Corvinus Economics Working Papers

01/2024

Marianna E. Nagy
and Anita Varga

A long-step interior point framework and a related function class for linear optimization

http://unipub.lib.uni-corvinus.hu/10192

Corvinus University of Budapest

A LONG-STEP INTERIOR POINT FRAMEWORK AND A RELATED FUNCTION CLASS FOR LINEAR OPTIMIZATION

MARIANNA E.-NAGY* AND ANITA VARGA†

Abstract. In this paper, we introduce a general long-step algorithmic framework for solving linear programming problems based on the algebraically equivalent transformation technique proposed by Darvay. The main characteristics of the proposed general interior point algorithm are based on the long-step method of Ai and Zhang, which was one of the first long-step algorithms with the best known theoretical complexity. We investigate a set of sufficient conditions on the transformation function applied in the algebraically equivalent transformation technique, under which the convergence and best known iteration complexity of the examined general algorithmic framework can be proved. As a result, we propose the first function class in connection with the algebraically equivalent transformation technique that can be used to introduce new long-step interior point methods. Furthermore, we propose construction rules that can be used to determine new elements of this function class. Additionally, we generalize Darvay's algebraically equivalent transformation technique to piecewise continuously differentiable transformation functions. We implemented the general algorithmic framework in MATLAB and tested its performance for six different transformation functions on a set of linear programming problem instances from the Netlib library.

Keywords. Linear programming, Interior point methods, Algebraically equivalent transformation technique

JEL classification number. C61

1. Introduction.

The literature on interior point algorithms (IPAs) is extensive and diverse. Several variants have already been defined for different problem classes. Classical methods have one thing in common: they follow a one-dimensional smooth curve, the so-called central path. When determining the search direction, we can follow different strategies. In this aspect, the two well-known families of algorithms are kernel function-based IPAs and self-regular function-based IPAs. In addition to these, Darvay introduced the concept of the algebraically equivalent transformation (AET) technique. This approach keeps the central path the same, but new search directions may be achieved by modifying the function used in the procedure. However, the literature currently lacks a comprehensive study of this approach. Our initial aim is to examine this technique, but in this paper, we consider an even more general case: we investigate the Newton system (determining the search direction) with a general right-hand side. We analyze this technique for LPs to keep our focus on investigating different search directions and not be distracted by the complexity of the problem to be solved. Meanwhile, we examine a long-step algorithm since it works more efficiently in practice than short-step algorithms, despite the better theoretical complexity of short-step variants. One of the first long-step IPAs with the best known iteration complexity was proposed by Ai and Zhang [2] in 2005 for monotone linear complementarity problems (LCPs). Based on their method and ideas, numerous authors introduced long-step IPAs with the best known theoretical complexity for different problem classes, e.g., for linear optimization [16, 32, 39], semidefinite optimization [21, 31, 34], horizontal linear complementarity problems (HLCPs) [36], and also for symmetric cone Cartesian $\mathcal{P}_*(\kappa)$-HLCPs [4, 5].

The main idea of the algebraically equivalent transformation (AET) technique of Darvay [7, 8] is to apply a continuously differentiable and invertible function to the centering equation of the central path problem to make it possible to find new search directions for interior point algorithms. For different transformation functions, different algorithms can be introduced.

In his first articles, Darvay [7, 8] applied the function $\varphi(t) = \sqrt{t}$ and introduced a short-step IPA for linear programming (LP) problems. Since then, his method has become widely known and has been generalized using different functions for several problem classes.

The function $\varphi(t) = \sqrt{t}$ is the most frequently applied in articles investigating the AET technique. For example, in 2018, Darvay and Rigó [16] proposed a long-step IPA for linear optimization with the best known theoretical complexity. Furthermore, in 2021, Illés et al. [24] introduced a predictor-corrector IPA for $\mathcal{P}_*(\kappa)$-LCPs. Using this function, Asadi and Mansouri [3] investigated a short-step IPA for $\mathcal{P}_*(\kappa)$-HLCPs.

The function $\varphi(t) = t - \sqrt{t}$ is also frequently studied in the literature and has been proposed by Darvay et al. [13] in 2016. In the last few years, it has been applied in several papers by Darvay and his coauthors.

*Corvinus Centre for Operations Research, Corvinus University of Budapest, Fővám tér 8., H-1093 Budapest, Hungary (marianna.eisenberg-nagy@uni-corvinus.hu)

†Corvinus Centre for Operations Research, Corvinus University of Budapest, Fővám tér 8., H-1093 Budapest, Hungary (anita.varga@uni-corvinus.hu)

They proposed a corrector predictor IPA for linear optimization [9] and another corrector predictor IPA for sufficient LCPs [11]. They also presented a short-step IPA for sufficient LCPs [10]. In addition to these functions, $\varphi(t) = \frac{\sqrt{t}}{2(1+\sqrt{t})}$ has been proposed by Kheirfam and Haghighi [27], to solve $\mathcal{P}^*(\kappa)$-LCPs.
In 2018 Darvay and Takács [17] introduced a new type of AET technique for linear programming problems based on a different rearrangement of the centering equation. Using this new type of AET technique, recently, Darvay et al. [12] proposed a predictor-corrector IPA to solve sufficient LCPs.
Other methods have also been investigated in the literature to find new search directions for IPAs. For this purpose, Peng et al. [33] considered barrier functions defined by self-regular kernel functions. With this approach, they could reduce the gap between the complexity bounds of short-step and long-step IPAs. The class of eligible kernel functions was proposed by Bai et al. [6]. These results were extended to $\mathcal{P}_*(\kappa)$-LCPs by Lesaja and Roos [30]. The class of positive-asymptotic kernel functions was introduced by Darvay and Takács in [14]. The relation of this methodology to the AET technique has been discussed in detail in the Ph.D. dissertation of Rigó [37, Section 2.5].
This paper proposes a new class of long-step IPAs for linear optimization by introducing a general Ai-Zhang-type long-step algorithmic framework based on the AET technique. The focus of our investigation is to determine a class of functions with which applying the AET technique, the resulting long-step IPA is convergent and has the best-known iteration complexity. As it turns out, the function derived from the scaled version of the transformed Newton-system plays a crucial role in our analysis, instead of the transformation function φ itself.
In the literature, two similar attempts have been made regarding the AET technique. Haddou et al. [23] investigated a general short-step IPA for monotone LCPs, and in the meantime, Illés et al. [25] analyzed a short-step IPA for $\mathcal{P}_*(\kappa)$-LCPs. The latter was recently generalized by Darvar et al.[15] for the Cartesian product of symmetric cones. The main difference between the results of [25, 15], and this paper lies in the different choices of algorithm type. Furthermore, they fix the values of the neighborhood and update parameters in the analysis. Considering the analysis of Haddou et al. [23], there are several important differences between their and our approaches. As has already been pointed out, their general algorithm is a weighted path-following method; therefore, they consider a different formulation of the central path problem, then apply a Darvay-type transformation without reorganizing the centering equation (similarly to [7]). In our analysis, the function $p(t)$ derived from the scaled system plays a key role. However, if we used the AET method described in [23], this function could not be properly defined, since it would change in each iteration. To the best of our knowledge, ours is the first such result for long-step algorithms.
Even though in the analysis we use the properties of the function $p(t)$ derived from the scaled system, to complete our discussion, we formulate a set of necessary conditions on $\varphi(t)$ as well. We also give some construction rules that can be used to construct new transformation functions that belong to our class.
Furthermore, we generalize the AET technique to the case of piecewise continuously differentiable transformation functions. We also point out during the analysis that the continuity, monotonicity, and convexity assumptions often made in papers related to the AET technique (and even for kernel or self-regular-based IPAs) can be relaxed, and the resulting functions can work well not just in theory, but in practice as well.

The paper is organized as follows. In Section 2, we recall the theoretical results used in our investigation. In Section 3, we define a new wide neighborhood, introduce a general large-update interior point algorithmic framework, and examine its correctness. In the last subsection, we prove that the complexity of the proposed methods is $O(\sqrt{n}L)$. Section 4 discusses some specific functions and related parameters. Section 5 deals with the sufficient conditions on the function $\varphi(t)$ and then considers the case of piecewise continuously differentiable transformation functions. We collect construction rules for the proposed function class in Section 5.2. In Section 6, we present our numerical results. Section 7 summarizes our conclusions.

1.1. Notations. Throughout this paper, the following notations are used: scalars and indices are denoted by lowercase Latin letters, while uppercase Latin letters denote matrices. Sets are denoted by capital calligraphic letters. $\mathbb{R}^n_+$ is the set of n-dimensional vectors with strictly positive coordinates, and $\mathbb{R}^n_\oplus$ denotes the set of n-dimensional vectors with nonnegative coordinates. We use componentwise operations on vectors. Namely, let $f: \mathbb{R} \to \mathbb{R}$ be a given function and $\mathbf{x} \in \mathbb{R}^n$ be a given vector. Then $f(\mathbf{x}) = [f(x_1), \dots, f(x_n)]^T$. Specifically, let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ be two vectors and $\alpha \in \mathbb{R}$. Then $\mathbf{a}^\alpha = [a_1^\alpha, a_2^\alpha, \dots, a_n^\alpha]^T$, $\mathbf{ab}$ is the Hadamard product of $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{a}/\mathbf{b} = [a_1/b_1, a_2/b_2 \dots, a_n/b_n]^T$. The vector $\mathbf{a}^+$ denotes the positive part of $\mathbf{a}$, i.e., $\mathbf{a}^+ = \max\{\mathbf{a}, \mathbf{0}\} \in \mathbb{R}^n$, and similarly, $\mathbf{a}^-$ is the negative part of $\mathbf{a}$, i.e., $\mathbf{a}^- = \min\{\mathbf{a}, \mathbf{0}\} \in \mathbb{R}^n$. The

Euclidean norm of vector $\mathbf{a}$ is denoted by $\|\mathbf{a}\|$, while $\|\mathbf{a}\|_1 = \sum_{i=1}^n |a_i|$ is the L^1 (Manhattan) norm of $\mathbf{a}$. Furthermore, $\|\mathbf{a}\|_\infty = \max_{i=1}^n |a_i|$ is the infinity norm of $\mathbf{a}$. The diagonal matrix with the elements of vector $\mathbf{a}$ in its diagonal is $\operatorname{diag}(\mathbf{a})$. We use the vector $\mathbf{e}$ for the vector of all ones. Finally, the set of all indices is $\mathcal{I} = \{1, \ldots, n\}$.

2. The theoretical basis of the general algorithm. Let us consider the primal-dual LP problem pair in standard form:

$$\left.\begin{aligned} \min\ \mathbf{c}^T\mathbf{x} \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned}\right\} \qquad\qquad \left.\begin{aligned} \max\ \mathbf{b}^T\mathbf{y} \\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c} \\ \mathbf{s} \geq \mathbf{0} \end{aligned}\right\} \tag{2.1}$$

where $A \in \mathbb{R}^{m\times n}$ has full row rank, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$ are given vectors.
The optimality criteria of (2.1) can be formulated in the following way:

$$\left.\begin{aligned} A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad \mathbf{s} \geq \mathbf{0} \\ \mathbf{x}\mathbf{s} = \mathbf{0}. \qquad\quad \end{aligned}\right\} \tag{2.2}$$

When proposing IPAs, instead of the last constraint of (2.2) (the complementarity condition), we consider a perturbed version

$$\left.\begin{aligned} A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad \mathbf{s} \geq \mathbf{0} \\ \mathbf{x}\mathbf{s} = \nu\mathbf{e}, \qquad\quad \end{aligned}\right\} \tag{2.3}$$

where ν is a given positive parameter. This system is the central path problem that belongs to the primal-dual LP pair (2.1).
Let $\mathcal{F} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) :\ A\mathbf{x} = \mathbf{b},\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c},\ \mathbf{x} \geq \mathbf{0},\ \mathbf{s} \geq \mathbf{0}\}$ denote the set of primal-dual feasible solutions and $\mathcal{F}_+ = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F} :\ \mathbf{x} > \mathbf{0},\ \mathbf{s} > \mathbf{0}\}$ the set of strictly feasible solutions. It is well-known that if $\mathcal{F}_+$ is not empty, then for each $\nu > 0$ system (2.3) has a unique solution which is called the ν-center [38]. As ν tends to 0, the solutions of this system converge to an optimal solution of the LP problem.
The algebraically equivalent transformation technique has been proposed by Darvay [8] to find new search directions in interior point algorithms. His main idea was to transform the central path problem (2.3) to the following equivalent form:

$$\left.\begin{aligned} A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \\ A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \quad \mathbf{s} \geq \mathbf{0} \\ \varphi\left(\frac{\mathbf{x}\mathbf{s}}{\nu}\right) = \varphi\left(\mathbf{e}\right), \qquad \end{aligned}\right\} \tag{2.4}$$

where $\varphi : (\xi^2, \infty) \to \mathbb{R}$ is a continuously differentiable function with $\varphi'(u) > 0$ for all $u \in (\xi^2, \infty)$, $\xi \in [0, 1)$.
After applying Newton's method to system (2.4), we get the following system for the search directions:

$$\left.\begin{aligned} A\Delta\mathbf{x} &= \mathbf{0} \\ A^T\Delta\mathbf{y} + \Delta\mathbf{s} &= \mathbf{0} \\ \mathbf{s}\Delta\mathbf{x} + \mathbf{x}\Delta\mathbf{s} &= \nu\frac{\varphi(\mathbf{e}) - \varphi\left(\frac{\mathbf{x}\mathbf{s}}{\nu}\right)}{\varphi'\left(\frac{\mathbf{x}\mathbf{s}}{\nu}\right)} =: \mathbf{a}_\varphi. \end{aligned}\right\} \tag{2.5}$$

As can be seen from the previous formula, the right-hand side of the Newton-system depends on the chosen transformation function; therefore, we can determine different search directions for different functions φ.
To facilitate the analysis of interior point algorithms, we usually consider a scaled version of (2.5). Let

$$\mathbf{v} = \sqrt{\frac{\mathbf{x}\mathbf{s}}{\nu}}, \quad \mathbf{d_x} = \frac{\mathbf{v}\Delta\mathbf{x}}{\mathbf{x}}, \quad \mathbf{d_s} = \frac{\mathbf{v}\Delta\mathbf{s}}{\mathbf{s}}, \text{ and } \bar{A} = A \operatorname{diag}\left(\frac{\mathbf{v}}{\mathbf{s}}\right).$$

With these notations, the scaled Newton-system can be written as:

$$\left.\begin{aligned} \bar{A}\mathbf{d_x} &= \mathbf{0} \\ \bar{A}^T\Delta\mathbf{y} + \mathbf{d_s} &= \mathbf{0} \\ \mathbf{d_x} + \mathbf{d_s} &= \frac{\varphi(\mathbf{e}) - \varphi(\mathbf{v}^2)}{\mathbf{v}\varphi'(\mathbf{v}^2)} =: \mathbf{p}_\varphi. \end{aligned}\right\} \tag{2.6}$$

We need to ensure that $\mathbf{p}_\varphi$ is well-defined to get a correct algorithm. Therefore, we assume that $v_i > \xi$ is satisfied for all $i \in \mathcal{I}$ for all iterates of the procedure. It means that in addition to $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+$ the condition $\mathbf{v} > \xi\mathbf{e}$ must be satisfied as well. Therefore, it is reasonable to define $\mathcal{F}_+^\varphi$:

$$\mathcal{F}_+^\varphi = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+ : \mathbf{v} > \xi\mathbf{e}\}.$$

We will see that with a suitable value for the neighborhood parameter β (see the definition later), the condition $\mathbf{v} > \xi\mathbf{e}$ is always satisfied throughout the algorithm.
Let p be the function for which $p(v_i) = (p_\varphi)_i$ holds for all $v_i \in (\xi, \infty)$, i.e.,

$$p : (\xi, \infty) \to \mathbb{R}, \quad p(t) = \frac{\varphi(1) - \varphi(t^2)}{t\varphi'(t^2)}. \tag{2.7}$$

The function p describes the coordinatewise transformation applied on the vector $\mathbf{v}$ to get the right-hand side of the scaled system (2.6).

The general algorithmic framework proposed in this paper is based on the approach of Ai and Zhang [2]. One of their important ideas was to decompose the Newton-directions into two parts and use different step-lengths with the two components.
If we apply this approach to (2.5), we get the following two systems:

$$\left.\begin{aligned} A\Delta\mathbf{x}_- &= \mathbf{0} \\ A^T\Delta\mathbf{y}_- + \Delta\mathbf{s}_- &= \mathbf{0} \\ \mathbf{s}\Delta\mathbf{x}_- + \mathbf{x}\Delta\mathbf{s}_- &= \mathbf{a}_\varphi^- \end{aligned}\right\} \qquad \left.\begin{aligned} A\Delta\mathbf{x}_+ &= \mathbf{0} \\ A^T\Delta\mathbf{y}_+ + \Delta\mathbf{s}_+ &= \mathbf{0} \\ \mathbf{s}\Delta\mathbf{x}_+ + \mathbf{x}\Delta\mathbf{s}_+ &= \mathbf{a}_\varphi^+. \end{aligned}\right\} \tag{2.8}$$

We would like to point out that $\Delta\mathbf{x}_+$ does not denote the positive part of the vector $\Delta\mathbf{x}$ (the sign + is in the bottom right corner instead of the upper right), but it is the solution of the system with $\mathbf{a}_\varphi^+$ on its right-hand side. The notation is similar for the other solution vectors.
If α_1 and α_2 are the chosen step-lengths, then the new point $(\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha))$ is calculated as $\mathbf{x}(\alpha) = \mathbf{x} + \alpha_1\Delta\mathbf{x}_- + \alpha_2\Delta\mathbf{x}_+$, $\mathbf{y}(\alpha) = \mathbf{y} + \alpha_1\Delta\mathbf{y}_- + \alpha_2\Delta\mathbf{y}_+$ and $\mathbf{s}(\alpha) = \mathbf{s} + \alpha_1\Delta\mathbf{s}_- + \alpha_2\Delta\mathbf{s}_+$.
We can formulate the scaled version of (2.8) by introducing the following notations:

$$\mathbf{dx}_+ = \frac{\mathbf{v}\Delta\mathbf{x}_+}{\mathbf{x}}, \ \mathbf{ds}_+ = \frac{\mathbf{v}\Delta\mathbf{s}_+}{\mathbf{s}}, \ \mathbf{dx}_- = \frac{\mathbf{v}\Delta\mathbf{x}_-}{\mathbf{x}}, \ \mathbf{ds}_- = \frac{\mathbf{v}\Delta\mathbf{s}_-}{\mathbf{s}}.$$

This way, we can define the scaled systems as follows:

$$\left.\begin{aligned} \bar{A}\mathbf{dx}_- &= \mathbf{0} \\ \bar{A}^T\Delta\mathbf{y}_- + \mathbf{ds}_- &= \mathbf{0} \\ \mathbf{dx}_- + \mathbf{ds}_- &= \mathbf{p}_\varphi^- \end{aligned}\right\} \qquad \left.\begin{aligned} \bar{A}\mathbf{dx}_+ &= \mathbf{0} \\ \bar{A}^T\Delta\mathbf{y}_+ + \mathbf{ds}_+ &= \mathbf{0} \\ \mathbf{dx}_+ + \mathbf{ds}_+ &= \mathbf{p}_\varphi^+. \end{aligned}\right\} \tag{2.9}$$

In our analysis, we fix the value of ν as $\tau\mu$, where $\mu = (\mathbf{x}^T\mathbf{s})/n$ and $0 < \tau < 1$ is a given parameter, i.e., from a point $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+^\varphi$ we take a step towards the $\tau\mu$-center (which is the solution of the central path problem (2.3) for $\nu = \tau\mu$).
Let $\mathcal{I}_+ = \{i \in \mathcal{I} : \tau\mu - x_is_i > 0\} = \{i \in \mathcal{I} : v_i < 1\}$, and $\mathcal{I}_- = \mathcal{I} \setminus \mathcal{I}_+$. If a point $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ is on the central path, then $x_is_i = \tau\mu$ holds for all indices. Therefore, these two index sets can be considered as a partition of $\mathcal{I}$, based on whether the centering equation is under- or over-fulfilled.
Traditionally, in papers proposing Ai-Zhang-type methods, the index sets $\mathcal{I}_+$ and $\mathcal{I}_-$ are defined in a slightly different way, namely, the indices for which $v_i = 1$ are included in the set $\mathcal{I}_+$, instead of $\mathcal{I}_-$. The reason behind this small change will be explained in the following section.

3. The general algorithmic framework and its analysis. The function p plays a central role in our paper. In this section, we give a set of conditions on p under which the convergence and the best known complexity of the general Ai-Zhang-type long-step algorithm investigated can be proved. Then in Section 5, we will transform them into conditions on the transformation function φ, but at this point of the analysis, we do not use the connection between φ and p. Therefore, from now on, we omit the subscript from the notation of the vector $\mathbf{p}_\varphi$ and use $\mathbf{p}$ instead.

We defined $p(t)$ in (2.7) over the interval (ξ, ∞). However, by applying the following useful upper bound on the coordinates of the vector $\mathbf{v}$, we can restrict our analysis to a much narrower interval. This upper bound is generally applied and plays an important role in analyzing Ai-Zhang-type methods since Ai-Zhang-type neighborhoods do not restrict the value of v_i (when i is in $\mathcal{I}_-$), unlike other types of neighborhoods.

REMARK 1. *We can give an upper bound on v_i by using*

$$\sum_{i\in\mathcal{I}} v_i^2 = \sum_{i\in\mathcal{I}} \frac{x_i s_i}{\tau\mu} = \frac{1}{\tau\mu}\mathbf{x}^T\mathbf{s} = \frac{n}{\tau}. \tag{3.1}$$

Since $v_i \geq 0$ for all indices, $v_i \leq t^* := \sqrt{\frac{n}{\tau}}$ for all $i \in \mathcal{I}$.

According to Remark 1, we have an upper bound on the value of the coordinates of $\mathbf{v}$; therefore, it is enough to define $p(t)$ and give the necessary lower and upper bounds over the interval $(\xi, t^*]$ instead of (ξ, ∞).

Throughout the paper, we assume that $p(t) : (\xi, t^*] \to \mathbb{R}$ satisfies the following properties:

(P_1) The inequality $p(t) \geq 1 - t^2$ holds over $(\xi, 1)$.
(P_2) There exists a positive constant c for which $p(t) \geq -c(t - 1/t)$ holds for all $t \in [1, t^*]$.
(P_3) There exists a positive constant r for which $p(t) \leq -r(t - 1/t)$ is satisfied for all $t \in [1, t^*]$.
(P_4) There exist two constants $\varrho \in [1, 2)$ and $\eta \in (\xi, 1)$ such that $p(t) \leq \varrho(1 - t^2)$ for all $t \in (\eta, 1)$.

Due to the Ai-Zhang-type decomposition and the special neighborhood definition, the domain of the function p is divided into two intervals with respect to the proposed properties; $(\xi, 1)$ and $[1, t^*]$. From (P_2) and (P_3) it follows that $p(1) = 0$. These two properties give lower and upper bounds for the function $p(t)$ over the interval $[1, t^*]$. The properties (P_1) and (P_4) refer to the behavior of the function $p(t)$ over the interval $(\xi, 1)$. As will turn out from our analysis, in some sense, the constraints over the interval $[1, t^*]$ play a more important role in proving the desired properties of the general IPA.

The constraint (P_4) is not directly necessary for our proof to work; however, it guarantees the existence of a suitable update and neighborhood parameters (to be discussed later in this section). Therefore, (P_4) is included here mainly for the sake of completeness.

An important equivalence comes from (P_1) and (P_3), that is,

$$p(t) > 0 \quad \text{if and only if} \quad t \in (\xi, 1), \tag{3.2}$$

in other words

$$p_i > 0 \quad \text{if and only if} \quad i \in \mathcal{I}_+.$$

Note that we do not assume the continuity of $p(t)$; however, all functions previously considered in the literature related to the AET technique are continuous.

Furthermore, in our case, it is not required that $p(t)$ should be monotonically decreasing, but for almost all cases in the literature, this property is also satisfied. The only exception is the function $\varphi(u) = u^2 - u + \sqrt{u}$ introduced by Illés et al. in [25], however, it is still monotonically decreasing over $[1, \infty)$, which is the more relevant interval from the point of view of our analysis. This function satisfies conditions (P_1)-(P_4) with parameters $c = 1$ and $r = 1/2$.

In the analysis, we will use a neighborhood that depends only on the positive part of the vector $\mathbf{p}$:

$$\mathcal{W}(\tau, \beta) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+^\varphi : \|\mathbf{p}^+\| \leq \beta\right\},$$

where similarly to τ, $0 < \beta < 1$ is a given parameter.

In their paper, Ai and Zhang used the function $\varphi(u) = u$, and to define the wide neighborhood $\widetilde{\mathcal{W}}$, they applied the constraint $\|\mathbf{v}\mathbf{p}^+\| \leq \beta$:

$$\widetilde{\mathcal{W}}(\tau, \beta) = \left\{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+^\varphi : \|\mathbf{v}\mathbf{p}^+\| = \left\|\frac{1}{\tau\mu}\mathbf{a}^+\right\| \leq \beta\right\}.$$

As can be seen from the definition, $\widetilde{\mathcal{W}}$ can be defined not just in the case of the identity function but for any suitable transformation function φ, i.e., $\widetilde{\mathcal{W}}$ is a generalization of the original neighborhood of Ai and Zhang. The neighborhood $\mathcal{W}(\tau, \beta)$ can be considered as a slight modification of $\widetilde{\mathcal{W}}(\tau, \beta)$ (by using the norm of $\mathbf{p}^+$ instead of $\mathbf{vp}^+$, and adding the technical condition mentioned above). Since $v_i < 1$ for all $i \in \mathcal{I}_+$, $\|\mathbf{vp}^+\| \leq \|\mathbf{p}^+\|$ holds, our new wide neighborhood is a subset of the one introduced by Ai and Zhang, i.e., $\mathcal{W}(\tau, \beta) \subseteq \widetilde{\mathcal{W}}(\tau, \beta)$. Therefore, due to the slightly different neighborhood definitions, our approach gives a different algorithm in the particular case when $\varphi(u) = u$.
To show that $\mathcal{W}(\tau, \beta)$ is a wide neighborhood, we use a line of thought similar to Ai and Zhang in [2]. The wide neighborhood $\mathcal{N}_\infty^-$ was introduced by Kojima et al. [28]. It is defined as follows:

$$\mathcal{N}_\infty^-(1-\tau) = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+ : \mathbf{xs} \geq \tau\mu\mathbf{e}\} = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+ : \mathbf{v} \geq \mathbf{e}\}.$$

A point $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}_+$ is in the neighborhood $\mathcal{N}_\infty^-(1-\tau)$ if and only if $\mathcal{I}_- = \mathcal{I}$. This fact also motivates the slightly different definition we gave for $\mathcal{I}_+$ and $\mathcal{I}_-$, since we compare the new neighborhood $\mathcal{W}(\tau, \beta)$ to $\mathcal{N}_\infty^-(1-\tau)$ to prove that the new neighborhood is wide.

LEMMA 3.1. *Let $0 < \beta,\ \tau < 1$. The relationship between neighborhoods $\mathcal{W}$ and $\mathcal{N}_\infty^-$ is the following:*
1. *Assume that (P_3) holds, then $\mathcal{N}_\infty^-(1-\tau) \subseteq \mathcal{W}(\tau, \beta)$.*
2. *Suppose that (P_1) holds and let $\gamma = (1-\beta)\tau$. Then $\mathcal{W}(\tau, \beta) \subseteq \mathcal{N}_\infty^-(1-\gamma)$ is true.*

Proof. 1. If $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}_\infty^-(1-\tau)$, then $\mathbf{xs} \geq \tau\mu\mathbf{e}$, i.e. $\mathbf{v} \geq \mathbf{e}$. From the assumption, it follows that $\mathcal{I}_+ = \emptyset$, therefore $\|\mathbf{p}^+\| = 0$. Furthermore, from $\mathbf{v} \geq \mathbf{e}$, we also have $\mathbf{v} \geq \xi\mathbf{e}$, since $\xi \in [0, 1)$.
2. Let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{W}(\tau, \beta)$, and assume on the contrary that there exists an index $i \in \mathcal{I}$ such that $x_i s_i < \gamma\mu$ holds. Then, for this index, $v_i^2 = \frac{x_i s_i}{\tau\mu} < \frac{\gamma}{\tau} < 1$. Since $p(t) \geq 1 - t^2$ on the interval $(\xi, 1)$, we can write

$$p(v_i) \geq 1 - v_i^2 > 1 - \frac{\gamma}{\tau} = 1 - (1-\beta) = \beta.$$

But $p_i \leq \|\mathbf{p}^+\| \leq \beta$ holds for all index $i \in \mathcal{I}$ by the definition of $\mathcal{W}(\tau, \beta)$, which is a contradiction. Therefore, $x_i s_i \geq \gamma\mu$ is satisfied for all indices $i \in \mathcal{I}$. ∎

Throughout this paper, we fix the value of the parameter γ as $(1-\beta)\tau$, since in this way the proofs and estimations are easier to understand. However, it would be possible to carry out the analysis with a general parameter γ and tailor its value to each function $p(t)$ before performing the numerical tests. In this way, the set of suitable parameters would possibly be larger. However, the complexity of the IPA would remain the same, i.e., this could help with the numerical tests, but would not modify the theoretical results significantly. For example, in [20], for the function $p(t) = 2(t - t^2)/(2t - 1)$ (a corresponding transformation function is $\varphi(u) = u - \sqrt{u}$), we proposed an Ai-Zhang-type IPA for LP problems with $\gamma = 1/4(1 + \sqrt{1 - 2\beta})^2\tau$ which is suitable for this special function and gives slightly better bounds.
In the Remark 1, we gave an upper bound on the value of v_i. Using the neighborhood definition, a lower bound on v_i can also be given.

COROLLARY 3.1. *Assume that (P_1) is satisfied and let $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{W}(\tau, \beta)$. Then*

$$\frac{x_i s_i}{\tau\mu} = v_i^2 \geq \frac{\gamma}{\tau} = 1 - \beta \quad \forall\, i \in \mathcal{I}.$$

Before we start discussing the analysis, we give the pseudocode of the general algorithmic framework. The function $p(t)$ is part of the input; therefore, for each different function p we get a different long-step IPA.

Algorithm 3.1 Outline of the general algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, a function $p(t) : (\xi, \infty) \to \mathbb{R}$
  an update parameter $0 < \tau < 1$, a neighborhood parameter $0 < \beta < 1$,
  an accuracy parameter $\varepsilon > 0$,
  a starting point $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s_0}) \in \mathcal{W}(\tau, \beta)$ with $\mu_0 = \frac{\mathbf{x}_0^T\mathbf{s}_0}{n}$
$\mathbf{x} := \mathbf{x}_0$, $\mathbf{y} := \mathbf{y}_0$, $\mathbf{s} := \mathbf{s}_0$ and $\mu := \mu_0$
while $\mathbf{x}^T\mathbf{s} > \varepsilon$ **do**
  Determine $\Delta\mathbf{x}_+, \Delta\mathbf{s}_+, \Delta\mathbf{y}_+$ and $\Delta\mathbf{x}_-, \Delta\mathbf{s}_-, \Delta\mathbf{y}_-$ according to (2.8)
  $(\alpha_1, \alpha_2) := \operatorname{argmin}\{\mu(\alpha) : (\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) \in \mathcal{W}(\tau, \beta)\}$,
  where $\mathbf{x}(\alpha) = \mathbf{x} + \alpha_1\Delta\mathbf{x}_- + \alpha_2\Delta\mathbf{x}_+$, $\mathbf{y}(\alpha) = \mathbf{y} + \alpha_1\Delta\mathbf{y}_- + \alpha_2\Delta\mathbf{y}_+$ and $\mathbf{s}(\alpha) = \mathbf{s} + \alpha_1\Delta\mathbf{s}_- + \alpha_2\Delta\mathbf{s}_+$
  $\mathbf{x} := \mathbf{x}(\alpha)$
  $\mathbf{y} := \mathbf{y}(\alpha)$
  $\mathbf{s} := \mathbf{s}(\alpha)$
  $\mu := \frac{\mathbf{x}^T\mathbf{s}}{n}$
end while

As can be seen from the pseudocode (and the definition of $\mathcal{W}(\tau, \beta)$), the parameters β and τ play an important role in our analysis. Therefore, in addition to the properties (P_1)-(P_4) given for the function $p(t)$, we also introduce conditions for the constants β and τ:

(C_1) $0 < \beta < 2(1 - \xi^2)/3 \quad$ and $\quad 0 < \tau < 1$,
(C_2) $\sqrt{\beta\tau} < \dfrac{r}{c}(1 - \tau)$,
(C_3) $1 - \sqrt{1 - \beta} + \dfrac{1}{2(1 - \sqrt{\beta\tau})} \le \dfrac{1 - t^2}{p(t)} \quad$ for all $t \in \left[\sqrt{1 - \frac{3}{2}\beta}, 1\right)$,

where c and r are the positive numbers defined in the conditions (P_2) and (P_3).
Although (C_3) depends on the function $p(t)$, this constraint gives bounds on the value of the parameters β and τ. We included (P_4) in the list of assumptions on $p(t)$ to ensure that there exist β and τ satisfying (C_3). Indeed, from (P_4) it follows that $(1 - t^2)/p(t) \ge 1/\varrho > 1/2$ is satisfied over $(\eta, 1)$. According to (C_3), we should choose the value of β so that $\eta < \sqrt{1 - 3/2\beta}$ is satisfied, which gives an upper bound on β, namely $\beta < 2/3(1 - \eta^2)$. The left-hand side of (C_3) is strictly increasing in both β and τ, and its value can be arbitrarily close to $1/2$ if we choose suitably small values for the parameters, i.e., it can be less than $1/\varrho$. Furthermore, (C_1) and (C_2) give only upper bounds on the parameters, so these conditions allow us to choose arbitrarily small values for β and τ. Therefore, when (P_4) is satisfied, there exist suitable parameters β and τ.
The other direction is also true, i.e., if there exist β and τ that satisfy (C_3), then (P_4) is satisfied with $\varrho = 1/\left(1 - \sqrt{1 - \beta} + \frac{1}{2(1 - \sqrt{\beta\tau})}\right)$ and any $\sqrt{1 - \frac{3}{2}\beta} < \eta < 1$. Consequently, (P_4) is a necessary and sufficient condition for the existence of suitable parameter values.

3.1. Convergence analysis. From now on, we assume that $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{W}(\beta, \tau)$ and examine the new iterate. For this, let us introduce the following notations:

$$\mathbf{dx}(\alpha) = \alpha_1\mathbf{dx}_- + \alpha_2\mathbf{dx}_+, \quad \mathbf{ds}(\alpha) = \alpha_1\mathbf{ds}_- + \alpha_2\mathbf{ds}_+,$$

$$\mathbf{h}(\alpha) = \tau\mu\mathbf{v}^2 + \alpha_1\tau\mu\mathbf{v}\mathbf{p}^- + \alpha_2\tau\mu\mathbf{v}\mathbf{p}^+,$$

where $\alpha_1, \alpha_2 \in [0, 1]$ are given step-lengths, their values will be specified later. With these notations, $\mathbf{x}(\alpha)\mathbf{s}(\alpha) = (\mathbf{x} + \alpha_1\Delta\mathbf{x}_- + \alpha_2\Delta\mathbf{x}_+)(\mathbf{s} + \alpha_1\Delta\mathbf{s}_- + \alpha_2\Delta\mathbf{s}_+)$ can be written as

$$\mathbf{x}(\alpha)\mathbf{s}(\alpha) = \mathbf{h}(\alpha) + \tau\mu\mathbf{dx}(\alpha)\mathbf{ds}(\alpha).$$

The search directions are orthogonal, as usually is the case for LP problems, since

$$\mathbf{dx}(\alpha)^T\mathbf{ds}(\alpha) = \alpha_1^2\mathbf{dx}_-^T\mathbf{ds}_- + \alpha_1\alpha_2(\mathbf{dx}_-^T\mathbf{ds}_+ + \mathbf{dx}_+^T\mathbf{ds}_-) + \alpha_2^2\mathbf{dx}_+^T\mathbf{ds}_+,$$

where $\mathbf{dx}_+$ and $\mathbf{dx}_-$ are in the kernel of matrix $\bar{A}$, while $\mathbf{ds}_+$ and $\mathbf{ds}_-$ are in the rowspace of matrix $\bar{A}$ (see system (2.9)), therefore all four scalar products are 0 in the previous expression.
The next lemmas give positive lower bounds on the coordinates of $\mathbf{h}(\alpha)$.

Lemma 3.2. *Assume that (P_2) and (P_3) hold, and let $\alpha_1 \leq \frac{1}{c}$, then $h_i(\alpha) \geq \tau\mu$ for all $i \in \mathcal{I}_-$.*

Proof. In the case of $i \in \mathcal{I}_-$, $v_i \geq 1$, $p_i \leq 0$ according to (3.2) (based on (P_3)) and $h_i(\alpha) = \tau\mu v_i(v_i + \alpha_1 p_i)$. Using ($P_2$) and the assumption on α_1, we get

$$h_i(\alpha) - \tau\mu = \tau\mu(v_i^2 - 1 + \alpha_1 v_i p_i) \geq \tau\mu(v_i^2 - 1)(1 - c\alpha_1) \geq 0,$$

thus, for all $i \in \mathcal{I}_-$, $h_i(\alpha) \geq \tau\mu$ holds. ∎

Lemma 3.3. *Assume that (P_1)-(P_3) hold, and let $\alpha_1 \leq \frac{1}{c}$, then $\mathbf{h}(\alpha) \geq \gamma\mu\mathbf{e} = (1-\beta)\tau\mu\mathbf{e} > \mathbf{0}$.*

Proof. When $i \in \mathcal{I}_+$, based on Corollary 3.1, we get $h_i(\alpha) \geq \tau\mu v_i^2 = x_i s_i \geq \gamma\mu$. While in the case of $i \in \mathcal{I}_-$, from Lemma 3.2 it follows that $h_i(\alpha) \geq \tau\mu \geq \gamma\mu$. ∎

The next technical lemma will be used to show that the iterates remain in the neighborhood $\mathcal{W}(\tau, \beta)$.

Lemma 3.4. *Assume that (P_2) holds and let $\alpha_1 \leq \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 \leq 1$, then*

$$\| [\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)]^- \|_1 = \| [\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)]^+ \|_1 \leq \frac{1}{2}\beta.$$

Proof. Following the proof of Lemma 3.5 by Ai and Zhang [2], using the orthogonality of $\mathbf{dx}(\alpha)$ and $\mathbf{ds}(\alpha)$, we have

$$\begin{aligned}\|[\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)]^-\|_1 = \|[\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)]^+\|_1 &\leq \frac{1}{4}\|\mathbf{dx}(\alpha) + \mathbf{ds}(\alpha)\|^2 \\ &= \frac{1}{4}\|\alpha_1(\mathbf{dx}_- + \mathbf{ds}_-) + \alpha_2(\mathbf{dx}_+ + \mathbf{ds}_+)\|^2 = \frac{1}{4}\left(\alpha_1^2\|\mathbf{p}^-\|^2 + \alpha_2^2\|\mathbf{p}^+\|^2\right).\end{aligned}$$

By the definition of $\mathcal{W}(\tau, \beta)$, we have $\|\mathbf{p}^+\| \leq \beta$. We need an upper bound on the term $\|\mathbf{p}^-\|^2$:

$$\|\mathbf{p}^-\|^2 = \sum_{i \in \mathcal{I}_-} p_i^2 = \sum_{i \in \mathcal{I}_-} p^2(v_i) \leq c^2 \sum_{i \in \mathcal{I}_-} \left(v_i - \frac{1}{v_i}\right)^2 \leq c^2 \sum_{i \in \mathcal{I}_-} v_i^2 \leq c^2 \sum_{i \in \mathcal{I}} v_i^2 \leq c^2 \frac{n}{\tau},$$

where we used (P_2), and in the last inequality, we applied (3.1). Using this estimation and substituting the values of α_1 and α_2, we get

$$\frac{1}{4}\left(\alpha_1^2\|\mathbf{p}^-\|^2 + \alpha_2^2\|\mathbf{p}^+\|^2\right) \leq \frac{1}{4}\frac{1}{c^2}\frac{\beta\tau}{n}c^2\frac{n}{\tau} + \frac{1}{4}\beta^2 \leq \frac{1}{2}\beta,$$

which proves the statement. ∎

Let $\mu(\alpha) = \frac{\mathbf{x}(\alpha)^T\mathbf{s}(\alpha)}{n}$ be the duality gap after taking the Newton-step. The following lemmas examine the effect of an iteration on the duality gap. First, we give a lower bound on $\mu(\alpha)$.

Lemma 3.5. *If (P_2) holds, then*

$$\mu(\alpha) \geq (1 - \alpha_1 c)\mu.$$

Proof. Let us consider the definition of $\mu(\alpha)$:

$$\begin{aligned}\mu(\alpha) &= \mu + \frac{\alpha_1\tau\mu}{n}\mathbf{v}^T\mathbf{p}^- + \frac{\alpha_2\tau\mu}{n}\mathbf{v}^T\mathbf{p}^+ \geq \mu + \frac{\alpha_1\tau\mu}{n}\mathbf{v}^T\mathbf{p}^- = \mu + \frac{\alpha_1\tau\mu}{n}\sum_{i \in \mathcal{I}_-} v_i p(v_i) \\ &\geq \mu + \frac{\alpha_1\tau\mu}{n}\sum_{i \in \mathcal{I}_-}\left(-c\left(v_i^2 - 1\right)\right) \geq \mu - c\frac{\alpha_1\tau\mu}{n}\sum_{i \in \mathcal{I}} v_i^2 = \mu - c\frac{\alpha_1\tau\mu}{n}\frac{n}{\tau} = \mu(1 - \alpha_1 c).\end{aligned}$$

For the first estimation, we used $\mathbf{v}^T\mathbf{p}^+ \geq 0$, and for the second inequality, we applied (P_2). The last inequality follows from the nonnegativity of the parameter c. ∎

The following proposition plays an important role in calculating an upper bound on $\mu(\alpha)$.

Proposition 3.1. *If (P_3) holds, then $\mathbf{v}^T\mathbf{p}^- \leq rn\left(1 - \frac{1}{\tau}\right)$.*

Proof. Using (P$_3$) and the nonnegativity of $1 - v_i^2$ for $i \in \mathcal{I}_+$, we get

$$\mathbf{v}^T\mathbf{p}^- = \sum_{i\in\mathcal{I}_-} v_i p_i \leq \sum_{i\in\mathcal{I}_-} r(1-v_i^2) \leq \sum_{i\in\mathcal{I}} r(1-v_i^2) = rn\left(1-\frac{1}{\tau}\right).$$

∎

The next proposition will be used to prove that the reduction in the duality gap is positive.

Proposition 3.2. *Suppose that (P$_2$) and (P$_3$) are satisfied and let $\alpha_1 \leq \frac{1}{c}$, then*

$$\alpha_1\left[(1-\tau)r - c\sqrt{\beta\tau}\right] < 1.$$

Proof. Let us consider

$$(1-\tau)r - c\sqrt{\beta\tau} < (1-\tau)r < r \leq c \leq \frac{1}{\alpha_1},$$

which proves the statement. ∎

As usual for Ai-Zhang-type methods, from now on, we fix the value $\alpha_2 = 1$ in the analysis, i.e., we take a full Newton-step in the positive direction $(\Delta\mathbf{x}_+, \Delta\mathbf{y}_+, \Delta\mathbf{s}_+)$. Furthermore, we will fix the step-length $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$, and in this case, the assumption $\alpha_1 \leq \frac{1}{c}$ in the previous lemmas is automatically satisfied. The next part of the analysis shows that a certain ratio needs to be maintained between the two step-lengths to be able to prove the convergence of the method.
Now, we examine the decrease in the duality gap after an iteration.

Lemma 3.6. *Let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$. Suppose that (P$_2$) and (P$_3$) hold. Then*

$$\mu(\alpha) \leq \left(1 - \left[(1-\tau)r - c\sqrt{\beta\tau}\right]\alpha_1\right)\mu.$$

Furthermore, if (C$_2$) is also satisfied, then the duality gap decreases, i.e., $\mu(\alpha) < \mu$ holds.

Proof. By the definition of $\mu(\alpha)$,

$$\mu(\alpha) = \mu + \frac{\alpha_1\tau\mu}{n}\mathbf{v}^T\mathbf{p}^- + \frac{\alpha_2\tau\mu}{n}\mathbf{v}^T\mathbf{p}^+.$$

To estimate the term $\mathbf{v}^T\mathbf{p}^+$, we use the fact that $\mathbf{v}\mathbf{p}^+ \geq \mathbf{0}$ and $v_i < 1$ for all $i \in \mathcal{I}_+$, so $\mathbf{v}^T\mathbf{p}^+ = \|\mathbf{v}\mathbf{p}^+\|_1 \leq \|\mathbf{p}^+\|_1$. From the Cauchy-Schwartz inequality, it follows that $\|\mathbf{u}\|_1 \leq \sqrt{n}\|\mathbf{u}\|$. Therefore,

$$\mathbf{v}^T\mathbf{p}^+ \leq \sqrt{n}\beta,$$

since $\|\mathbf{p}^+\| \leq \beta$ by the definition of $\mathcal{W}(\tau,\beta)$.
Using the previous estimation and Proposition 3.1, we get

$$\mu(\alpha) \leq \mu + \frac{\alpha_1\tau\mu}{n}rn\frac{\tau-1}{\tau} + \frac{\alpha_2\tau\mu}{n}\sqrt{n}\beta = \mu + \alpha_1(\tau-1)r\mu + c\sqrt{\beta\tau}\alpha_1\mu = \mu\left(1 - \left[(1-\tau)r - c\sqrt{\beta\tau}\right]\alpha_1\right).$$

Here, the multiplier of μ is positive by Proposition 3.2, and it is less than one due to (C$_2$), therefore, $\mu(\alpha) < \mu$, i.e., the duality gap decreases. ∎

The next lemma also has an important role in proving that the new iterates remain in the wide neighborhood $\mathcal{W}(\tau,\beta)$.

Lemma 3.7. *Assume that the properties (P$_1$)-(P$_3$) and (C$_2$) hold, and let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$. Then*

$$\|\left[\tau\mu(\alpha)\mathbf{e} - \mathbf{h}(\alpha)\right]^+\| \leq \beta\tau\mu(\alpha)\left(1 - \alpha_2\sqrt{\frac{\gamma}{\tau}}\right).$$

Proof. According to Lemma 3.2, we have $h_i(\alpha) \geq \tau\mu$ for all $i \in \mathcal{I}_-$. From this, it follows that $\tau\mu(\alpha) - h_i(\alpha) \leq 0$ for all $i \in \mathcal{I}_-$.

In the case of $i \in \mathcal{I}_+$ by Lemma 3.6, (P_1) and Corollary 3.1, we have

$$\tau\mu(\alpha) - h_i(\alpha) = \tau\mu(\alpha) - \tau\mu v_i^2 - \alpha_2\tau\mu v_i p_i \leq \tau\mu(\alpha) - \frac{\mu(\alpha)}{\mu}\left(\tau\mu v_i^2 + \alpha_2\tau\mu v_i p_i\right)$$
$$= \tau\mu(\alpha)\left(1 - v_i^2 - \alpha_2 v_i p_i\right) \leq \tau\mu(\alpha)\left(1 - \alpha_2\sqrt{\frac{\gamma}{\tau}}\right) p_i.$$

Therefore, using this upper bound and the definition of $\mathcal{W}(\tau,\beta)$, we get

$$\|\left[\tau\mu(\alpha)\mathbf{e} - \mathbf{h}(\alpha)\right]^+\| \leq \tau\mu(\alpha)\left(1 - \alpha_2\sqrt{\frac{\gamma}{\tau}}\right)\|\mathbf{p}^+\| \leq \beta\tau\mu(\alpha)\left(1 - \alpha_2\sqrt{\frac{\gamma}{\tau}}\right).$$

□

The following lemma gives a lower bound on the expression $\mathbf{x}(\alpha)\mathbf{s}(\alpha)$.

Lemma 3.8. *Assume that (P_1)-(P_3) hold, and let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$. Then*

$$\mathbf{x}(\alpha)\mathbf{s}(\alpha) \geq \left(1 - \frac{3}{2}\beta\tau\right)\mu\mathbf{e}.$$

Proof. From Lemma 3.3 and Lemma 3.4, it follows that

$$\mathbf{x}(\alpha)\mathbf{s}(\alpha) = \mathbf{h}(\alpha) + \tau\mu\mathbf{dx}(\alpha)\mathbf{ds}(\alpha) \geq \gamma\mu\mathbf{e} - \tau\mu\|\left[\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)\right]^-\|\mathbf{e}$$
$$\geq \gamma\mu\mathbf{e} - \tau\mu\frac{1}{2}\beta\mathbf{e} = \mu\left(\gamma - \frac{\beta\tau}{2}\right)\mathbf{e} = \mu\left(1 - \frac{3}{2}\beta\tau\right)\mathbf{e}.$$

□

Corollary 3.2. *Assume that (P_1)-(P_3) hold, and let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$. If $\beta < \frac{2}{3}$, then $\mathbf{x}(\alpha)\mathbf{s}(\alpha) > \mathbf{0}$ holds.*

The next proposition is an analogue of Proposition 3.2 of Ai and Zhang [2] for LP problems. In their paper, they formulate this statement for monotone LCPs, but the proof remains the same in this special case.

Proposition 3.3 (Ai and Zhang [2, Proposition 3.2]). *Assume that $(\mathbf{x},\mathbf{y},\mathbf{s}) \in \mathcal{F}^+$ and let $(\Delta\mathbf{x},\Delta\mathbf{y},\Delta\mathbf{s})$ be the solution of the system*

$$\begin{aligned} A\Delta\mathbf{x} &= \mathbf{0} \\ A^T\Delta\mathbf{y} + \Delta\mathbf{s} &= \mathbf{0} \\ \mathbf{s}\Delta\mathbf{x} + \mathbf{x}\Delta\mathbf{s} &= \mathbf{z}. \end{aligned}$$

If $\mathbf{z} + \mathbf{xs} > 0$ and $(\mathbf{x} + t_0\Delta\mathbf{x})(\mathbf{s} + t_0\Delta\mathbf{s}) > 0$ hold for some $t_0 \in (0,1]$, then $\mathbf{x} + t\Delta\mathbf{x} > 0$ and $\mathbf{s} + t\Delta\mathbf{s} > 0$ for all $t \in (0,t_0]$.

Based on this proposition, we can show that the new iterates $\mathbf{x}(\alpha)$ and $\mathbf{s}(\alpha)$ are strictly positive, i.e., feasible.

Lemma 3.9. *Suppose that (P_1)-(P_3) hold, and let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$, furthermore $\beta < \frac{2}{3}$. Then $\mathbf{x}(\alpha) > \mathbf{0}$ and $\mathbf{s}(\alpha) > \mathbf{0}$ holds.*

Proof. Proposition 3.3 can be applied with $\mathbf{z} = \alpha_1\tau\mu\mathbf{vp}^- + \alpha_2\tau\mu\mathbf{vp}^+$ since according Lemma 3.3, $\mathbf{z} + \mathbf{xs} = \mathbf{h}(\alpha) > \mathbf{0}$ and by Corollary 3.2, $t_0 = 1$ is a proper choice. □

Using Lemma 3.8, we can examine what are the necessary conditions for maintaining the technical condition $\mathbf{v} > \xi\mathbf{e}$ after an iteration. For most functions from the literature, $\xi = 0$ holds, therefore in most cases, the following lemma does not give additional requirements on the parameters.

Lemma 3.10. *Suppose that (P_1)-(P_3), (C_1) and (C_2) hold, let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$, then $\mathbf{v}(\alpha) > \xi\mathbf{e}$.*

Proof. Since $\mu \geq \mu(\alpha)$ according to Lemma 3.6, from Lemma 3.8 we get

$$\mathbf{x}(\alpha)\mathbf{s}(\alpha) \geq \left(\gamma - \frac{\beta\tau}{2}\right)\mu\mathbf{e} \geq \left(\gamma - \frac{\beta\tau}{2}\right)\mu(\alpha)\mathbf{e}.$$

Combining it with the assumption on β, it follows that

$$\mathbf{v}(\alpha) = \sqrt{\frac{\mathbf{x}(\alpha)\mathbf{s}(\alpha)}{\tau\mu(\alpha)}} \geq \sqrt{\frac{\gamma}{\tau} - \frac{\beta}{2}}\mathbf{e} = \sqrt{1 - \frac{3}{2}\beta}\mathbf{e} > \xi\mathbf{e}. \tag{3.3}$$ $\square$

Now we have all the necessary results for proving that $\|\mathbf{p}(\alpha)^+\| \leq \beta$ holds, where $\mathbf{p}(\alpha)$ is the new right-hand side of the Newton-system after taking an $\alpha = [\alpha_1; \alpha_2]$-long step. Together with Lemma 3.10, this means that the new iterates after the Newton-step remain in the neighborhood $\mathcal{W}(\tau, \beta)$.

Lemma 3.11. *Suppose that (P_1)-(P_3) and (C_1)-(C_3) hold. Let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$, and $\alpha_2 = 1$. Then*

$$\|\mathbf{p}(\alpha)^+\| \leq \beta$$

holds.

Proof. We need to examine only the indices where $\mathbf{p}(\alpha)$ is positive, namely $\mathbf{v}(\alpha)$ is less than one. In this case

$$p_i(\alpha) = \frac{p(v_i(\alpha))}{1 - v_i^2(\alpha)}\left(1 - v_i^2(\alpha)\right) \leq \frac{1}{1 - \sqrt{1-\beta} + \frac{1}{2(1-\sqrt{\beta\tau})}}\left(1 - v_i^2(\alpha)\right),$$

by (C_3) and using the lower bound (3.3).
Therefore,

$$\|\mathbf{p}(\alpha)^+\| \leq \frac{1}{1 - \sqrt{1-\beta} + \frac{1}{2(1-\sqrt{\beta\tau})}}\left\|\left[\mathbf{e} - \mathbf{v}^2(\alpha)\right]^+\right\|.$$

To estimate the second term, we use Lemma 3.4, Lemma 3.7 and Lemma 3.5:

$$\begin{aligned}
\left\|\left[\mathbf{e} - \mathbf{v}^2(\alpha)\right]^+\right\| &= \frac{1}{\tau\mu(\alpha)}\left\|\left[\tau\mu(\alpha)\mathbf{e} - \mathbf{x}(\alpha)\mathbf{s}(\alpha)\right]^+\right\| \leq \frac{1}{\tau\mu(\alpha)}\left(\left\|\left[\tau\mu(\alpha)\mathbf{e} - \mathbf{h}(\alpha)\right]^+\right\| + \tau\mu\left\|\left[\mathbf{dx}(\alpha)\mathbf{ds}(\alpha)\right]^-\right\|\right) \\
&\leq \frac{1}{\tau\mu(\alpha)}\left(\beta\tau\mu(\alpha)\left(1 - \sqrt{1-\beta}\right) + \tau\mu\frac{\beta}{2}\right) \leq \frac{\beta}{\mu(\alpha)}\left(\mu(\alpha)\left(1 - \sqrt{1-\beta}\right) + \frac{1}{2}\frac{\mu(\alpha)}{1-\sqrt{\beta\tau}}\right) \\
&= \beta\left(1 - \sqrt{1-\beta} + \frac{1}{2}\frac{1}{1-\sqrt{\beta\tau}}\right).
\end{aligned}$$

Combining the two estimations completes the proof. $\square$

3.2. The complexity of the algorithms. At the end of the analysis, we prove that the general algorithmic framework has the best known iteration complexity when it is applied with a function from the proposed class (i.e., $p(t)$ satisfies (P_1)-(P_4)), and the parameters fulfill the conditions (C_1)-(C_3). As it has already been discussed in Section 3, the condition (P_4) was not mentioned in the analysis since, for the results that were proved earlier, this condition is not necessary. However, for our final theorem, we have to include this constraint to ensure that there exists a parameter pair (β, τ) that satisfies our conditions (C_1)-(C_3).

Theorem 3.12. *Suppose that the parameters $\beta, \tau \in (0, 1)$ and a function $p(t)$ are given and they satisfy all conditions (P_1)-(P_4) and (C_1)-(C_3) with constants $c > 0$ and $r > 0$. Assume that a starting point $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0) \in \mathcal{W}(\tau, \beta)$ is given. Let $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$. Then the algorithm provides an ε-optimal solution in*

$$O\left(\sqrt{n}\log\frac{\mathbf{x}_0^T\mathbf{s}_0}{\varepsilon}\right)$$

iterations.

Proof. According to Lemma 3.6, the following inequality holds for the duality gap in the kth iteration:

$$\frac{\mathbf{x}_k^T\mathbf{s}_k}{n} = \mu_k \leq \mu_{k-1}\left(1 - \left[(1-\tau)r - \sqrt{\beta\tau}c\right]\alpha_1\right) \leq \mu_0\left(1 - \left[(1-\tau)r - \sqrt{\beta\tau}c\right]\alpha_1\right)^k.$$

From this, it follows that

$$\mathbf{x}_k^T\mathbf{s}_k \leq \left(1-\left[(1-\tau)r-\sqrt{\beta\tau}c\right]\alpha_1\right)^k \mu_0 n.$$

Therefore $\mathbf{x}_k^T\mathbf{s}_k \leq \varepsilon$ holds if

$$\left(1-\left[(1-\tau)r-\sqrt{\beta\tau}c\right]\alpha_1\right)^k \mu_0 n \leq \varepsilon$$

is satisfied.
Taking the natural logarithm of both sides, we obtain

$$k\log\left(1-\left[(1-\tau)r-\sqrt{\beta\tau}c\right]\alpha_1\right)+\log(\mu_0 n)\leq \log\varepsilon.$$

Using the inequality $-\log(1-\vartheta)\geq\vartheta$, we can require the fulfillment of the stronger inequality

$$-k\alpha_1\left[(1-\tau)r-\sqrt{\beta\tau}c\right]+\log(\mu_0 n)\leq\log\varepsilon.$$

This inequality is satisfied when

$$k \geq c\sqrt{\frac{n}{\beta\tau}}\frac{1}{(1-\tau)r-\sqrt{\beta\tau}c}\log\left(\frac{\mathbf{x}_0^T\mathbf{s}_0}{\varepsilon}\right),$$

and this proves our statement. □

4. Constants and properties for special functions. In this section, we give some examples of functions that belong to the proposed class.
The first five rows of Table 1 show known functions from the literature on the AET method. The other rows show new functions that have been introduced in this context in this paper for the first time, up to the best of our knowledge. The rows 7-9 and 12 describe classes of functions, based on the values of the parameters k and m.
In the last four columns, we propose some values for the parameters c, r, β, and τ that satisfy the conditions (P_1)-(P_4) and (C_1)-(C_3). During the analysis, we fixed the value of γ as $\gamma=(1-\beta)\tau$, but even in this case there are different possible values for β and τ that satisfy the conditions.
The fourth function (introduced by Kheirfam and Haghighi [27]) does not belong to the function class defined in this paper since there is no positive constant c that satisfies the property (P_2). This means that the function $p(t)=1-t^2$ decreases too quickly over the interval $[1,t^*]$. Therefore for this function, the convergence of the general long-step algorithm cannot be proved with our approach.
The fifth transformation function was introduced by Illés et al. [25], and it has an inflection point at $t=1/4$. As it can be seen from Table 1, the ninth function is a generalization of the third one. In general, for this function $p(t)$ we cannot give a closed formula for the corresponding φ since the integral of $\frac{1}{p(t)}$ is a hypergeometric function. This function also shows that we can get a wider function class by building our analysis on the scaled system, since for a function $\varphi(t)$, the corresponding $p(t)$ can always be determined using (2.7).
All previous papers related to the AET technique consider continuous functions $p(t)$. However, the properties required for our analysis do not include the continuity of the function $p(t)$. Therefore, we can define noncontinuous functions $p(t)$ with finitely many jump discontinuities. One such example can be seen in the tenth row of Table 1. According to Lemma 5.1, there are infinitely many functions φ that give this function $p(t)$, the table shows only an example. This function φ is not differentiable in $\frac{1}{\sqrt{\tau}}$, but the convergence and best known iteration complexity still follows from our analysis. Since this function φ is still invertible, we can apply the AET technique by taking the right-hand derivative in formula (2.5) in the point $\frac{1}{\sqrt{\tau}}$. In general, invertible functions with finitely many jump discontinuities can be handled similarly.
In the eleventh and twelfth rows we defined functions $p(t)$ which are not strictly decreasing over $[1,t^*]$.

	$\varphi(t)$	$p(t)$	Conditions	ξ	c	r	β	τ
1.	t	$\frac{1}{t}-t$	-	0	1	1	$\frac{1}{8}$	$\frac{1}{8}$
2.	$\sqrt{t}$	$2(1-t)$	-	0	2	1	$\frac{1}{4}$	$\frac{1}{4}$
3.	$t-\sqrt{t}$	$\frac{2(t-t^2)}{2t-1}$	-	$\frac{1}{2}$	1	$\frac{8}{9}$	$\frac{1}{8}$	$\frac{1}{8}$
4.	$\frac{\sqrt{t}}{2(1+\sqrt{t})}$	$1-t^2$	-	0	$\nexists$	1	$\frac{1}{8}$	$\frac{1}{8}$
5.	$t^2-t+\sqrt{t}$	$\frac{2(1-t^4+t^2-t)}{4t^3-2t+1}$	-	0	1	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$
6.	$t\arctan t$	$\frac{\left(\pi/4-t^2\arctan t^2\right)}{t\left(\arctan t^2+\frac{t^2}{1+t^4}\right)}$	-	0	1	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$
7.	$t^k\ln t$	$\frac{-2t\ln t}{2k\ln t+1}$	$k\geq 1$	$e^{-\frac{1}{2k}}$	1	$\frac{1}{2k}$	$\frac{1}{8k}$	$\frac{1}{8k}$
8.	t^k	$\frac{1-t^{2k}}{kt^{2k-1}}$	$k\geq 1$	0	1	$\frac{1}{k}$	$\frac{1}{8k}$	$\frac{1}{8k}$
9.	No closed formula	$\frac{mt^k}{mt^k-1}(1-t)$	$m\geq 2,\ k\geq 1$	$m^{-\frac{1}{k}}$	1	$\frac{1}{2}$	$\frac{1}{8}\left(1-\frac{1}{\sqrt[k]{m}}\right)$	$\frac{1}{8}\left(1-\frac{1}{\sqrt[k]{m}}\right)$
10.	$\begin{cases} t & \text{if } t\leq\frac{1}{\sqrt{\tau}}, \\ \sqrt{t}+\sqrt{8}-\sqrt[4]{8} & \text{if } t>\frac{1}{\sqrt{\tau}} \end{cases}$	$\begin{cases} \frac{1}{t}-t & \text{if } t\leq\frac{1}{\sqrt{\tau}}, \\ 2\,(1-t) & \text{if } t>\frac{1}{\sqrt{\tau}} \end{cases}$	-	0	2	1	$\frac{1}{8}$	$\frac{1}{8}$
11.	No closed formula	$-\cos t\ln\left(\frac{1}{2}t\right)-t+\cos 1\ln\left(\frac{1}{2}\right)+1$	-	0	2	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$
12.	No closed formula	$k(\cos t-\cos 1)-t+1$	$k\in[1,2]$	0	2	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$

TABLE 1
Constants and properties for special functions

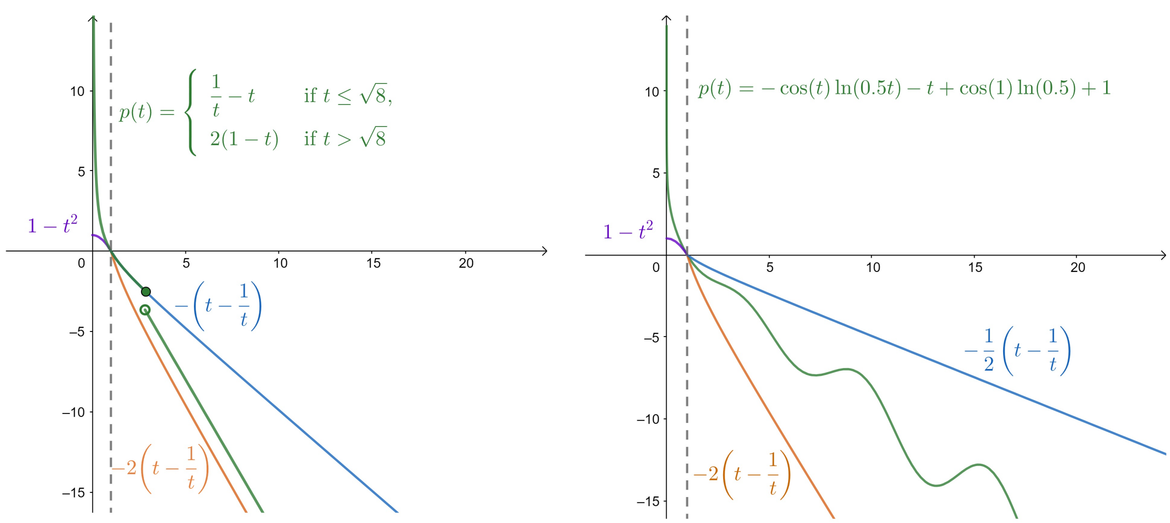

FIGURE 1. *Functions 10 and 11 and suitable lower and upper bounds*

5. Properties of the function φ. As can be seen in Section 3.1, our analysis only uses the properties of the function $p(t)$, and this function does not need to be written in the form (2.7) for some function φ. If we would like to decide for a function φ whether our analysis gives a convergent algorithm with complexity $O(\sqrt{n}L)$, it is enough to calculate the corresponding function p using (2.7), and then check whether the properties given in this paper are satisfied.

However, in this section, we still would like to show some important observations regarding the function φ as well. If we obtain p by using the AET technique, the functions p and φ are connected through (2.7), which can be viewed as a differential equation for φ when the value of p is given, and our goal is to determine which functions result in the same right-hand side vector in the Newton-system when using the AET method. To ensure that the expression in (2.7) is well-defined, we assume that $\varphi'(t)\neq 0$ for all $t\in(\xi^2,\infty)$.

The next lemma shows that if we multiply a function φ with a nonzero number or add a constant to it (i.e., shift the function vertically), then the corresponding function p remains the same. This is a simple consequence of the definition of p.

LEMMA 5.1. *Let $\varphi_1 : (\xi^2, \infty) \to \mathbb{R}$, $\xi \in [0, 1)$ be a continuously differentiable function with $\varphi_1'(u) \neq 0$ for all $u \in (\xi^2, \infty)$, and determine the corresponding function p_1 using the formula (2.7). Consider the function $\varphi_2(u) = \zeta\varphi_1(u) + \sigma$, where $\zeta, \sigma \in \mathbb{R}$ are given constants, $\zeta \neq 0$. Then for the corresponding function p_2 (calculated using (2.7) with φ_2)*

$$p_1(t) = p_2(t) \quad \forall\, t \in (\xi, \infty)$$

holds.

The previous lemma means that the same function p belongs to infinitely many functions φ.
Another important property here is that the sign of ζ is not restricted; the same right-hand side vector can belong to both increasing and decreasing functions. Therefore, the assumption that φ should be strictly increasing (which is mentioned in many papers using the AET method) is not necessary; it is enough to assume that this function is invertible and its derivative does not vanish over the domain. The first paper by Darvay that introduces the AET method [8] also requires only the invertibility and continuous differentiability of the function and does not mention its increasing (or decreasing) property.
According to Lemma 5.1, we can assume without loss of generality that $\varphi(u)$ is strictly increasing (so $\varphi'(u) > 0$ over its domain) and $\varphi(1) = 0$. These assumptions make it easier to formulate the necessary conditions for $\varphi(u)$. If we keep the original assumption of Darvay that $\varphi(u)$ should be continuously differentiable, then the properties (P_1)-(P_4) can be reformulated for the transformation function $\varphi(u)$ as follows:

- (P_1') The inequality $\frac{\varphi(u)}{\varphi'(u)} \leq -\sqrt{u}(1-u)$ holds for all $u \in (\xi^2, 1)$.
- (P_2') There exists a positive constant c for which $\frac{\varphi(u)}{\varphi'(u)} \leq c(u-1)$ for all $u \in [1, (t^*)^2]$.
- (P_3') There exists a positive constant r for which $\frac{\varphi(u)}{\varphi'(u)} \geq r(u-1)$ for all $u \in [1, (t^*)^2]$.
- (P_4') There exist two constants ϱ and η such that $\varrho \in [1, 2)$ and $\eta \in (\xi, 1)$ and $\frac{\varphi(u)}{\varphi'(u)} \geq -\varrho\sqrt{u}(1-u)$ for all $u \in (\eta^2, 1)$.

It can be observed that the conditions listed above do not refer to φ itself but to $\ln|\varphi|$, since the function $\frac{\varphi(u)}{\varphi'(u)}$ is the reciprocal of the derivative of $\ln|\varphi(u)|$

5.1. A modified AET technique with piecewise continuously differentiable transformation functions. For $p(t)$, we have already discussed that when all conditions are met, the function can have finitely many jump discontinuities, and the analysis still gives the desired results. The corresponding function $\varphi(t)$ can be determined by using one-sided derivatives at the discontinuities.
Having jump discontinuities in the function $\varphi(t)$ is also possible, and the monotonicity of φ can also change, provided that it remains invertible and all conditions necessary for our analysis are met; see, for example, Figure 2. However, the next corollary shows that it is enough to consider the case when $\varphi(t)$ is continuous and strictly increasing (or decreasing) over its whole domain.

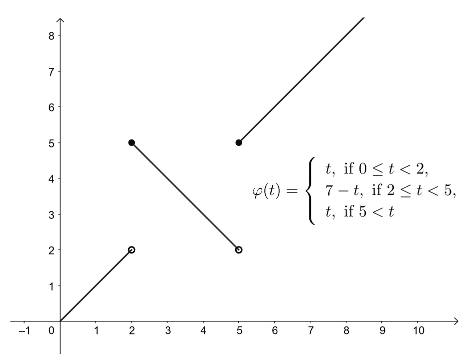

FIGURE 2. *A suitable function φ where the monotonicity changes.It gives the same function p as the identity function.*

COROLLARY 5.1. *Let $\varphi_1 : (\xi^2, \infty) \to \mathbb{R}$, $\xi^2 \in [0, 1)$ be a function with finitely many jump discontinuities or monotonicity changes and determine the corresponding function $p(t)$ using formula (2.7). Then we can* *define a continuous function $\varphi_2 : (\xi^2, \infty) \to \mathbb{R}$, $\xi^2 \in [0, 1)$ that strictly increases or decreases over (ξ^2, ∞), for which the corresponding function $p(t)$ is the same.*

Proof. If φ_1 is increasing or decreasing, but has jump discontinuities, then by using proper vertical shifts at the breaking points we can define a continuous function φ_2 and according to Lemma 5.1, these transformations do not modify the value of $p(t)$.
If the function φ_1 changes its monotonicity, then we can construct a strictly increasing function φ_2, by modifying over the intervals where φ_1 is decreasing, and multiply the value of φ_1 by -1. (A strictly decreasing function can be constructed similarly, by taking the intervals where φ_1 is increasing.) These changes do not modify $p(t)$, according to Lemma 5.1. After these, it is possible that φ_2 still has jump discontinuities, but these can be handled as in the previous case. $\square$

Therefore, in this subsection, we assume that φ is continuous, strictly increasing, and $\varphi(1) = 0$. However, we discard the differentiability assumption on φ and assume that there are finitely many points $t_1, t_2, \ldots, t_k$ where φ is not differentiable. If φ is calculated from a function p that satisfies our conditions (P$_1$)-(P$_4$), and has finitely many jump discontinuities in $t_1, t_2, \ldots, t_k$, then φ is continuously differentiable over the intervals (ξ^2, t_1^2) and (t_i^2, t_{i+1}^2), $i \in \{1, \ldots, k-1\}$ and the one-sided derivatives exist in $t_1^2, t_2^2, \ldots, t_k^2$. Thus, we suppose that φ is a piecewise continuously differentiable (PC^1) function.
We need a further assumption that the one-sided derivatives at the breaking points are not zero.
Let us define the function

$$d_\varphi(u) = \begin{cases} \varphi'(u), & \text{if } \exists\, i \in \{1, \ldots, k-1\} \text{ s.t. } u \in (t_i^2, t_{i+1}^2), \\ \lim_{z \to u^+} \varphi'(z) & \text{if } u \in \{t_1^2, t_2^2, \ldots, t_k^2\}. \end{cases}$$

In the definition of $d_\varphi(u)$, when $\varphi(u)$ is not differentiable, we replace $\varphi'(u)$ with the right-hand side derivative. However, we could use any of the one-sided derivatives in the definition instead, and the subsequent results would remain the same.
The properties of $\varphi(u)$ in the piecewise differentiable case can be formulated as:

(P$_1''$) The inequality $\frac{\varphi(u)}{d_\varphi(u)} \le -\sqrt{u}(1-u)$ holds for all $u \in (\xi^2, 1)$.

(P$_2''$) There exists a positive constant c for which $\frac{\varphi(u)}{d_\varphi(u)} \le c(u-1)$ for all $u \in [1, (t^*)^2]$.

(P$_3''$) There exists a positive constant r for which $\frac{\varphi(u)}{d_\varphi(u)} \ge r(u-1)$ for all $u \in [1, (t^*)^2]$.

(P$_4''$) There exist two constants ϱ and η such that $\varrho \in [1, 2)$ and $\eta \in (\xi, 1)$ and $\frac{\varphi(u)}{d_\varphi(u)} \ge -\varrho\sqrt{u}(1-u)$ for all $u \in (\eta^2, 1)$.

In the original AET technique of Darvay, the continuous differentiability of the transformation function is assumed so that Newton's method can be applied to determine the search directions. If we discard this assumption, we need to generalize the AET technique to the PC^1 case.
Instead of Newton's method, we can apply the extended Newton method proposed by Kojima and Shindo [29] in 1986. First, we need to show that the nonlinear mapping corresponding to the system (2.4) in the case of a PC^1 function φ is a piecewise continuously differentiable mapping from $\mathbb{R}^{n+m+n}$ to $\mathbb{R}^{n+m+n}$.

Definition 5.2 (Kojima and Shindo [29]). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous mapping and $\{U_i : \; i \in \Lambda\}$ be a countable family of closed subsets of $\mathbb{R}^n$ such that the following conditions are satisfied:*

- *$cl(int\ U_i) = U_i$ for every $i \in \Lambda$.*
- *$(int\ U_i) \cap (int\ U_j) = \emptyset$ for all $i, j \in \Lambda$ and $i \neq j$.*
- *$\bigcup_{i \in \Lambda} U_i = \mathbb{R}^n$.*
- *$\{U_i : \; i \in \Lambda\}$ has a locally finite property. (For any $\mathbf{x} \in \mathbb{R}^n$ there exists an open neighborhood V of $\mathbf{x}$ such that $\{i : \; V \cap U_i \neq \emptyset\}$ is finite.)*
- *For each $i \in \Lambda$ the restriction $F|_{U_i}$ of the mapping to each U_i is a continuously differentiable mapping.*

Then F is a PC^1 mapping on the subdivision $\{U_i : \; i \in \Lambda\}$ of $\mathbb{R}^n$.

Let Γ denote the set of all n-tuples of $\{0, 1, \ldots, k-1, k\}$. Then $|\Gamma| = (k+1)^n$. For each $\mathbf{r} \in \Gamma$ we can define a set $U_\mathbf{r}$ in the following way:

$$U_\mathbf{r} = \left\{ (\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathbb{R}^{n+m+n} : \; \mathbf{xs} \in \left(\bigtimes_{j=1}^{n} \left[t_{r_j}, t_{r_j+1} \right] \right), \; \mathbf{y} \in \mathbb{R}^m \right\},$$

where we applied the notations $t_0 = \xi^2$ and $t_{k+1} = \infty$. If we consider all $(k+1)^n$ such sets ($U_{\mathbf{r}}$ for all $\mathbf{r} \in \Gamma$), then the nonlinear mapping corresponding to system (2.4) is a PC^1 mapping on the subdivision $\{U_{\mathbf{r}} : \mathbf{r} \in \Gamma\}$ of $\mathbb{R}^{n+m+n}$.
Therefore, the extended Newton-method can be applied to (2.4), provided that φ is a piecewise continuously differentiable function. This way, for the Newton-directions, we get

$$\left.\begin{aligned} A\Delta\mathbf{x} &= \mathbf{0} \\ A^T\Delta\mathbf{y} + \Delta\mathbf{s} &= \mathbf{0} \\ \mathbf{s}\Delta\mathbf{x} + \mathbf{x}\Delta\mathbf{s} &= \mathbf{a}_\varphi = \nu\frac{\varphi(\mathbf{e}) - \varphi\left(\frac{\mathbf{xs}}{\nu}\right)}{d_\varphi\left(\frac{\mathbf{xs}}{\nu}\right)} \end{aligned}\right\} \tag{5.1}$$

In the definition of d_φ, we considered the right-hand side derivatives in the breaking points but mentioned that we could use any of the one-sided derivatives. The case is the same for the modified Newton-system (5.1). Therefore, using the one-sided derivatives, the AET technique can be applied even in the case when the transformation function is not continuously differentiable but piecewise continuously differentiable.

5.2. Construction rules. In this subsection, we give some lemmas on how new functions φ can be constructed using those for which the desired properties have already been proved. According to our numerical results, when we run the IPA with the theoretical step-length, all coordinates of the vector $\mathbf{v}$ are greater than 1 during the whole computation, and in addition, the conditions referring to the interval $[1,(t^*)^2]$ are much easier to check. Therefore, it is reasonable to concentrate on the validity of these construction rules over $[1,(t^*)^2]$. In this case, we need to check whether (P'$_2$) and (P'$_3$) are satisfied for the proposed new functions. Furthermore, the differentiability of φ is not necessary for our method to work, therefore, we can define φ with a different assignment rule over $(\xi, 1)$, for which the other conditions (P'$_1$) and (P'$_4$) are satisfied, and from (P'$_4$) it follows that suitable values exist for β and τ that satisfy (C$_1$) -(C$_3$).
To simplify our discussion regarding construction rules, we consider the case when φ is continuously differentiable over the interval $[1,(t^*)^2]$, but the results could be extended to the PC^1 case as well.
The following lemma shows that when two functions satisfy the given conditions over $[1,(t^*)^2]$, their product is also suitable.

Lemma 5.3. *Assume that the functions $\varphi_1(u): [1,(t^*)^2] \to \mathbb{R}$ and $\varphi_2(u): [1,(t^*)^2] \to \mathbb{R}$ are continuous and strictly increasing (or both strictly decreasing) over their domain and satisfy the conditions (P'$_2$) and (P'$_3$) with positive constants c_1, c_2, r_1 and r_2, respectively; furthermore, $\varphi_1(1) = 0$ and $\varphi_2(1) = 0$. Then their product $\varphi(u) = \varphi_1(u)\varphi_2(u)$ also satisfies the conditions with positive constants $c = \frac{c_1c_2}{c_1+c_2}$ and $r = \frac{r_1r_2}{r_1+r_2}$.*

Proof. Since φ_1 and φ_2 are continuous and strictly increasing or decreasing functions, their product is also continuous and strictly monotonic. Due to the monotonicity assumptions, φ_1 and φ_2 attain 0 only at $u = 1$, and the same holds for $\varphi(t)$.
For $u = 1$, the statement is true since both sides of the inequalities are equal to 0.
Suppose that $u > 1$, then using monotonicity and continuity assumptions on φ_1 and φ_2, (P'$_2$) and (P'$_3$) can be rewritten as

$$\frac{\varphi'(u)}{\varphi(u)} \geq \frac{1}{c(u-1)} \text{ and } \frac{\varphi'(u)}{\varphi(u)} \leq \frac{1}{r(u-1)}, \tag{5.2}$$

respectively.
Since $\varphi'(u) = \varphi_1'(u)\varphi_2(u) + \varphi_1(u)\varphi_2'(u)$,

$$\frac{\varphi'(u)}{\varphi(u)} = \frac{\varphi_1'(u)}{\varphi_1(u)} + \frac{\varphi_2'(u)}{\varphi_2(u)}.$$

Using the constraints on φ_1 and φ_2, it can be shown by a simple calculation that $\varphi(u)$ satisfies (P'$_2$) with $c = \frac{c_1c_2}{c_1+c_2}$, and (P'$_3$) with $r = \frac{r_1r_2}{r_1+r_2}$. ∎

Corollary 5.2. *Assume that the function $\varphi(u): [1,(t^*)^2] \to \mathbb{R}$ is continuous and strictly monotonic and satisfies the conditions (P'$_2$) and (P'$_3$) with c and r, and $\varphi(1) = 0$. Then the function $u^k\varphi(u)$ $(k \geq 1)$ also satisfies the conditions.*

Proof. Since we assumed that $\varphi(u)$ satisfies the above-mentioned conditions, and we know that the function u^k $(k \geq 1)$ also (see Table 1), the statement follows from Lemma 5.3. ∎

Lemma 5.4. *Assume that the functions* $\varphi_1(u) : [1,(t^*)^2] \to \mathbb{R}$ *and* $\varphi_2(u)$ $[1,(t^*)^2] \to \mathbb{R}$ *are continuous and strictly monotonic and both satisfy the conditions (P'*$_2$*) and (P'*$_3$*) with positive constants* c_1, c_2, r_1 *and* r_2, *respectively; furthermore,* $\varphi_1(1) = 0$ *and* $\varphi_2(1) = 0$. *Assume that* $\varphi_1(u) \geq 0$ *and* $\varphi_2(u) \geq 0$ *for all* $u \in (1,(t^*)^2)$. *Then their sum* $\varphi(u) = \varphi_1(u) + \varphi_2(u)$ *also satisfies the conditions over* $[1,(t^*)^2]$, *with* $c = \max\{c_1, c_2\}$ *and* $r = \min\{r_1, r_2\}$.

Proof. For $u = 1$, the statement holds since both sides of the inequalities are 0 according to our assumptions. Assume that $u > 1$ and consider the reformulated inequalities (5.2).
Since $\frac{\varphi'(u)}{\varphi(u)} = \frac{\varphi_1'(u)+\varphi_2'(u)}{\varphi_1(u)+\varphi_2(u)}$, using (P'$_2$) (in the form $\varphi_1(u) \leq c_1(u-1)$ and $\varphi_2(u) \leq c_2(u-1)$), we get

$$\frac{\varphi'(u)}{\varphi(u)} \geq \min\left\{\frac{1}{c_1}, \frac{1}{c_2}\right\} \frac{1}{u-1},$$

therefore $c = \max\{c_1, c_2\}$ is a suitable constant.
For the second inequality, using (P'$_3$) we obtain

$$\frac{\varphi'(u)}{\varphi(u)} \leq \max\left\{\frac{1}{r_1}, \frac{1}{r_2}\right\} \frac{1}{u-1},$$

thus $r = \min\{r_1, r_2\}$ is a proper constant for this constraint. ∎

When $\varphi_1(u) \leq 0$ and $\varphi_2(u) \leq 0$ hold for all $[1,(t^*)^2]$, their sum also satisfies the conditions, since $\varphi_1(u) + \varphi_2(u) = -((-\varphi_1(u)) + (-\varphi_2(u))$. Furthermore, $(-\varphi_1(u)) + (-\varphi_2(u))$ satisfies the conditions of Lemma 5.4, and if we multiply φ by a non-zero number, we get the same function p, according to Lemma 5.1.

Lemma 5.5. *Let* $\varphi(u)$ *be a strictly monotonic polynomial over* $[1,(t^*)^2]$, *and assume that* $\varphi(1) = 0$.
Then $\varphi(u)$ *satisfies the conditions (P'*$_2$*) and (P'*$_3$*).*

Proof. According to our assumptions, $\varphi(u) = (u-1)s(u)$, where $s(u)$ is a polynomial of degree at least 0. Furthermore, in the case of $u = 1$, the statement holds, since both sides of (P'$_2$) and (P'$_3$) are 0. Therefore, it can be assumed that $u > 1$.
Let us denote the degree of the polynomial $\varphi(u)$ by k. We apply induction on k.
When $k = 1$, the transformation function can be written as $\varphi(u) = \zeta(u-1)$, where ζ is a nonzero constant. In this case, $\frac{\varphi'(u)}{\varphi(u)} = \frac{1}{u-1}$, therefore, $c = 1$ and $r = 1$ are suitable constants, the constraints are satisfied for $k = 1$.
Assume that all polynomials of degree $k-1$ (which are invertible over $[1,(t^*)^2]$ with $\varphi(1) = 0$) satisfy the inequalities (P'$_2$) and (P'$_3$) with constants c_{k-1} and r_{k-1}.
Let $\varphi(u) = (u-1)s(u)$ be a polynomial of degree k (i.e., $s(u)$ is a polynomial of degree $k-1$). In this case,

$$\frac{\varphi'(u)}{\varphi(u)} = \frac{1}{u-1} + \frac{s'(u)}{s(u)}$$

By induction,

$$\frac{1}{c_{k-1}} \frac{1}{u-1} \leq \frac{s'(u)}{s(u)} \leq \frac{1}{r_{k-1}} \frac{1}{u-1}.$$

Using these inequalities, and choosing the values of c_k and r_k as

$$c_k = \frac{c_{k-1}}{c_{k-1}+1} \text{ and } r_k = \frac{r_{k-1}}{r_{k-1}+1},$$

it can be shown that (P'$_2$) and (P'$_3$) are satisfied.
Therefore, for a suitable polynomial of degree k, $c = \frac{1}{k}$ and $r = \frac{1}{k}$ are constants that satisfy the inequalities, and the statements of the lemma are true for general k. ∎

Besides the construction rules defined in the previous lemmas, we can also apply the results that follow from Corollary 5.1. Namely, we can multiply a function φ with a nonzero number, change the monotonicity of the transformation function, or add jump discontinuities, and the resulting function will remain in the defined function class.

6. Numerical results. In this section, we present our numerical results to examine the effect of the choice of the function $p(t)$ on the performance of the general long-step IPA.
First, we implemented a greedy variant of the examined algorithm. That is, we fixed the value of α_2 as 1 and calculated the value of α_1 as the largest step-length for which the new point remains in the neighborhood $\mathcal{W}(\tau,\beta)$. The pseudocode of the greedy variant is described in the following figure:

Algorithm 6.1 Outline of the greedy algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, a function $p(t) : (\xi,\infty) \to \mathbb{R}$
  an update parameter $0 < \tau < 1$, a neighborhood parameter $0 < \beta < 1$,
  an accuracy parameter $\varepsilon > 0$,
  a starting point $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s_0}) \in \mathcal{W}(\tau,\beta)$ with $\mu_0 = \frac{\mathbf{x}_0^T\mathbf{s}_0}{n}$
$\mathbf{x} := \mathbf{x}_0$, $\mathbf{y} := \mathbf{y}_0$, $\mathbf{s} := \mathbf{s}_0$ and $\mu := \mu_0$
while $\mathbf{x}^T\mathbf{s} > \varepsilon$ **do**
  Determine $\Delta\mathbf{x}_+, \Delta\mathbf{s}_+, \Delta\mathbf{y}_+$ and $\Delta\mathbf{x}_-, \Delta\mathbf{s}_-, \Delta\mathbf{y}_-$ according to (2.8)
  $\alpha_2 := 1$
  $\bar{\alpha}_1 := \max\{\alpha_1 \in [0,1] : (\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{s}(\alpha)) \in \mathcal{W}(\tau,\beta)\}$,
  where $\mathbf{x}(\alpha) = \mathbf{x} + \bar{\alpha}_1\Delta\mathbf{x}_- + \alpha_2\Delta\mathbf{x}_+$, $\mathbf{y}(\alpha) = \mathbf{y} + \bar{\alpha}_1\Delta\mathbf{y}_- + \alpha_2\Delta\mathbf{y}_+$ and $\mathbf{s}(\alpha) = \mathbf{s} + \bar{\alpha}_1\Delta\mathbf{s}_- + \alpha_2\Delta\mathbf{s}_+$
  $\mathbf{x} := \mathbf{x}(\alpha)$
  $\mathbf{y} := \mathbf{y}(\alpha)$
  $\mathbf{s} := \mathbf{s}(\alpha)$
  $\mu := \frac{\mathbf{x}^T\mathbf{s}}{n}$
end while

Since the step-length applied in the analysis ($\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta}{\tau n}}$) is a lower bound on the value of $\bar{\alpha}_1$ calculated in the greedy variant, and the duality gap is a strictly decreasing function of α_1, the complexity of the greedy IPA is at least as good as of the IPA investigated in the analysis.
The numerical tests were carried out on a Dell laptop with an Intel i7 processor and 16 GB RAM. We solved LP problem instances from the Netlib library [22]. After transforming the problem instances to standard form, we eliminated the redundant constraints by applying the procedure `eliminateRedundantRows.m` by Ploskas and Samaras [35]. For the modified LP problems, we applied a similar method to procedure CLEAN by Adler et al. [1] to eliminate fix-valued variables.
To start the algorithm from a strictly feasible initial point in the neighborhood $\mathcal{W}(\tau,\beta)$, we first transformed the problems into symmetric form, then applied the self-dual embedding technique [40]. To avoid doubling the number of constraints during the symmetric reformulation, we carried it out according to the last Remark of [26, p. 232]. For the embedded problem, $\mathbf{x} = \mathbf{e}$ and $\mathbf{s} = \mathbf{e}$ is a suitable initial point since it is strictly feasible and is in the neighborhood $\mathcal{W}(\tau,\beta)$.
The numerical results for the classic AET functions $\varphi(t) = t$, $\varphi(t) = \sqrt{t}$ and $\varphi(t) = t - \sqrt{t}$ can be found in our previous paper [20]. The running times required for the presolve and postsolve procedures can also be found in [20]. We also published some preliminary numerical results in [18] for the functions $\varphi(t) = t\arctan t$, $\varphi(t) = t^2$ and $\varphi(t) = t\ln t$.
For the numerical tests in this paper, we chose different functions from Table 1. Since according to Lemma 5.1, infinitely many functions φ belong to a function $p(t)$, instead of φ, we display the function p in Table 2. In the numerical tests, the applied functions were the following:

- $p_1(t) = \dfrac{1}{t} - t,$
- $p_2(t) = \dfrac{1-t^4}{2t^3},$
- $p_3(t) = 1 - t^2,$
- $p_4(t) = \dfrac{-2t\ln t}{4\ln t + 1},$
- $p_5(t) = \begin{cases} \frac{1}{t} - t & \text{if } t \leq \frac{1}{\sqrt{\tau}}, \\ 2\,(1-t) & \text{if } t > \frac{1}{\sqrt{\tau}}, \end{cases}$

- $p_6(t) = -\cos t \ln(t/2) - t - \cos 1 \ln(2) + 1$.

The value of the precision parameter ε was 10^{-5}. The chosen values of β and τ are shown in the second row of Table 2.
The function $p_1(t)$ corresponds to the case of $\varphi(t) = t$ and is shown for reference. As can be seen from Table 1, $p_3(t)$ does not belong to our function class, i.e. theoretically, the convergence of the general method cannot be proved for this particular function, but as can be seen from the numerical results, all problems could be solved using $p_3(t)$ as well, and the running times with this function were among the best.
The functions $p_2(t)$ and $p_4(t)$ are special cases of the functions given in the seventh and eighth rows of Table 1, with $k = 2$. Since the values of the parameters β and τ depend on the value of k, in these two cases we applied the settings $\beta = 1/16$ and $\tau = 1/16$. For the other functions, $\beta = 1/8$ and $\tau = 1/8$ were suitable parameter values that met all the conditions. Therefore, in the case of $p_2(t)$ and $p_4(t)$, the applied wide neighborhood was narrower, and we could only perform smaller updates when choosing our next target point. And this had a significant effect on the numerical results; the average number of iterations and running times were worse for these two functions. Based on these observations, choosing higher-order transformation functions for the AET method does not help numerically (even though we have already seen in Lemma 5.5 that there are polynomials of any degree in our function class).
The function $p_5(t)$ is the first AET function that has a jump discontinuity, and in terms of the average number of iterations, this gave the best result. If we consider only the five functions that belong to our class, this also gave the best average running time.
The function $p_6(t)$ is non-decreasing over $[1, \infty)$. So far, functions that change their monotonicity have not been examined in the context of the AET method; therefore, this function is important from a theoretical point of view. But, as can be seen in Table 2, it was not efficient in practice.

We also implemented the theoretical version of the algorithmic framework, where we applied the same steplengths as in the analysis, namely, $\alpha_1 = \frac{1}{c}\sqrt{\frac{\beta\tau}{n}}$ and $\alpha_2 = 1$. Since this variant is inefficient in practice, the main reason for implementing it was to make some observations on the behavior of the analyzed algorithm. With the staring points $\mathbf{x}_0 = \mathbf{e}$ and $\mathbf{s}_0 = \mathbf{e}$, at the beginning, all coordinates of the vector $\mathbf{v}$ are equal to $\frac{1}{\sqrt{\tau}}$. We observed that the coordinates remained in a really narrow interval around this value, not just for LP problems, but also for sufficient LCPs [19].
In the case of the theoretical variant, all coordinates are greater than 1, that is, the iterates never leave the neighborhood $\mathcal{N}_\infty^-(1-\tau)$. This is the main reason why we only considered the interval $[1, \infty)$ when introducing construction rules for the function $\varphi(t)$ in Section 5. In the case of the greedy variant, the largest coordinates of $\mathbf{v}$ are also far from the upper bound $\sqrt{n/\tau}$ applied in the analysis; in fact, the upper bound seems to be independent of the problem size even in this latter case. The lower bound for the greedy variant is always below 1, since we take the largest step for which the new iterate is still in $\mathcal{W}(\tau, \beta)$. The limits on the coordinates of $\mathbf{v}$ for four test problems are shown in Table 3.

7. Conclusion. In this paper, we introduced an Ai-Zhang-type long-step algorithmic framework for linear optimization. We also proposed a new class of AET functions for which the convergence and best known complexity of the general algorithm can be proved.
In the analysis, the function $p(t)$ derived from the right-hand side of the scaled Newton-system plays a far more significant role than the transformation function $\varphi(t)$ itself. Therefore, the conditions defining our function class have been formulated for the function $p(t)$. However, we also discussed a suitable set of sufficient properties for the function $\varphi(t)$ so that the corresponding function $p(t)$ is suitable for the general algorithm, and gave different construction rules that can be used to define new functions that belong to our class.
We also pointed out during the analysis that the continuity, monotonicity, and convexity assumptions often made in papers related to the AET technique can be relaxed, and the resulting functions can work well not just in theory but in practice as well. Following this line of thought, we generalized the AET technique for piecewise continuously differentiable transformation functions.
Furthermore, we implemented the general algorithmic framework in MATLAB and tested different transformation functions on problem instances from the Netlib library. We found that the performance of the methods for which the same parameter settings could be applied is similar for LP problems. We also made

			$p_1(t)$		$p_2(t)$		$p_3(t)$		$p_4(t)$		$p_5(t)$		$p_6(t)$	
			$\beta=\tau=\frac{1}{8}$		$\beta=\tau=\frac{1}{16}$		$\beta=\tau=\frac{1}{8}$		$\beta=\tau=\frac{1}{16}$		$\beta=\tau=\frac{1}{8}$		$\beta=\tau=\frac{1}{8}$	
Name	**m**	**n**	**Iter.**	**Sec.**	**Iter.**	**Sec.**	**Iter.**	**Sec.**	**Iter.**	**Sec.**	**Iter.**	**Sec.**	**Iter.**	**Sec.**
25fv47	1856	3712	88	1269.94	105	1512.84	81	1026.14	103	1478.27	79	999.36	100	1192.32
adlittle	139	278	22	0.34	23	0.38	22	0.33	22	0.35	22	0.30	24	0.36
afiro	53	106	15	0.04	16	0.06	16	0.04	16	0.04	16	0.04	18	0.04
agg	591	1182	36	21.22	48	29.35	31	15.63	44	26.38	32	16.11	45	21.45
agg2	758	15	35	41.01	44	51.71	33	34.35	41	47.78	32	32.01	43	39.92
agg3	758	15	44	51.79	60	70.67	38	37.25	56	65.33	36	35.88	52	47.96
bandm	418	836	42	9.74	53	12.66	38	7.69	50	12.12	37	7.57	49	9.28
beaconfd	222	444	18	0.90	25	1.20	16	1.09	22	1.04	20	1.08	22	0.86
blend	116	232	19	0.20	20	0.28	19	0.18	20	0.24	19	0.18	20	0.20
bnl1	1550	3100	103	886.69	136	1164.02	87	654.98	126	1081.27	83	625.76	131	941.94
bore3d	260	520	49	3.43	64	4.68	44	2.67	62	4.33	43	2.73	55	3.22
brandy	247	494	43	2.76	50	3.43	38	2.03	48	3.02	37	2.03	47	2.46
degen2	759	1518	24	29.01	26	30.89	24	24.62	25	29.73	25	25.55	27	26.08
e226	443	886	44	11.95	56	15.61	39	9.26	55	15.17	37	8.47	49	10.63
etamacro	869	1738	93	157.95	150	254.89	77	118.05	131	222.52	75	110.39	131	207.65
fffff800	1007	2014	80	203.84	106	277.41	69	152.29	98	250.03	65	143.86	97	235.00
finnis	1038	2076	70	194.14	99	272.90	63	151.88	90	249.28	63	150.36	86	226.52
fit1d	2077	4154	36	724.19	41	819.92	34	601.19	40	802.85	33	588.45	41	773.31
fit1p	2078	4156	38	761.87	46	915.07	35	618.90	43	855.96	36	635.58	43	809.35
ganges	1933	3866	25	411.37	28	460.17	24	351.52	27	441.87	26	371.26	32	493.58
gfrd_pnc	1404	2808	35	227.25	48	312.67	31	176.64	46	299.25	30	172.32	43	278.50
grow15	1247	2494	25	119.02	40	187.76	24	99.31	31	146.25	25	102.46	45	199.84
grow7	583	1166	26	15.28	34	19.76	24	11.50	29	17.39	24	12.20	38	21.26
israel	318	636	48	5.51	65	7.53	42	4.21	61	7.04	43	4.26	57	6.43
kb2	79	158	27	0.12	32	0.19	23	0.11	30	0.14	23	0.10	33	0.17
lotfi	366	732	29	4.70	39	6.62	28	3.93	33	5.58	28	4.17	36	5.96
nug05	227	454	11	0.57	11	0.64	14	0.62	11	0.58	15	0.71	14	0.80
nug06	488	976	13	4.86	12	4.36	15	4.80	13	4.92	16	4.99	15	5.45
nug07	933	1866	20	43.04	20	43.42	21	39.46	20	42.84	23	42.58	22	44.95
nug08	1634	3268	16	168.29	15	158.43	17	155.60	15	158.28	18	163.79	17	170.40
recipe	249	498	19	1.17	24	1.60	19	1.02	23	1.43	20	1.12	23	1.53
sc105	164	328	14	0.31	17	0.49	14	0.27	15	0.36	16	0.32	18	0.43
sc205	317	634	17	1.99	15	1.86	17	1.80	19	2.23	17	1.72	17	2.09
sc50a	79	158	14	0.09	17	0.11	15	0.07	16	0.10	15	0.07	17	0.11
sc50b	78	156	13	0.07	15	0.09	13	0.06	13	0.07	14	0.07	14	0.08
scagr25	672	1344	30	25.47	34	28.67	26	18.78	32	27.26	26	19.11	35	28.58
scagr7	186	372	21	0.63	23	0.82	19	0.49	23	0.67	19	0.52	25	0.75
scfxm1	583	1166	60	34.56	77	44.30	53	25.56	73	41.59	51	24.92	71	39.25
scfxm2	1164	2328	68	259.06	85	324.68	63	207.37	82	312.19	61	202.96	80	289.06
scfxm3	1745	3490	71	855.43	88	1057.33	63	660.59	84	1007.07	62	650.95	81	922.37
scrs8	1270	2540	68	331.70	92	446.30	62	260.76	84	409.68	58	244.69	82	377.31
sctap1	662	1324	54	42.90	61	49.09	55	36.48	61	49.56	53	36.23	59	45.58
scsd1	762	1524	17	20.85	18	21.74	17	16.90	17	21.10	18	19.19	19	21.86
scsd6	1352	2704	22	131.71	22	132.39	22	114.00	23	138.29	22	114.87	23	131.66
scsd8	2752	5504	19	878.98	18	837.57	20	813.24	18	835.24	22	885.31	21	913.11
share2b	164	328	21	0.44	25	0.56	20	0.34	25	0.54	21	0.40	23	0.50
ship04l	1965	3930	29	499.50	34	581.51	30	446.96	32	547.33	33	502.70	36	579.85
ship08s	1714	3428	40	459.16	55	628.24	37	370.29	51	581.44	38	398.16	49	533.78
standata	1350	2700	44	255.01	49	283.15	45	230.22	49	284.26	47	240.90	47	258.29
standgub	1351	2702	44	256.99	49	285.00	45	230.77	49	285.52	46	241.85	47	260.53
standmps	1350	2700	53	306.94	61	351.32	51	257.47	59	340.02	51	295.51	55	302.73
stocfor1	159	318	33	0.63	47	1.03	27	0.55	44	0.83	28	0.60	41	0.86
		Avg.	37.40	187.20	46.88	225.33	34.62	153.85	44.23	214.55	34.60	156.67	44.52	201.66

TABLE 2
Numerical results for the selected Netlib test problems

some interesting observations, namely that the higher-order or not monotonous transformation functions do not perform as efficiently in practice as other simpler functions, and therefore, they have theoretical rather than practical importance.

We also examined the implementation of our method with the theoretical step-lengths. Similarly to [19],

Name	**Function**	**Theoretical algorithm**		**Greedy algorithm**		**Theoretical upper bound**
		$v_{\min}$	$v_{\max}$	$v_{\min}$	$v_{\max}$	
afiro	$p_1(t)$	2.7869	2.8419	0.9568	4.9871	29.1205
	$p_5(t)$	2.8254	2.8313	0.9395	4.1782	
	$p_6(t)$	2.7588	2.8514	0.8322	4.5475	
blend	$p_1(t)$	2.7753	2.8396	0.9568	5.4633	43.0814
	$p_5(t)$	2.8267	2.8302	0.9568	4.1210	
	$p_6(t)$	2.7332	2.8470	0.8322	5.4240	
recipe	$p_1(t)$	2.7724	2.8341	0.9568	5.0944	63.1190
	$p_5(t)$	2.8271	2.8297	0.9395	4.0486	
	$p_6(t)$	2.7138	2.8392	0.8322	6.3958	
bandm	$p_1(t)$	2.7811	2.8346	0.9568	4.6539	81.7802
	$p_5(t)$	2.8275	2.8293	0.9395	4.3239	
	$p_6(t)$	2.7355	2.8416	0.8322	4.4222	

TABLE 3
Bounds on the coordinates of $\mathbf{v}$

we discovered that if the starting point is well-centered, then the coordinates of the vector $\mathbf{v}$ remain in the interval $[1, t^*]$. This means that allowing $v_i < 1$ in Ai-Zhang-type methods is mostly for theoretical reasons. It makes it possible to prove the complexity $O(\sqrt{n}L)$, but if we apply the step-length used during the analysis, the iterates always remain in the smaller neighborhood $\mathcal{N}_\infty^-(1-\tau)$.

In our future research, we would like to examine this latter phenomenon from a theoretical point of view and introduce similar AET function classes for other problems and other types of IPAs. Further, we would like to examine the relationship between our function class and others proposed in the literature. e.g., the classes of Haddou et al. [23] and Illés et al. [25]. In [23], the authors propose a general short-step weighted path-following method. They consider a different formulation of the central path problem, then apply a Darvay-type transformation without reorganizing the centering equation (similarly to [7]). Comparing their function class to ours raises several questions; for example, if we used the AET method described in [23], the function $p(t)$ could not be appropriately defined since it would change in every iteration. Illés et al. [25] proposed an algorithmic framework and a related function class for $\mathcal{P}_*(\kappa)$-LCPs. The main difference between their approach and ours is that they consider short-step IPAs while we analyze a long-step framework; furthermore, they fix the values of the algorithm parameters while we gave sets of feasible parameter values by formulating sufficient constraints on them.

Furthermore, we would like to compare our function class to other, not AET-related function classes, e.g., kernel function [30] and self-regular function-based approaches [33]. This is a promising research area since these methods consider suitable transformations of the right-hand side of the scaled system, similarly to our case. Since continuity and often convexity of the function on the right-hand side of the scaled Newton-system were assumed in the previously examined cases, we hope that the extensions proposed in this paper will open new possibilities in this area.

REFERENCES

[1] I. Adler, N. Karmarkar, M. G. Resende, and G. Veiga, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA J Comput, 1 (1989), pp. 84–106, https://doi.org/10.1287/ijoc.1.2.84.

[2] W. Ai and S. Zhang, *An $O(\sqrt{n}L)$ iteration primal-dual path-following method, based on wide neighborhoods and large updates, for monotone LCP*, SIAM J. Optim., 16 (2005), pp. 400–417, https://doi.org/10.1137/040604492.

[3] S. Asadi and H. Mansouri, *Polynomial interior-point algorithm for $\mathcal{P}_*(\kappa)$ horizontal linear complementarity problems*, Numer. Algorithm., 63 (2013), pp. 385–398, https://doi.org/10.1007/s11075-012-9628-0.

[4] S. Asadi, H. Mansouri, Z. Darvay, M. Zangiabadi, and N. Mahdavi-Amiri, *Large-neighborhood infeasible predictor–corrector algorithm for horizontal linear complementarity problems over Cartesian product of symmetric cones*, J. Optim. Theory Appl., 180 (2019), pp. 811–829, https://doi.org/10.1007/s10957-018-1402-6.

[5] S. Asadi, H. Mansouri, G. Lesaja, and M. Zangiabadi, *A long-step interior-point algorithm for symmetric cone Cartesian $\mathcal{P}_*(\kappa)$-HLCP*, Optimization, 67 (2018), pp. 2031–2060, https://doi.org/10.1080/02331934.2018.1512604.

[6] Y.-Q. Bai, M. El Ghami, and C. Roos, *A new efficient large-update primal-dual interior-point method based on a finite barrier*, SIAM Journal on Optimization, 13 (2002), pp. 766–782, https://doi.org/10.1137/S1052623401398132.

[7] Zs. Darvay, *A weighted-path-following method for linear optimization*, Studia Universitatis Babes-Bolyai, Series Informatica, 47 (2002), pp. 3–12.

[8] Zs. Darvay, *New interior point algorithms in linear programming*, Adv. Model. Optim., 5 (2003), pp. 51–92.
[9] Zs. Darvay, T. Illés, B. Kheirfam, and P. R. Rigó, *A corrector–predictor interior-point method with new search direction for linear optimization*, Cent. Eur. J. Oper. Res., 28 (2020), pp. 1123–1140, https://doi.org/10.1007/s10100-019-00622-3.
[10] Zs. Darvay, T. Illés, and Cs. Majoros, *Interior-point algorithm for sufficient LCPs based on the technique of algebraically equivalent transformation*, Optim. Lett., 15 (2021), pp. 357–376, https://doi.org/10.1007/s11590-020-01612-0.
[11] Zs. Darvay, T. Illés, J. Povh, and P. R. Rigó, *Feasible corrector-predictor interior-point algorithm for $\mathcal{P}_*(\kappa)$-linear complementarity problems based on a new search direction*, SIAM J. Optim., 30 (2020), pp. 2628–2658, https://doi.org/10.1137/19M1248972.
[12] Zs. Darvay, T. Illés, and P. R. Rigó, *Predictor-corrector interior-point algorithm for $\mathcal{P}_*(\kappa)$-linear complementarity problems based on a new type of algebraic equivalent transformation technique*, Eur. J. Oper. Res., (2021), https://doi.org/10.1016/j.ejor.2021.08.039.
[13] Zs. Darvay, I. M. Papp, and P. R. Takács, *Complexity analysis of a full-Newton step interior-point method for linear optimization*, Period. Math. Hung., 73 (2016), pp. 27–42, https://doi.org/10.1007/s10998-016-0119-2.
[14] Zs. Darvay and P. R. Rigó, *New interior-point algorithm for symmetric optimization based on a positive-asymptotic barrier function*, Numerical Functional Analysis and Optimization, 39 (2018), pp. 1705–1726, https://doi.org/10.1080/01630563.2018.1492938.
[15] Zs. Darvay and P. R. Rigó, *Interior-point algorithm for symmetric cone horizontal linear complementarity problems based on a new class of algebraically equivalent transformations*, Optim. Lett., 18 (2024), pp. 615–634, https://doi.org/10.1007/s11590-023-02020-w.
[16] Zs. Darvay and P. R. Takács, *Large-step interior-point algorithm for linear optimization based on a new wide neighbourhood*, Cent. Eur. J. Oper. Res., 26 (2018), pp. 551–563, https://doi.org/10.1007/s10100-018-0524-0.
[17] Zs. Darvay and P.-R. Takács, *New method for determining search directions for interior-point algorithms in linear optimization*, Optim. Lett., 12 (2018), pp. 1099–1116, https://doi.org/10.1007/s11590-017-1171-4.
[18] M. E.-Nagy and A. Varga, *A numerical comparison of long-step interior point algorithms for linear optimization*, SOR '21 Proceedings - The 16th International Symposium on Operational Research in Slovenia, (2021), pp. 75–80.
[19] M. E.-Nagy and A. Varga, *A new Ai–Zhang type interior point algorithm for sufficient linear complementarity problems*, J. Optim. Theor. Appl., (2022), https://doi.org/10.1007/s10957-022-02121-z.
[20] M. E.-Nagy and A. Varga, *A new long-step interior point algorithm for linear programming based on the algebraic equivalent transformation*, Cent. Eur. J. Oper. Res., (2022), https://doi.org/10.1007/s10100-022-00812-6.
[21] Z. Feng and L. Fang, *A new $O(nL)$-iteration predictor–corrector algorithm with wide neighborhood for semidefinite programming*, J. Comput. Appl. Math., 256 (2014), pp. 65–76, https://doi.org/10.1016/j.cam.2013.07.011.
[22] D. M. Gay, *Electronic mail distribution of linear programming test problems*, Math Program Soc COAL Newsl, 13 (1985), pp. 10–12.
[23] M. Haddou, T. Migot, and J. Omer, *A generalized direction in interior point method for monotone linear complementarity problems*, Optimization Letters, 13 (2019), pp. 35–53.
[24] T. Illés, P. R. Rigó, and R. Török, *Predictor-corrector interior-point algorithm based on a new search direction working in a wide neighbourhood of the central path*, Corvinus Econ. Work. Paper., (2021).
[25] T. Illés, P. R. Rigó, and R. Török, *Unified approach of primal-dual interior-point algorithms for a new class of AET functions*, Corvinus Econ. Work. Paper., (2022).
[26] B. Jansen, C. Roos, and T. Terlaky, *The theory of linear programming: skew symmetric self-dual problems and the central path*, Optim, 29 (1994), pp. 225–233, https://doi.org/10.1080/02331939408843952.
[27] B. Kheirfam and M. Haghighi, *A full-Newton step feasible interior-point algorithm for $\mathcal{P}_*(\kappa)$-LCP based on a new search direction*, Croat. Oper. Res. Rev., (2016), pp. 277–290, https://doi.org/10.17535/crorr.2016.0019.
[28] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in mathematical programming, Springer, 1989, pp. 29–47, https://doi.org/10.1007/978-1-4613-9617-8_2.
[29] M. Kojima and S. Shindo, *Extension of Newton and quasi-Newton methods to systems of PC^1 equations*, Journal of the Operations Research Society of Japan, 29 (1986), pp. 352–375, https://doi.org/10.15807/jorsj.29.352.
[30] G. Lesaja and C. Roos, *Unified analysis of kernel-based interior-point methods for $\mathcal{P}_*(\kappa)$-linear complementarity problems*, SIAM Journal on Optimization, 20 (2010), pp. 3014–3039, https://doi.org/10.1137/090766735.
[31] Y. Li and T. Terlaky, *A new class of large neighborhood path-following interior point algorithms for semidefinite optimization with $O\left(n\log\frac{Tr(X^0S^0)}{\varepsilon}\right)$ iteration complexity*, SIAM. J. Optim., 20 (2010), pp. 2853–2875, https://doi.org/10.1137/080729311.
[32] C. Liu, H. Liu, and W. Cong, *An $O(\sqrt{n}L)$ iteration primal-dual second-order corrector algorithm for linear programming*, Optim. Lett., 5 (2011), pp. 729–743, https://doi.org/10.1007/s11590-010-0242-6.
[33] J. Peng, C. Roos, and T. Terlaky, *Self-regularity: A New Paradigm for Primal-dual Interior-point Algorithms*, Princeton Series in Applied Mathematics, Princeton University Press, 2002, https://doi.org/10.2307/j.ctt7sf0f.
[34] M. Pirhaji, H. Mansouri, and M. Zangiabadi, *An $O\left(\sqrt{n}L\right)$ wide neighborhood interior-point algorithm for semidefinite optimization*, Comput. Appl. Math., 36 (2017), pp. 145–157, https://doi.org/10.1007/s40314-015-0220-9.
[35] N. Ploskas and N. Samaras, *Linear Programming Using MATLAB®*, vol. 127, Springer, 2017, https://doi.org/10.1007/978-3-319-65919-0.
[36] F. A. Potra, *Interior point methods for sufficient horizontal LCP in a wide neighborhood of the central path with best known iteration complexity*, SIAM. J. Optim., 24 (2014), pp. 1–28, https://doi.org/10.1137/120884341.
[37] P. R. Rigó, *New trends in algebraic equivalent transformation of the central path and its applications*, PhD thesis, Budapest University of Technology and Economics, 2020.
[38] G. Sonnevend, *An "analytical centre" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in System modelling and optimization, Springer, 1986, pp. 866–875.

[39] X. Yang, Y. Zhang, and H. Liu, *A wide neighborhood infeasible-interior-point method with arc-search for linear programming*, J. Appl. Math. Comput., 51 (2016), pp. 209–225, https://doi.org/10.1007/s12190-015-0900-z.
[40] Y. Ye, M. J. Todd, and S. Mizuno, *An $O\left(\sqrt{n}L\right)$-iteration homogeneous and self-dual linear programming algorithm*, Math Oper Res, 19 (1994), pp. 53–67, https://doi.org/10.1287/moor.19.1.53.