghighi [40].

In the last two cases the authors used only the right-hand side of the scaled system, because they introduced small-update IPAs. In the following, we will present a method for determining the scaled corrector and predictor systems. Consider the following notations:

\[v = \sqrt{\frac{xs}{\mu}}, \quad d = \sqrt{\frac{x}{s}}, \quad d_x = \frac{d^{-1} \Delta x}{\sqrt{\mu}} = \frac{v \Delta x}{x}, \quad d_s = \frac{d \Delta s}{\sqrt{\mu}} = \frac{v \Delta s}{s}.\] (6)

Using these notations we have

\[\Delta x = \frac{x d_x}{v} \quad \text{and} \quad \Delta s = \frac{s d_s}{v}.\]

Substituting these in the second equation of system (5) we obtain:

\[\frac{xs d_x}{v} + \frac{xs d_s}{v} = \mu \frac{\varphi(e) - \varphi\left(\frac{xs}{\mu}\right)}{\varphi'\left(\frac{xs}{\mu}\right)}.\] (7)
Hence, the scaled system of the transformed Newton system (5) is the following:

\[- \bar{M} d_x + d_s = 0,\]
\[ d_x + d_s = p_v, \tag{8} \]

where \( \bar{M} = DMD, \ D = \text{diag} (d) \) and

\[ p_v = \frac{\varphi(e) - \varphi(v^2)}{v \varphi'(v^2)}. \]

Using Theorem 2.2 and Corollary 2.1 it can be shown that the scaled transformed Newton system (8) has unique solution.

Depending on the functions \( \varphi \) vector \( p_v \) can have different values:

- \( \varphi(t) = t \) yields \( p_v = v^{-1} - v; \)
- \( \varphi(t) = \sqrt{t} \) yields \( p_v = 2(e - v); \)
- \( \varphi(t) = t - \sqrt{t} \) yields \( p_v = \frac{2(v - v^2)}{2v - e}; \)
- \( \varphi(t) = \frac{\sqrt{t}}{2(1 + \sqrt{t})} \) yields \( p_v = e - v^2. \)

In the following subsection we give a unification of scaled predictor and scaled corrector systems in case of PC IPAs.

### 3.1 New general framework for determining search directions in case of PC IPAs

In this subsection we introduce a unification of the scaled systems in case of PC IPAs for sufficient LCPs, which is a novelty of this paper.

Firstly we determine the scaled corrector system, which coincides with system (8).

This system has the following solution:

\[ d^c_x = (I + \bar{M})^{-1} p_v, \quad d^c_s = \bar{M}(I + \bar{M})^{-1} p_v. \]

We have seen that depending on the function \( \varphi \) vector \( p_v \) has different values. Using

\[ \Delta^c x = \frac{x d^c_x}{v} \quad \text{and} \quad \Delta^c s = \frac{s d^c_s}{v}. \]
the $\Delta^c x$ and $\Delta^c s$ search directions can be easily calculated.

In order to obtain the scaled predictor system, we decompose $a_\varphi$ in the transformed Newton system (5) in the following way:

$$ a_\varphi = f(x, s, \mu) + g(x, s), $$

where $f(x, s, 0) = 0$. Since, we would like to make as greedy predictor step as possible, we set $\mu = 0$ in this decomposition.

Then, we obtain:

$$ -M \Delta x + \Delta s = 0, $$
$$ S \Delta x + X \Delta s = g(x, s). \quad (9) $$

Using equation (7) we obtain

$$ g(x, s) = S \Delta x + X \Delta s = \frac{x s d_x}{v} + \frac{x s d_s}{v}. $$

Hence, we have the following scaled predictor system:

$$ -\bar{M} d_x + d_s = 0, $$
$$ d_x + d_s = \frac{v g(x, s)}{x s}, \quad (10) $$

where $\bar{d} = DMD$ and which has the solution:

$$ d^p_x = (I + \bar{M})^{-1} \frac{v g(x, s)}{x s}, \quad d^p_s = \bar{M} (I + \bar{M})^{-1} \frac{v g(x, s)}{x s}. $$

Using

$$ \Delta^p x = \frac{x d^p_x}{v} \quad \text{and} \quad \Delta^p s = \frac{s d^p_s}{v} $$

the $\Delta^p x$ and $\Delta^p s$ search directions can be easily calculated. This framework shows that in order to introduce PC IPAs we have to decompose the right-hand side of the nonlinear equation of the transformed Newton-system into two parts: the one which depends and the other which does not depend on the positive parameter $\mu$. Note that this decomposition is not trivial and we have no guaranty that such decomposition exists when we use AET for all functions $\varphi$ that can be used in the AET for short step IPAs.
4 New PC IPA for sufficient LCPs based on a new search direction

In this section we propose a PC IPA based on the directions obtained by using the function $\varphi(t) = t - \sqrt{t}$ proposed in [22]. The introduced PC IPA uses these directions in the both predictor and corrector steps. Considering this search direction we obtain the following decomposition:

$$a_\varphi = \frac{\sqrt{\mu} x_s}{2\sqrt{x_s} - \sqrt{\mu} e} - x_s,$$

hence $f(x, s, \mu) = \frac{\sqrt{\mu} x_s}{2\sqrt{x_s} - \sqrt{\mu} e}$, which satisfies the condition $f(x, s, 0) = 0$ and $g(x, s) = -x_s$. In this case, the transformed Newton system (5) is the following:

$$-M \Delta x + \Delta s = 0,$$

$$S \Delta x + X \Delta s = \frac{\sqrt{\mu} x_s}{2\sqrt{x_s} - \sqrt{\mu} e} - x_s.$$  \hfill (11)

We define the following proximity measure, which is used to measure the distance of the iterates $(x, s)$ from the central path.

$$\delta(x, s, \mu) := \delta(v) := \frac{\|p_v\|}{2} = \frac{\|v - v^2\|}{2v - e}.$$  \hfill (12)

Using this, we give the $\tau$-neighbourhood of the central path in the following way:

$$\mathcal{N}(\tau, \mu) := \{(x, s) \in \mathcal{F}^+ : \delta(x, s, \mu) \leq \tau\},$$

where $\tau$ is a threshold parameter. There are several IPAs, that use firstly corrector steps and after that predictor step (Potra [56]). This can be explained by the fact that after a corrector step we reach a proper neighbourhood of the central path. In the literature, several authors call these methods corrector-predictor algorithms. Our algorithm also performs firstly a corrector step and after that a predictor one. In this way, the algorithm starts with $(x, s) \in \mathcal{N}(\tau, \mu)$, which holds in case of the starting points $(x^0, s^0)$, because $\delta(x^0, s^0, \mu) \leq \tau$. The algorithm performs corrector and predictor steps. In a corrector step we obtain $d^c_x$ and $d^c_s$ by solving the scaled corrector system:

$$-\bar{M}d^c_x + d^c_s = 0,$$

$$d^c_x + d^c_s = \frac{2(v - v^2)}{2v - e},$$  \hfill (13)
where we used the scaling notations considered in Section 3 and $\bar{M} = DMD$ and $D = diag(d)$. Then, using Theorem 2.2 and Corollary 2.1 it can be shown that the scaled transformed Newton system (13) has unique solution:

$$d_c^x = (I + \bar{M})^{-1} \frac{2(v - v^2)}{2v - e}, \quad d_c^s = \bar{M}(I + \bar{M})^{-1} \frac{2(v - v^2)}{2v - e}.$$  

Using

$$\Delta^c x = \frac{x d_c^x}{v} \quad \text{and} \quad \Delta^c s = \frac{s d_c^s}{v}$$

the $\Delta^c x$ and $\Delta^c s$ search directions can be easily calculated. Let

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

In the predictor step we define the following notations:

$$v^+ = \sqrt{\frac{x^+ s^+}{\mu}}, \quad d^+ = \frac{x^+}{s^+}, \quad D_+ = diag(d^+), \quad \bar{M}_+ = D_+ MD_+.$$  

The scaled predictor system in this case is the following:

$$-\bar{M}_+ d^p_x + d^p_s = 0,$$

$$d^p_x + d^p_s = -v^+.$$  

(14)

which has the solution

$$d^p_x = -(I + \bar{M}_+)^{-1} v^+ \quad \text{and} \quad d^p_s = -\bar{M}_+(I + \bar{M}_+)^{-1} v^+.$$  

Then, using

$$\Delta^p x = \frac{x^+}{v^+} d^p_x \quad \text{and} \quad \Delta^p s = \frac{s^+}{v^+} d^p_s,$$

the search directions $\Delta^p x$ and $\Delta^p s$ can be easily calculated. The point after a predictor step is

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

where $\theta \in (0, \frac{1}{2})$ is the update parameter.

We summarize the new PC IPA as in Figure 1.
Let $\epsilon > 0$ be the accuracy parameter, $0 < \theta < 1$ the update parameter (default $\theta = \frac{1}{5(1+2\kappa)\sqrt{n}}$) and $\tau$ the proximity parameter (default $\tau = \frac{1}{2(3+4\kappa)}$).

Assume that for $(x^0, s^0)$ the $(x^0)^T s^0 = n\mu^0$, $\mu^0 > 0$ holds such that $\delta(x^0, s^0, \mu^0) \leq \tau$ and $\frac{x^0 s^0}{\mu^0} > \frac{1}{4}$.

begin
  $x := x^0$; $s := s^0$; $\mu := \mu^0$;
  while $x^T s > \epsilon$ do begin
    calculate $(\Delta^c x, \Delta^c s)$ from system (13) and let $x^+ = x + \Delta^c x$, $s^+ = s + \Delta^c s$;
    calculate $(\Delta^p x, \Delta^p s)$ from system (14) and let $x^p = x^+ + \theta \Delta^p x$, $s^p = s^+ + \theta \Delta^p s$;
    $\mu^p = (1 - \theta)\mu$;
    $x := x^p$, $s := s^p$, $\mu := \mu^p$;
  end
end.

Figure 1: New PC IPA for sufficient LCPs

5 Analysis of the PC IPA

In this section we present the analysis of the introduced PC IPA for sufficient LCP. Firstly, we give two important results based on the properties of the matrix $M$. Throughout this section $M$ is a $P_*(\kappa)$-matrix. From

$$-M\Delta^p x + \Delta^p s = 0,$$

we obtain

$$(1 + 4\kappa) \sum_{i \in I_+} \Delta^p x_i \Delta^p s_i + \sum_{i \in I_-} \Delta^p x_i \Delta^p s_i \geq 0,$$

(15)

where $I_+ = \{i : \Delta^p x_i \Delta^p s_i > 0\}$ and $I_- = \{i : \Delta^p x_i \Delta^p s_i < 0\}$. Using (6) we have

$$d^p_x d^p_s = \frac{\Delta^p x \Delta^p s}{\mu}.$$

Hence, we can write (15) in the following way:

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

(16)

Operations Research Reports No. 2018-03
We prove a lemma, similar to that of Lemma 1 in the paper of Kheirfam [39]. The main difference is that we use different function $\varphi$. From this it follows that some steps of the proof are different.

**Lemma 5.1.** One has

$$
\|d^p_x d^p_s\| \leq \frac{n(1 + 2\kappa)(1 + 2\delta^+)^2}{2},
$$

where $\delta^+ = \delta(x^+, s^+, \mu) = \|v^+ - (v^+)^2\|_{2v^+ - e}^2$.

**Proof.** We use the second equation of the scaled predictor system (14) and we obtain the following:

$$
\sum_{i \in I_+} d^p_{x_i} d^p_{s_i} \leq \frac{1}{4} \sum_{i \in I_+} (d^p_{x_i} + d^p_{s_i})^2 \leq \frac{1}{4} \sum_{i=1}^n (d^p_{x_i} + d^p_{s_i})^2 = \frac{1}{4} \|d^p_x + d^p_s\|^2 = \frac{\|v^+\|^2}{4}.
$$

Using this relation and (16) we get

$$
\|v^+\|^2 = \|d^p_x + d^p_s\|^2 = \|d^p_x\|^2 + \|d^p_s\|^2 + 2 \left( \sum_{i \in I_+} d^p_{x_i} d^p_{s_i} + \sum_{i \in I_-} d^p_{x_i} d^p_{s_i} \right) \\
\geq \|d^p_x\|^2 + \|d^p_s\|^2 - 8\kappa \sum_{i \in I_+} d^p_{x_i} d^p_{s_i} \\
\geq \|d^p_x\|^2 + \|d^p_s\|^2 - 2\kappa \|v^+\|^2.
$$

This means that $\|d^p_x\|^2 + \|d^p_s\|^2 \leq (1 + 2\kappa)\|v^+\|^2$. Moreover, we give an upper bound for $\|v^+\|$ depending on $\delta^+$ and $n$. For this, let us use the following notation $\sigma^+ = \|e - v^+\|$, the centrality measure used in [39]. We have

$$
\|v^+\| = \|v^+ - e + e\| \leq \|v^+ - e\| + \|e\| = \sigma^+ + \sqrt{n} \leq \sqrt{n}(\sigma^+ + 1) (17)
$$

Furthermore,

$$
\delta^+ = \left\| \frac{v^+ - (v^+)^2}{2v^+ - e} \right\| = \|v^+\| \cdot \|e - v^+\| \\
\geq \frac{\|v^+ - \frac{e}{2}\| \cdot \|e - v^+\|}{2\|v^+ - \frac{e}{2}\|} = \frac{\sigma^+}{2},
$$

(18)

hence $\sigma^+ \leq 2\delta^+$. Using (17) and (18) we obtain

$$
\|v^+\| \leq \sqrt{n}(1 + 2\delta^+).
$$

(19)
Hence,
\[
\|d^p_x d^p_s\| \leq \|d^p_x\|\|d^p_s\| \leq \frac{1}{2} (\|d^p_x\|^2 + \|d^p_s\|^2)
\leq \frac{1}{2} (1 + 2\kappa) \|v^+\|^2 \leq \frac{n(1 + 2\kappa)(1 + 2\delta^+)^2}{2},
\]
which yields the result.

Let
\[
q_v = d^c_x - d^c_s. \tag{20}
\]
Then,
\[
d^c_x = \frac{p_v + q_v}{2}, \quad d^c_s = \frac{p_v - q_v}{2} \quad \text{and} \quad d^c_x d^c_s = \frac{p_v^2 - q_v^2}{4}. \tag{21}
\]

**Lemma 5.2.** The following inequality holds:
\[
\|q_v\| \leq 2\sqrt{1 + 4\kappa} \delta^2,
\]
where \(\delta = \delta(x, s, \mu)\).

**Proof.** Similarly to (16) we obtain
\[
(1 + 4\kappa) \sum_{i \in I^c_+} d^c_x_i d^c_s_i + \sum_{i \in I^c_-} d^c_x_i d^c_s_i \geq 0, \tag{22}
\]
where \(I^c_+ = \{i : \Delta^c x_i \Delta^c s_i > 0\}\) and \(I^c_- = \{i : \Delta^c x_i \Delta^c s_i < 0\}\). Using (22) we have
\[
(d^c_x)^T d^c_s \geq -4\kappa \sum_{i \in I^c_+} d^c_x_i d^c_s_i. \tag{23}
\]

Moreover,
\[
\sum_{i \in I^c_+} d^c_x_i d^c_s_i = \frac{1}{4} \sum_{i \in I^c_+} (d^c_x_i + d^c_s_i)^2 - \frac{1}{4} \sum_{i \in I^c_-} (d^c_x_i + d^c_s_i)^2
\leq \frac{1}{4} \sum_{i \in I^c_+} (d^c_x_i + d^c_s_i)^2 \leq \frac{1}{4} \sum_{i=1}^n (d^c_x_i + d^c_s_i)^2
= \frac{1}{4} \|d^c_x + d^c_s\|^2 = \delta^2. \tag{24}
\]
Using (23) and (24) we obtain the following inequality:
\[(d^c_x)^T d^c_s \geq -4\kappa \delta^2.\]  
(25)

Using (25) and (20) we have

\[4\delta^2 = \|d^c_x + d^c_s\|^2 = \|d^c_x - d^c_s\|^2 + 4(d^c_x)^T d^c_s \geq \|q_v\|^2 - 16\kappa \delta^2.\]  
(26)

From (26) we obtain

\[\|q_v\| \leq 2\sqrt{1 + 4\kappa \delta^2}.\]  
(27)

### 5.1 The predictor step

The next lemma gives a sufficient condition for the strict feasibility of the predictor step. Note that the introduction of function \(u\) which depends on \(\delta^+, \theta\) and \(n\) plays key role in this part of the analysis, because the condition \(u(\delta^+, \theta, n) > 0\) is necessary to prove the feasibility of the predictor step. Later we will present how we have fixed the parameters and we will also show how this influences the values of the function \(u\).

**Lemma 5.3.** Let \((x^+, s^+) > 0\) and \(\mu > 0\) such that \(\delta^+ := \delta(x^+, s^+, \mu) < \frac{1}{2}\). Furthermore, let \(0 < \theta < 1\). Let \(x^p = x^+ + \theta \Delta^p x\), \(s^p = s^+ + \theta \Delta^p s\) be the iterates after a predictor step. Then, \(x^p, s^p > 0\) if \(u(\delta^+, \theta, n) > 0\), where

\[u(\delta^+, \theta, n) := (1 - 2\delta^+)^2 - \frac{n(1 + 2\kappa)\theta^2(1 + 2\delta^+)^2}{2(1 - \theta)}.\]

**Proof.** Firstly, we consider the following notations:

\[x^p(\alpha) = x^+ + \alpha \theta \Delta^p x, \quad s^p(\alpha) = s^+ + \alpha \theta \Delta^p s,\]

for \(0 \leq \alpha \leq 1\). We have

\[x^p(\alpha) = \frac{x^+}{v^+}(v^+ + \alpha \theta d^p_x), \quad s^p(\alpha) = \frac{s^+}{v^+}(v^+ + \alpha \theta d^p_s).\]

Using the second equation of the scaled predictor system (14) we get the following:

\[x^p(\alpha)s^p(\alpha) = \mu(v^+ + \alpha \theta d^p_x)(v^+ + \alpha \theta d^p_s) = \mu \left((v^+)^2 + \alpha \theta v^+(d^p_x + d^p_s) + \alpha^2 \theta^2 d^p_x d^p_s \right) = \mu \left((1 - \alpha \theta)(v^+)^2 + \alpha^2 \theta^2 d^p_x d^p_s \right).\]  
(28)
From (28) we have the following:

\[
\begin{align*}
\min \left( \frac{x_p(\alpha)s_p(\alpha)}{\mu(1-\alpha \theta)} \right) & \geq \min \left( (v^+)^2 + \frac{\alpha^2 \theta^2}{1-\alpha \theta} d_x^p d_s^p \right) \\
& \geq \min \left( (v^+)^2 - \frac{\alpha^2 \theta^2}{1-\alpha \theta} \|d_x^p d_s^p\|_{\infty} e \right) \\
& \geq \min \left( (v^+)^2 - \frac{\theta^2}{1-\theta} \|d_x^p d_s^p\|_{\infty} e \right)
\end{align*}
\]

(29)

The last inequality follows from the fact that

\[ f(\alpha) = \frac{\alpha^2 \theta^2}{1-\alpha \theta} \]

is strictly increasing for \(0 \leq \alpha \leq 1\) and each fixed \(0 < \theta < \frac{1}{2} \). Moreover, using

\[ |1 - v_i^+| \leq \|e - v^+\|, \forall i = 1, \ldots, n \]

we have

\[ 1 - \sigma^+ \leq v_i \leq 1 + \sigma^+, \forall i = 1, \ldots, n. \]

Using this and (18) we have

\[ \min (v^+)^2 \geq (1 - \sigma^+)^2 \geq (1 - 2\delta^+)^2 \]

(30)

From Lemma 5.1 and (30) we obtain

\[
\begin{align*}
\min \left( \frac{x_p(\alpha)s_p(\alpha)}{\mu(1-\alpha \theta)} \right) & \geq (1 - 2\delta^+)^2 - \frac{n(1 + 2\kappa)\theta^2(1 + 2\delta^+)^2}{2(1-\theta)} \\
& = u(\delta^+, \theta, n) > 0.
\end{align*}
\]

(31)

This yields \(x_p(\alpha)s_p(\alpha) > 0\) for \(0 \leq \alpha \leq 1\). Therefore, \(x_p(\alpha)\) and \(s_p(\alpha)\) do not change sign on \(0 \leq \alpha \leq 1\). Since \(x_p(0) = x^+ > 0\) and \(s_p(0) = s^+ > 0\), we can conclude that \(x_p(1) = x^p > 0\) and \(s_p(1) = s^p > 0\), which proves the lemma.

Therefore, using the notations considered in (6) let us introduce

\[ v^p = \sqrt{\frac{x_p s_p}{\mu^p}}, \]

where \(\mu^p = (1-\theta)\mu\). Substituting \(\alpha = 1\) in (28) and (31) we get

\[ (v^p)^2 = (v^+)^2 + \frac{\theta^2}{1-\theta} d_x^p d_s^p, \]

(32)
\[ \min (v^P)^2 \geq u(\delta^+, \theta, n) > 0. \]  

(33)

In the following lemma we investigate the effect of a predictor step and the update of \( \mu \) on the proximity measure. In this lemma the condition \( u(\delta^+, \theta, n) > \frac{1}{4} \) should hold, because due to the used function \( \varphi(t) = t - \sqrt{t} \) in the determination of the search directions, we have to ensure that in each iteration of the algorithm, the components of the \( v \) vectors of the scaled space are greater than \( \frac{1}{2} \). Using this and (33) follows the condition \( u(\delta^+, \theta, n) > \frac{1}{4} \).

Lemma 5.4. Let \( \delta^+ := \delta(x^+, s^+, \mu) < \frac{1}{2}, \mu^p = (1 - \theta)\mu, \) where \( 0 < \theta < 1, u(\delta^+, \theta, n) > \frac{1}{4} \) and let \( x^p \) and \( s^p \) denote the iterates after a predictor step. Then, \( v^p > \frac{1}{2}e \) and

\[ \delta^p := \delta(x^p, s^p, \mu^p) \leq \frac{\sqrt{u(\delta^+, \theta, n)} ((3 + 4\kappa)\delta^2 + (1 - 2\delta^+)^2 - u(\delta^+, \theta, n))}{2u(\delta^+, \theta, n) + \sqrt{u(\delta^+, \theta, n)} - 1}, \]

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

Proof. Since \( u(\delta^+, \theta, n) > \frac{1}{4} > 0 \), from Lemma 5.3 we have that \( x^p, s^p > 0 \), hence the predictor step is strictly feasible. Using (33) we have

\[ \min (v^p) \geq \sqrt{u(\delta^+, \theta, n)} > \frac{1}{2}, \]

which proves the first part of the lemma. Furthermore,

\[ \delta^p := \left\| \frac{v^p - (v^p)^2}{2v^p - e} \right\| \]

\[ = \left\| \frac{v^p (e - v^p)(e + v^p)}{(2v^p - e)(e + v^p)} \right\| = \left\| \frac{v^p (e - (v^p)^2)}{(2v^p - e)(e + v^p)} \right\|. \]

(34)

Let \( h : \left( \frac{1}{2}, \infty \right) \to \mathbb{R}, h(t) = \frac{t}{(2t - 1)(1 + t)}. \) This function is decreasing with
respect to $t$. Using this, (32), (33) and (34) we obtain the following inequality:

$$\delta^p = \left\| \frac{v^p (e - (v^p)^2)}{(2v^p - e)(e + v^p)} \right\| \leq \frac{\min(v^p)}{(2 \min(v^p) - 1)(1 + \min(v^p))} \left\| e - (v^p)^2 \right\|$$

$$\leq \frac{\sqrt{u(\delta^+, \theta, n)}}{(2 \sqrt{u(\delta^+, \theta, n) - 1})(1 + \sqrt{u(\delta^+, \theta, n)})} \left\| e - (v^+)^2 - \frac{\theta^2}{1 - \theta}d^p_xd^p_s \right\|$$

$$\leq \frac{\sqrt{u(\delta^+, \theta, n)}}{(2 \sqrt{u(\delta^+, \theta, n) - 1})(1 + \sqrt{u(\delta^+, \theta, n)})} \cdot \left( \left\| e - (v^+)^2 \right\| + \frac{\theta^2}{1 - \theta} \left\| d^p_xd^p_s \right\| \right). \quad (35)$$

We will give an upper bound for $\left\| e - (v^+)^2 \right\|$. Using the definition of $v^+ = \sqrt{\frac{x^+s^+}{\mu}}$, (6), (20), (21) and Lemma 5.2 we have

$$\left\| e - (v^+)^2 \right\| = \left\| (v + d^c_x)(v + d^c_s) - e \right\| = \left\| v^2 + v(d^c_x + d^c_s) - e + d^c_xd^c_s \right\|$$

$$\leq \left\| v^2 + vp_v - e \right\| + \left\| \frac{p^2_v - q^2_v}{4} \right\|. \quad (36)$$

Furthermore,

$$v^2 + vp_v - e = v^2 + \frac{2v^2(e - v)}{2v - e} - e$$

$$= \frac{(v - e)^2}{2v - e} \leq \frac{(v - e)^2v^2}{(2v - e)^2} = \frac{p^2_v}{4}. \quad (37)$$

Using (36), (37), Lemma 5.2, the $\|x^2\| \leq \|x\|^2$ and the triangle inequality we have

$$\left\| e - (v^+)^2 \right\| \leq \left\| v^2 + vp_v - e \right\| + \left\| \frac{p^2_v - q^2_v}{4} \right\|$$

$$\leq \frac{\|p_v\|^2}{4} + \frac{\|p_v\|^2}{4} + \frac{\|q_v\|^2}{4} \leq (3 + 4\kappa)\delta^2. \quad (38)$$
From (35), (38) and Lemma 5.1 we obtain

\[
\delta^p \leq \frac{\sqrt{u(\delta^+, \theta, n)}}{(2\sqrt{u(\delta^+, \theta, n)} - 1)} \left( \|e - (v^+)^2\| + \frac{\theta^2}{1 - \theta} \|d_x^p d_s^p\| \right)
\]

\[
\leq \frac{\sqrt{u(\delta^+, \theta, n)} ((3 + 4\kappa)\delta^2 + (1 - 2\delta^+)^2 - u(\delta^+, \theta, n))}{2u(\delta^+, \theta, n) + \sqrt{u(\delta^+, \theta, n)} - 1},
\]  

(39)

which proves the second statement of the lemma.

5.2 The corrector step

The corrector part of the introduced PC IPA is similar to the classical small-update IPAs. Hence, the following lemmas related to the corrector steps can be easily derived from two unpublished manuscripts related to small-update IPAs for solving sufficient LCPs and \(P_*(\kappa)\)-LCPs over Cartesian product of symmetric cones [4, 21]. The next lemma proves the strict feasibility of the full-Newton step.

Lemma 5.5. If \(\delta := \delta(x, s, \mu) < \frac{1}{\sqrt{1+4\kappa}}\) and \(v > \frac{1}{2}e\), then \(x^+ > 0\) and \(s^+ > 0\).

The following lemma gives an upper bound for the proximity measure of the iterates obtained by a full-Newton step.

Lemma 5.6. Let \(\delta = \delta(x, s, \mu) < \frac{1}{2\sqrt{1+4\kappa}}\) and \(v > \frac{1}{2}e\). Then, \(v^+ > \frac{1}{2}e\) and

\[
\delta^+ := \delta(x^+, s^+, \mu) \leq \frac{\sqrt{1 - (1 + 4\kappa)\delta^2}}{2(1 - (1 + 4\kappa)\delta^2) + \sqrt{1 - (1 + 4\kappa)\delta^2} - 1} (3 + 4\kappa)\delta^2.
\]

Furthermore,

\[
\delta^+ := \delta(x^+, s^+, \mu) < \frac{3 - \sqrt{3}}{2} (3 + 4\kappa)\delta^2.
\]

The next lemma gives an upper bound for the duality gap after a full-Newton step.

Lemma 5.7. We assume that we obtained \(x^+\) and \(s^+\) after a full-Newton step. Then,

\[
(x^+)^T s^+ \leq \mu(n + 2\delta^2).
\]

Furthermore, if \(\delta < \frac{1}{2(1+4\kappa)}\) and \(n \geq 4\), then

\[
(x^+)^T s^+ < \frac{9}{8} \mu n.
\]
The next lemma gives an upper bound for the duality gap after a main iteration.

**Lemma 5.8.** Let \((x^+, s^+) > 0\) and \(\mu > 0\) such that \(\delta^+ := \delta(x^+, s^+, \mu) < \frac{1}{2}\) and \(0 < \theta < 1\). If \(\delta < \frac{1}{2(1+4\kappa)}\) and \(x^p\) and \(s^p\) are the iterates obtained after the predictor step of the algorithm, then

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

**Proof.** From (32) and the definition of \(v^p\) we obtain

\[
(x^p)^T s^p = \mu^p e^T (v^p)^2 = \mu e^T \left((1 - \theta)(v^+)^2 + \theta^2 d^p_x d^p_s\right) = (1 - \theta)(x^+)^T s^+ + \mu \theta^2 (d^p_x)^T d^p_s. \tag{40}
\]

Multiplying the second equation of (14) by \((d^p_x)^T\) and by \((d^p_s)^T\), respectively, and summing the obtained two equations we get the following:

\[
(d^p_x)^T d^p_s = \frac{(x^+)^T s^+}{2\mu} - \frac{\|d^p_x\|^2 + \|d^p_s\|^2}{2} \leq \frac{(x^+)^T s^+}{2\mu}. \tag{41}
\]

From (40) and (41) we obtain

\[
(x^p)^T s^p \leq \left(1 - \theta + \frac{\theta^2}{2}\right)(x^+)^T s^+.
\]

If \(0 < \theta < 1\), then

\[
1 - \theta + \frac{\theta^2}{2} < 1 - \frac{\theta}{2}.
\]

Using this and Lemma 5.7 we obtain:

\[
(x^p)^T s^p \leq \left(1 - \theta + \frac{\theta^2}{2}\right)(x^+)^T s^+ \leq \left(1 - \frac{\theta}{2}\right)(x^+)^T s^+ < \frac{9n\mu p}{8(1-\theta)},
\]

which proves the lemma.

### 5.3 Fixing the parameter

In this subsection we fix the parameters \(\tau\) and \(\theta\) to guarantee that after a corrector and a predictor step, the proximity measure will not exceed the proximity parameter before.
Let \((x, s)\) be the iterate at the start of an iteration with \(x > 0\) and \(s > 0\) such that \((x, s) \in \mathcal{N}(\tau, \mu)\). After a corrector step, by Lemma 5.6 one has:

\[
\delta^+ := \delta(x^+, s^+, \mu) < \frac{3 - \sqrt{3}}{2} (3 + 4\kappa)\delta^2.
\]

The right-hand side of the above inequality is monotonically increasing with respect to \(\delta\), which implies:

\[
\delta^+ \leq \frac{3 - \sqrt{3}}{2} (3 + 4\kappa)\tau^2 = \omega(\tau).
\]

Following a predictor step and a \(\mu\)-update, by Lemma 5.4 we have,

\[
\delta^p := \delta(x^p, s^p, \mu^p) \leq \frac{\sqrt{u(\delta^+, \theta, n)} ((3 + 4\kappa)\delta^2 + (1 - 2\delta^+)^2 - u(\delta^+, \theta, n))}{2u(\delta^+, \theta, n) + \sqrt{u(\delta^+, \theta, n)} - 1},
\]

where \(\delta := \delta(x, s, \mu)\). It can be verified that \(u(\delta^+, \theta, n)\) is decreasing with respect to \(\delta^+\). In this way, we obtain \(u(\delta^+, \theta, n) \geq u(\omega(\tau), \theta, n)\). We have seen in Lemma 5.4 that the function \(h(t) = \frac{t}{(2t-1)(1+t)}, t > \frac{1}{2}\) is decreasing with respect to \(t\), therefore

\[
h(u(\delta^+, \theta, n)) \leq h(u(\omega(\tau), \theta, n)).
\]

Using that \(\delta < \tau\), \(\delta^+ < \omega(\tau)\) and

\[
(1 - 2\delta^+)^2 - u(\delta^+, \theta, n) = \frac{n(1 + 2\kappa) \theta^2 (1 + 2\delta^+)^2}{2(1 - \theta)},
\]

which is increasing with respect to \(\delta^+\), we obtain:

\[
\frac{\sqrt{u(\delta^+, \theta, n)} ((3 + 4\kappa)\delta^2 + (1 - 2\delta^+)^2 - u(\delta^+, \theta, n))}{2u(\delta^+, \theta, n) + \sqrt{u(\delta^+, \theta, n)} - 1} \leq \frac{\sqrt{u(\omega(\tau), \theta, n)} ((3 + 4\kappa)\tau^2 + (1 - 2\omega(\tau))^2 - u(\omega(\tau), \theta, n))}{2u(\omega(\tau), \theta, n) + \sqrt{u(\omega(\tau), \theta, n)} - 1}.
\]

(42)

To keep \(\delta^p \leq \tau\), it suffices that

\[
\frac{\sqrt{u(\omega(\tau), \theta, n)} ((3 + 4\kappa)\tau^2 + (1 - 2\omega(\tau))^2 - u(\omega(\tau), \theta, n))}{2u(\omega(\tau), \theta, n) + \sqrt{u(\omega(\tau), \theta, n)} - 1} \leq \tau.
\]
If we set \( \tau = \frac{1}{2(3+4\kappa)} \) and \( \theta = \frac{1}{5(1+2\kappa)\sqrt{n}} \), the above inequality holds. This means that \( x, s > 0 \) and \( \delta(x, s, \mu) \leq \frac{1}{2(3+2\kappa)} < \frac{1}{2\sqrt{1+4\kappa}} \) are maintained during the algorithm. Thus, the algorithm is well-defined. Moreover, one has

\[
\begin{align*}
    u(\delta^+, \theta, n) &= (1 - 2\delta^+)^2 - \frac{n(1 + 2\kappa)\theta^2(1 + 2\delta^+)^2}{2(1 - \theta)} \\
    &\geq (1 - 2\omega(\tau))^2 - \frac{n(1 + 2\kappa)\theta^2(1 + 2\omega(\tau))^2}{2(1 - \theta)} \geq 0.25,
\end{align*}
\]

so we can conclude that the predictor step is strictly feasible.

### 5.4 Complexity bound

In the following lemma we give an upper bound for the number of iterations produced by the algorithm.

**Lemma 5.9.** Let \( x^0 \) and \( s^0 \) be strictly feasible, \( \theta = \frac{1}{5(1+2\kappa)\sqrt{n}} \), \( \mu^0 = \frac{(x^0)^T s^0}{n} \) and \( \delta(x^0, s^0, \mu^0) \leq \tau \). Moreover, let \( x^k \) and \( s^k \) be the iterates obtained after \( k \) iterations. Then, \((x^k)^T s^k \leq \epsilon\) for

\[
k \geq 1 + \left\lceil \frac{1}{\theta} \log \frac{9 (x^0)^T s^0}{8 \epsilon} \right\rceil.
\]

**Proof.** It follows from Lemma 5.8 that

\[
(x^k)^T s^k < \frac{9n\mu^k}{8(1 - \theta)} = \frac{9n(1 - \theta)^{k-1}\mu^0}{8} = \frac{9(1 - \theta)^{k-1} (x^0)^T s^0}{8}.
\]

Then, the inequality \((x^k)^T s^k \leq \epsilon\) holds if

\[
\frac{9(1 - \theta)^{k-1} (x^0)^T s^0}{8} \leq \epsilon.
\]

Taking logarithms, we obtain

\[
(k - 1) \log(1 - \theta) + \log \frac{9 (x^0)^T s^0}{8} \leq \log \epsilon.
\]

Since \( \log(1 + \theta) \leq \theta, \theta \geq -1 \), we conclude that the above inequality holds if

\[
-\theta(k - 1) + \log \frac{9 (x^0)^T s^0}{8} \leq \log \epsilon.
\]

This implies the result.
Theorem 5.10. Let \( \tau = \frac{1}{2(3+4\kappa)} \) and \( \theta = \frac{1}{5(1+2\kappa)^\sqrt{n}} \). Then, Algorithm 1 is well defined and the algorithm requires at most
\[
O \left( (1 + 2\kappa)^\sqrt{n} \log \frac{9n\mu^0}{8\epsilon} \right)
\]
iterations. The output is a pair \((x, s)\) satisfying \( x^T s \leq \epsilon \).

6 Numerical results

We implemented the introduced PC IPA based on the \( \varphi(t) = t - \sqrt{t} \) search direction in the C++ programming language [23]. In Theorem 5.10, similarly to the description of our PC IPA, we suggested default values for the parameters \( \theta \) and \( \tau \) that played a key role in the complexity analysis of the algorithm. It is important to mention that our PC IPA is well defined for many different pairs of the parameters \( \theta \) and \( \tau \). Although the default values of the barrier and proximity parameters were essential in theory, the implemented version of our PC IPA used properly selected, but different \( \theta \) and \( \tau \) values. In order to obtain an efficient implementation, we modified our algorithm in the following way. The barrier parameter for the predictor step was calculated as \( \mu = \frac{x^T s}{n} \) and the proximity parameter was \( \epsilon = 10^{-5} \). We used Mehrotra’s heuristics [47] to calculate the barrier parameter for the corrector step. In each iteration of the algorithm we considered the direction determined by the predictor and corrector steps. After that, we calculated the maximal step size to the boundary of the feasible set. We did all computations on a desktop computer with Intel quad-core 2.60 GHz processor and 8 GB RAM.

We tested the algorithm on three families of LCPs. The first two families consist of LCPs with sufficient matrices having positive \( \kappa \) parameters. The third family of LCPs are related to testing copositivity of given matrices.

6.1 Results for LCPs with sufficient matrices

Since \( \varphi(t) = t \) is the identity function, this means that the central path is not transformed. Furthermore, it means that the AET approach does not need to be applied in this case. We can say that the majority of the published IPAs for sufficient LCPs does not use any transformation of the central path. However, the case of the identity map can be considered as a trivial, special subcase of the AET approach. Hence, the complexity analysis of IPAs without transformed central path is simpler than those of using AET technique with nontrivial functions \( \varphi \). After all, it is a natural question whether we gain any advantages by applying nontrivial functions \( \varphi \) in developing IPAs.
It should be mentioned that the theoretical complexity of IPAs based on AET approach coincides with the currently best known complexity results for IPAs. The PC IPA introduced by Illés and Nagy [34] does not use any transformation of the central path and possess \( O \left( (1 + \kappa)^{\frac{3}{2}} \sqrt{n \log \frac{n}{\epsilon}} \right) \) iteration complexity. Some years later, Kheirfam [39] developed a PC IPA, which uses the direction based on function \( \varphi(t) = \sqrt{t} \) and achieved \( O \left( (1 + 2\kappa)\sqrt{n \log \frac{n\mu^0}{\epsilon}} \right) \) result. Wang and Bai [67] proposed IPA for sufficient LCPs based on a different technique of determining search directions, namely considering parametric kernel functions. They obtained \( O \left( (1 + 2\kappa)\sqrt{n \log \frac{n\mu^0}{\epsilon}} \right) \) complexity result. From Theorem 5.10 we can see that there is no significant difference in the theoretical complexity of our algorithm and the other mentioned PC IPAs.

Kheirfam [39] compared his IPA to the ones introduced by Illés and Nagy [34] and Wang and Bai [67] from computational point of view. Most of the reported computational studies of IPAs for LCPs with symmetric positive semidefinite matrices. Kheirfam goes beyond this with selecting nonsymmetric \( P^*(0) \) matrices.

In our computational study we used sufficient \( P^*(\kappa) \)-matrices with \( \kappa > 0 \) generated by Illés and Morapitiy [33]. Up to our best knowledge, these are the first numerical results of IPAs for sufficient LCPs with positive handicap. Our test problems from the given \( M \) sufficient matrices have been generated in the following way:

\[
q := -Me + e.
\]

Table 1 and Table 2 contain the numerical results obtained for \( 10 \times 10 \) and \( 20 \times 20 \) sufficient matrices, respectively. Note that our implementation was developed in a way that we could easily obtain results for variants of our PC IPA using different functions \( \varphi \) in the AET technique. This can be achieved by changing the right hand side of the Newton-system. In our numerical study we compare our algorithm to variants of our PC IPA using AETs based on functions \( \varphi(t) = t \) and \( \varphi(t) = \sqrt{t} \). We can observe that for the small-sized problems the results are similar for all three variants of our PC IPA.

Let us consider Csizmadia’s sufficient matrix given in (2) that has exponentially large \( \kappa \) in the size of the matrix [24]. In spite of this fact we obtained very promising results for our algorithm which uses \( \varphi(t) = t - \sqrt{t} \). These are summarized in Table 3, where \( \kappa’ \) is the known lower bound on the handicap of the matrix.

The results can lead to further research on the topic, because it seems that the practical iteration complexity is significantly better than the theoretical (worst case) guaranty.

We set the maximum number of iterations to 3000. We also tested the
other two variants of PC IPA using AET based on the functions $\varphi(t) = t$ and $\varphi(t) = \sqrt{t}$ for LCPs with $P$-matrix given by Csizmadia [24]. However, these variants of PC IPA could not find $\varepsilon$-optimal solution even for the matrix with $n = 20$.

These computational tests show that the effect of the AET based on a given function $\varphi$ may help to solve computationally challenging LCPs.

### 6.2 LCPs implied by the copositivity tests

A real symmetric matrix $A$ is copositive if and only if the following quadratic function $\mathbf{x}^\top A \mathbf{x}$ is non-negative for any $\mathbf{x} \geq 0$. If this function is strictly positive for any non-zero $\mathbf{x} \geq 0$, then $A$ is strictly copositive. The copositive matrices form a proper (closed, convex, pointed and full-dimensional) cone. Unfortunately, detecting whether given matrix belongs to this cone is an NP-hard problem [26].

The copositive matrices have become very popular in the last two decades because they have very strong modelling power in combinatorial optimization. Indeed, large family of hard optimization problems can be rewritten as linear optimization problem over the copositive cone, see e.g. [8, 10, 25, 27].

Brás et al. have recently [9] presented tests for copositivity based on solving associated LCP. More precisely, the copositivity of $A \in \mathbb{R}^{n-1} \times \mathbb{R}^{n-1}$ can be revealed by considering the solutions of (LCP), with:

$$
M_{cop} = \begin{pmatrix} A & e_{n-1} \\ e_{n-1}^\top & 0 \end{pmatrix}, \quad q_{cop} = \begin{pmatrix} 0_{n-1} \\ -1 \end{pmatrix},
$$

Table 1: Columns 2-4 contain numbers of iterations needed by PC IPAs with different $\varphi$ to reach $\varepsilon = 10^{-5}$. We were solving LCPs with $10 \times 10$-matrices $M$ taken from [33], which have positive handicap.
\[ \varphi(t) = t \quad \varphi(t) = \sqrt{t} \quad \varphi(t) = t - \sqrt{t} \]

<table>
<thead>
<tr>
<th></th>
<th>( \varphi(t) = t )</th>
<th>( \varphi(t) = \sqrt{t} )</th>
<th>( \varphi(t) = t - \sqrt{t} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS_20_01</td>
<td>7</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>MS_20_02</td>
<td>6</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>MS_20_03</td>
<td>7</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>MS_20_04</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>MS_20_05</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2: In columns 2-4 we report number of iterations needed by PC IPAs with different \( \varphi \) to reach proximity \( \varepsilon = 10^{-5} \). We were solving LCPs with \( 20 \times 20 \)-matrices \( M \) taken from [33], which have positive handicap.

<table>
<thead>
<tr>
<th>( \varphi(t) = t - \sqrt{t} )</th>
<th>( n = 10 )</th>
<th>( n = 20 )</th>
<th>( n = 100 )</th>
<th>( n = 200 )</th>
<th>( n = 500 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations</td>
<td>53</td>
<td>91</td>
<td>97</td>
<td>112</td>
<td>153</td>
</tr>
<tr>
<td>CPU time (s)</td>
<td>&lt; 0.001</td>
<td>0.05</td>
<td>0.358</td>
<td>3.171</td>
<td>74.215</td>
</tr>
<tr>
<td>( \kappa' )</td>
<td>4095.75</td>
<td>( 2^{32} - 0.25 )</td>
<td>( 2^{192} - 0.25 )</td>
<td>( 2^{392} - 0.25 )</td>
<td>( 2^{992} - 0.25 )</td>
</tr>
</tbody>
</table>

Table 3: Numerical results for (LCP) with matrix \( M \) as in (2). Results were obtained by our PC IPA based on \( \varphi(t) = t - \sqrt{t} \).

where \( e_{n-1} \) and \( 0_{n-1} \) denotes the \( (n-1) \)-dimensional vectors of all ones and zeros, respectively. They have proved (see [9, Corollary 4]) that:

C1. if \( \exists (x^*, s^*) \in F^* \) with \( x^*_n > 0 \), then \( A \) is not copositive;

C2. if \( \forall (x^*, s^*) \in F^* \) we have \( x^*_n = 0 \), then \( A \) is on the boundary (it is copositive, but not strictly copositive), and

C3. if \( F^* = \emptyset \) then \( A \) is strictly copositive.

In practice we solve (LCP) with data \( M_{\text{cop}} \) and \( q_{\text{cop}} \) using the algorithm presented in Figure 1 based on the AET \( \varphi(t) = t - \sqrt{t} \). Practical implementation uses slightly different set of parameters than the theoretical one.
Namely, the parameter $\theta$ is replaced by $\sigma_1, \sigma_2 \in (0,1)$. The role of $\sigma_1$ is to decrease $\mu$ to $\sigma_1 \mu$, while $\sigma_2$ controls the step length in each iteration. For each matrix instance that we want to classify as strictly copositive, on the boundary or not-copositive, we run our algorithm for all combinations of $\sigma_1 \in \{0.05, 0.1, 0.15, \ldots, 0.5\}$ and $\sigma_2 \in \{0.025, 0.05, 0.075, \ldots, 0.2\}$. In practical implementations of IPAs it is important to control the relative infeasibility and the relative complementarity gap in order to converge towards the solution of the LCP. We achieved this by considering the following constraints:

$$\frac{\|Mx - s + q\|}{1 + \|q\|} \leq \bar{\varepsilon}_1 \quad \text{and} \quad \frac{x^Ts}{1 + x_0^Ts_0} \leq \bar{\varepsilon},$$

where $\bar{\varepsilon}_1, \bar{\varepsilon}_2 > 0$ and $\bar{\varepsilon} = \min\{\bar{\varepsilon}_1, \bar{\varepsilon}_2\}$. In the test computations we set $\varepsilon = 10^{-5}$. We can define the $\varepsilon > 0$ used in the theoretical analysis of our algorithm as follows, $0 < \varepsilon \leq (1 + x_0^Ts_0) \bar{\varepsilon}$.

During the computational tests we found that the criteria C1-C3 are numerically very sensitive. Thus, we introduced an additional, safeguard stopping rule, namely we set the maximum number of iterations to 3000. Based on the results of all runs we predict the (non)copositivity of given matrix upon the following simple rules:

R1. if at least one run of the algorithm returns an $\varepsilon$-optimal solution for (LCP) with data $M_{\text{cop}}$ and $q_{\text{cop}}$ and $x_n > \bar{\varepsilon}$, then $A$ is not copositive;

R2. if for all runs that return an $\varepsilon$-optimal solution for (LCP) with data $M_{\text{cop}}$ and $q_{\text{cop}}$ we have $x_n \leq \bar{\varepsilon}$, then $A$ is on the boundary of the copositive cone;

R3. if the algorithm never returns an $\varepsilon$-optimal solution for (LCP) with data $M_{\text{cop}}$ and $q_{\text{cop}}$, then we conclude that $\mathcal{F}^* = \emptyset$, hence $A$ is strictly copositive.

We evaluated this approach on matrices $M_1 - M_7$ from [9], for which we know the real status of copositivity. Motivated by the discussion with the authors of [9] and with their help we constructed a set of matrices related to maximum clique problem, for which we knew by construction the real status of copositivity. More precisely, for each graph $G$ with known clique number (the size of maximum clique in the graph) we took its adjacency matrix $A_G$ and its clique number $\omega_G$ and computed three matrices:

$$A_{\text{not}} = (\omega_G - 1)(E - A_G) - E,$$

$$A_{\text{bound}} = \omega_G(E - A_G) - E,$$

$$A_{\text{int}} = (\omega_G + 1)(E - A_G) - E.$$
where $E$ is the square matrix of appropriate order with only ones. By construction, the first matrix is not copositive, the second is on the boundary and the third is in the interior of the copositive cone, see [9] for justification.

The criteria C1-C3 well characterize the copositivity property of the given matrix. Since by computations we can obtain only $\varepsilon$-optimal solution for the given LCPs, the mentioned criteria could not be applied directly. Therefore, taking into consideration the computational results, we proposed heuristic decision rules (see R1-R3) in order to identify the copositivity property of the given matrix. According to the decision rules R1-R3, we present our classifications of matrices regarding the copositivity property in three tables. More precisely, the results on small-size graphs in Table 4, Table 5 and Table 6 are based on rules R1, R2 and R3, respectively.

Our tables contain 6 columns. The name of the graph is given in the first column. The structure of names consists of two parts, where the second one reflects on the type of the matrix (not-COP, on-BOUND and strict-COP). The second column contains the size of the matrix; the third column shows how many runs (out of 80) of the algorithm reached the maximum number of iterations. The fourth and fifth columns explain how many times we obtained an $\varepsilon$-optimal solution with $x_n > \bar{\varepsilon}$ and $x_n \leq \bar{\varepsilon}$, respectively. The last column contains 1 if our algorithm correctly revealed the true status of the matrix.

We can see that whenever the matrix is strictly copositive, we always detect this since the algorithm does not give any $\varepsilon$-optimal solution (see R3 rule). On the other hand, if the matrix is not copositive or is on the boundary, our algorithm does not always terminate within 3,000 iterations. However, when it does, it gives almost always the correct answers. Only in 5 out the remaining 59 matrices the algorithm gives wrong answer. These are the cases having 0 in the last column of Tables 4 and 5. The heuristic decision rule R1 supported wrong classification 3 times (c-fat16-1-not-COP, c-fat18-1-not-COP and cisqrg20-not-COP), while R2 failed to suggest right decision in 2 cases (c-fat16-1-on-BOUND and cisqrg20-on-BOUND).

7 Conclusions and future research

In this paper we presented a PC IPA for $P_\ast(\kappa)$-LCPs which uses a new search direction, based on AET method with $\varphi(t) = t - \sqrt{t}$. We proved that this PC IPA retains polynomial iteration complexity in the handicap of the problem’s matrix, the size of the problem, the bitsize of the data and the deviation from the complementarity gap. This is the first PC IPA for solving $P_\ast(\kappa)$-LCPs using the AET method with $\varphi(t) = t - \sqrt{t}$.
<table>
<thead>
<tr>
<th>instance</th>
<th>order</th>
<th>Max_Its</th>
<th>R1</th>
<th>R2</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1-not-COP</td>
<td>4</td>
<td>5</td>
<td>75</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M2-not-COP</td>
<td>4</td>
<td>2</td>
<td>78</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Hamming4-4-not-COP</td>
<td>16</td>
<td>46</td>
<td>34</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-2-4-not-COP</td>
<td>15</td>
<td>75</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-4-4-not-COP</td>
<td>15</td>
<td>75</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Johnson7-2-4-not-COP</td>
<td>21</td>
<td>79</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Keller2-not-COP</td>
<td>16</td>
<td>29</td>
<td>51</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c-fat14-1-not-COP</td>
<td>14</td>
<td>75</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c-fat16-1-not-COP</td>
<td>16</td>
<td>76</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>c-fat18-1-not-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cisqrg14-not-COP</td>
<td>14</td>
<td>57</td>
<td>14</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg16-not-COP</td>
<td>16</td>
<td>66</td>
<td>8</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg18-not-COP</td>
<td>18</td>
<td>69</td>
<td>8</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg20-not-COP</td>
<td>20</td>
<td>75</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>ggen14-not-COP</td>
<td>14</td>
<td>40</td>
<td>20</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>ggen16-not-COP</td>
<td>16</td>
<td>68</td>
<td>12</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen18-not-COP</td>
<td>18</td>
<td>74</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>ggen20-not-COP</td>
<td>20</td>
<td>65</td>
<td>8</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>ggen22-not-COP</td>
<td>22</td>
<td>67</td>
<td>4</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>ggen24-not-COP</td>
<td>24</td>
<td>73</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>krcgg14-not-COP</td>
<td>14</td>
<td>61</td>
<td>19</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>krcgg16-not-COP</td>
<td>16</td>
<td>62</td>
<td>18</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>krcgg18-not-COP</td>
<td>18</td>
<td>68</td>
<td>12</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis14-not-COP</td>
<td>14</td>
<td>45</td>
<td>35</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis16-not-COP</td>
<td>16</td>
<td>63</td>
<td>9</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>sanchis18-not-COP</td>
<td>18</td>
<td>66</td>
<td>14</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis20-not-COP</td>
<td>20</td>
<td>59</td>
<td>21</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis22-not-COP</td>
<td>22</td>
<td>67</td>
<td>13</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis24-not-COP</td>
<td>24</td>
<td>61</td>
<td>19</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: This table contains results for the not copositive matrices. The column R1 shows how often out of 80 runs the algorithm returned an \(\varepsilon\)-optimal solution with \(x_n > \bar{\varepsilon}\), while the column R2 shows how often the algorithm terminated with \(\varepsilon\)-optimal solution with \(x_n \leq \bar{\varepsilon}\). Based on the Rules R1–R3, we classify the matrix as not copositive (which is correct for matrices in this table), if and only if the value in the column R1 is greater than 0. This happens in all but three cases.
Table 5: This table contains results for the matrices that are on the boundary of the copositive cone. Based on the decision rules R1–R3, we classify the matrix to be on the boundary if there is a zero in the column R1 and a positive value in the column R2. This way we correctly reveal the copositivity status of all but two test matrices.

<table>
<thead>
<tr>
<th>instance</th>
<th>order</th>
<th>Max_Its</th>
<th>R1</th>
<th>R2</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5-on-BOUND</td>
<td>5</td>
<td>17</td>
<td>0</td>
<td>63</td>
<td>1</td>
</tr>
<tr>
<td>M6-on-BOUND</td>
<td>5</td>
<td>72</td>
<td>0</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>M7-on-BOUND</td>
<td>7</td>
<td>24</td>
<td>0</td>
<td>56</td>
<td>1</td>
</tr>
<tr>
<td>Hamming4-4-on-BOUND</td>
<td>16</td>
<td>76</td>
<td>0</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-2-4-on-BOUND</td>
<td>15</td>
<td>79</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-4-4-on-BOUND</td>
<td>15</td>
<td>79</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Johnson7-2-4-on-BOUND</td>
<td>21</td>
<td>79</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Keller2-on-BOUND</td>
<td>16</td>
<td>60</td>
<td>0</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>c-fat14-1-on-BOUND</td>
<td>14</td>
<td>77</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>c-fat16-1-on-BOUND</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c-fat18-1-on-BOUND</td>
<td>18</td>
<td>79</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg14-on-BOUND</td>
<td>14</td>
<td>68</td>
<td>0</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg16-on-BOUND</td>
<td>16</td>
<td>71</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg18-on-BOUND</td>
<td>18</td>
<td>79</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg20-on-BOUND</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ggen14-on-BOUND</td>
<td>14</td>
<td>52</td>
<td>0</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>ggen16-on-BOUND</td>
<td>16</td>
<td>71</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>ggen18-on-BOUND</td>
<td>18</td>
<td>74</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>ggen20-on-BOUND</td>
<td>20</td>
<td>71</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>ggen22-on-BOUND</td>
<td>22</td>
<td>74</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>ggen24-on-BOUND</td>
<td>24</td>
<td>75</td>
<td>0</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>krcgg14-on-BOUND</td>
<td>14</td>
<td>61</td>
<td>0</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>krcgg16-on-BOUND</td>
<td>16</td>
<td>68</td>
<td>0</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>krcgg18-on-BOUND</td>
<td>18</td>
<td>70</td>
<td>0</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>sanchis14-on-BOUND</td>
<td>14</td>
<td>50</td>
<td>0</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>sanchis16-on-BOUND</td>
<td>16</td>
<td>72</td>
<td>0</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>sanchis18-on-BOUND</td>
<td>18</td>
<td>65</td>
<td>0</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>sanchis20-on-BOUND</td>
<td>20</td>
<td>64</td>
<td>0</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>sanchis22-on-BOUND</td>
<td>22</td>
<td>72</td>
<td>0</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>sanchis24-on-BOUND</td>
<td>24</td>
<td>62</td>
<td>0</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 6: Results for the strictly copositive matrices show that our algorithm never terminated with an \( \varepsilon \)-optimal solution (the values in the columns R1 and R2 are zero). Therefore, for all matrices from this table we apply the rule R3 and (correctly) conclude that the matrix is strictly copositive.

<table>
<thead>
<tr>
<th>instance</th>
<th>order</th>
<th>Max_Its</th>
<th>R1</th>
<th>R2</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>M3-strict-COP</td>
<td>4</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M4-strict-COP</td>
<td>3</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Hamming4-4-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-2-4-strict-COP</td>
<td>15</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Johnson6-4-4-strict-COP</td>
<td>15</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Johnson7-2-4-strict-COP</td>
<td>21</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Keller2-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c-fat14-1-strict-COP</td>
<td>14</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c-fat16-1-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c-fat18-1-strict-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg14-strict-COP</td>
<td>14</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg16-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg18-strict-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>cisqrg20-strict-COP</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen14-strict-COP</td>
<td>14</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen16-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen18-strict-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen20-strict-COP</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen22-strict-COP</td>
<td>22</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ggen24-strict-COP</td>
<td>24</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>krcgg14-strict-COP</td>
<td>14</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>krcgg16-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>krcgg18-strict-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis14-strict-COP</td>
<td>14</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis16-strict-COP</td>
<td>16</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis18-strict-COP</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis20-strict-COP</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis22-strict-COP</td>
<td>22</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sanchis24-strict-COP</td>
<td>24</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Beside this, we gave a new unification of the Newton-systems and scaled systems in case of PC IPAs for sufficient LCPs. The constructed general framework showed that in order to introduce PC IPAs we had to decompose the right-hand side of the Newton-system into two parts: the one which depends and the other which does not depend on the positive parameter $\mu$. This decomposition is not trivial, and in general when we use AET we have no guaranty that such decomposition exists, i.e., with general AET we have no guaranty that we can develop a PC IPA for LCP.

We implemented the proposed PC IPA in the C++ programming language and tested on three families of LCPs. To the best of our knowledge, we presented the first numerical results for LCPs with $P_\ast(\kappa)$-matrices that have positive $\kappa$. Numerical tests show that our PC IPA works very well. Furthermore, on the test problems generated by Illés and Morapitiye [33] PC IPAs based on AET with $\varphi(t) = t$ and $\varphi(t) = \sqrt{t}$ showed equally good computational performance (see Tables 1 and 2). Important difference is on LCPs with matrices (2), where our PC IPA solves the problem, while PC IPAs based on other variants of AET do not (see Table 3).

We also apply our PC IPA on (LCP) with data $M_{\text{cop}}$ and $q_{\text{cop}}$ that are related to the copositivity tests of matrices from [9]. It is important to emphasize that these matrices may not be sufficient, therefore we can use our PC IPA only as a heuristics for this class of LCPs. Although the computations related to our heuristic decision rules (see R1-R3) are numerically very sensitive, our PC IPA gives results that yield after appropriate rounding the exact answer for the copositivity test for 83 out of 88 test matrices (94.32%). This confirms that our PC IPA turns the LCP into a promising tool to detect matrix copositivity.

Several questions remain open for future work. For instance, Illés et al. [36, 37] generalized some IPAs with AET using $\varphi(t) = t$ to handle general LCPs (i.e. solve these LCPs in EP-sense). Naturally arising question is which IPAs (or PC IPAs) using some nontrivial AET could be generalized in EP-sense to solve general LCPs.

Another important question would be how to find a general class of functions $\varphi$ for AET which gives IPAs with the best known complexity results for solving sufficient LCPs. Haddou et al. [32] proposed a family of smooth concave functions which yields IPAs with the best known iteration complexity bound for monotone LCPs (i.e. their matrix is either skew-symmetric or positive semidefinite). Note that our function $\varphi(t) = t - \sqrt{t}$ for the AET does not belong to the family defined by Haddou et al. Based on our computational tests with different AETs for monotone LCPs (for example for linear programming problems, see Darvay et al. [22]) and some small-sized sufficient LCPs.
(see Tables 1 and 2) our observation is that the effect of more advanced AETs on computational performance is not so significant. However, from Table 3 it is obvious that for hard (i.e. with exponentially large $\kappa$ parameter), sufficient LCPs there are major difference on computational performance depending on the chosen AET. Thus, it would be interesting to extend and computationally test the IPAs introduced by Haddou et al. to hard, sufficient LCPs as a step towards understanding the real effect of AETs on practical efficiency of IPAs.

Another interesting task is to generalize the presented PC IPA to more general problems, such as the general LCPs in the EP-sense and to sufficient LCPs over Cartesian product of symmetric cones.

Illés and Morapitiye [33] started the development of a test set problems for sufficient LCPs. It would be essential to expand the current test set of sufficient LCPs towards problems with larger size and with matrices having large (probably exponential) $\kappa$ parameter. This would create a natural requirement of computationally testing any newly introduced IPA for sufficient LCPs. A well-developed test set of sufficient LCPs needs to contain several items similar to that one defined by Csizmadia (see matrix in (2)). In the study of Klerk and E.-Nagy [24] on the matrix of Csizmadia, they pointed out that the complexity bounds of known IPAs for sufficient LCPs are not polynomial in the input size of the LCP, due to the fact that the matrix of the problem might have exponentially large $\kappa$ parameter. However, Table 3 shows that the variant of the PC IPA using $\phi(t) = t - \sqrt{t}$ AET does not perform exponentially many iterations. This opens some questions, among others, whether it is possible to develop thorough complexity analysis of our PC IPA which leads to complexity bound containing $\log \kappa$ instead of $\kappa$ parameter. If the conjecture of Klerk and E.-Nagy (see Subsection 2.1) holds, then the complexity bound containing $\log \kappa$ would yield to polynomial complexity bound of our PC IPA in the size and bit length of the problem. Namely, proving or disproving the mentioned conjecture would have major effect on the theory of sufficient LCPs. In our opinion, computing a sharp upper bound for the $\kappa$ parameter of Csizmadia’s matrix is the simplest step to understand the depth of the conjecture.

It could happen that our PC IPA might be applied to some structured LCPs having no sufficient matrices. It should be mentioned that in such cases, the PC IPA without convergence analysis is only a smart heuristics. Extending the computational study of our PC IPA to some wider classes of LCPs (coming from bimatrix games, testing copositivity of matrices, Arrow-Debreu type market exchange models etc.) would be also challenging future work. Based on our preliminary computational study (see Tables 4 – 6), fine tuning of our PC IPA to reveal matrix copositivity with higher accuracy.
is another interesting task to be addressed. Such tool might be useful in combinatorial optimization, too.

Acknowledgements

This research has been partially supported by the Hungarian Research Fund, OTKA (grant no. NKFIH 125700).

The research of T. Illés and P.R. Rigó has been partially supported by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM).

The research of Zs. Darvay and P.R. Rigó was supported by a grant of Romanian Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2016-0190, within PNCDI III.

The research of J. Povh was supported by Slovenian Research Agency grants J1-8155, N1-0057 and N1-0071.

The authors also want to thank to prof. Gabrielle Eichfelder for sharing the matrices for copositive tests in Subsection 6.2.

References


[5] S. Asadi and H. Mansouri. A path-following algorithm for $P_*(\kappa)$-horizontal linear complementarity problem based on Darvay’s direc-
PC IPA for sufficient LCPs based on a new search direction


Selected Operations Research Reports

2018-02 Zsolt Darvay, Tibor Illés, Csilla Majoros: Interior-point algorithm for sufficient LCPs based on the technique of algebraically equivalent transformation


2016-03 Petra-Renáta Takács, Zsolt Darvay: Infeasible interior-point method for symmetric optimization using a positive-asymptotic barrier

2016-01 Zsolt Darvay, Petra-Renáta Takács: New interior-point algorithm for symmetric optimization based on a positive-asymptotic barrier function

2015-02 Tibor Illés, Rihárd Molnár-Szipai: Strongly polynomial primal monotonic build-up simplex algorithm for maximal flow problems

2014-04 Tibor Illés: Lineáris optimalizálás elmélete és belsopontos algoritmusai

2014-01 Tibor Illés, Adrienn Nagy: Finiteness of the quadratic primal simplex method when s-monotone index selection rules are applied

2013-03 Tibor Illés: Lineáris optimalizálás elmélete és pivot algoritmusai

2013-02 Tibor Illés, Gábor Lovics: Approximation of the Whole Pareto-Optimal Set for the Vector Optimization Problem

2013-01 Zsolt Csizmadia, Tibor Illés, Adrienn Nagy: The s-Monotone Index Selection Rule for Criss-Cross Algorithms of Linear Complementarity Problems

2012-02 Tibor Illés, Adrienn Nagy: Computational aspects of simplex and MBU-simplex algorithms using different anti-cycling pivot rules

2012-01 Tibor Illés, Rihárd Molnár-Szipai: On Strongly Polynomial Variants of the MBU-Simplex Algorithm for a Maximum Flow Problem with Nonzero Lower Bounds