RAIRO-Oper. Res. 51 (2017) 299–328
DOI: 10.1051/ro/2016020

RAIRO Operations Research
www.rairo-ro.org

PRIMAL-DUAL ENTROPY-BASED INTERIOR-POINT ALGORITHMS FOR LINEAR OPTIMIZATION

Mehdi Karimi[1,*], Shen Luo[2,**] and Levent Tunçel[1,***]

Abstract. We propose a family of search directions based on primal-dual entropy in the context of interior-point methods for linear optimization. We show that by using entropy-based search directions in the predictor step of a predictor-corrector algorithm together with a homogeneous self-dual embedding, we can achieve the current best iteration complexity bound for linear optimization. Then, we focus on some wide neighborhood algorithms and show that in our family of entropy-based search directions, we can find the best search direction and step size combination by performing a plane search at each iteration. For this purpose, we propose a heuristic plane search algorithm as well as an exact one. Finally, we perform computational experiments to study the performance of entropy-based search directions in wide neighborhoods of the central path, with and without utilizing the plane search algorithms.

Mathematics Subject Classification. 90C05, 90C51, 41A46, 94A17.

Received October 28, 2014. Accepted January 22, 2016.

1. Introduction

Primal-dual interior-point methods have been proven to be one of the most useful algorithms in the area of modern interior-point methods for solving linear programming (LP) problems. In this paper, we are interested in a class of path-following algorithms that generate a sequence of primal-dual iterates within certain neighbourhoods of the central path. Several algorithms in this class have been studied, which can be distinguished by the choice of search direction. We introduce a family of search directions inspired by nonlinear reparametrizations of the central path equations, as well as the concept of entropy. Entropy and the underlying functions have been playing important roles in many different areas in mathematics, mathematical sciences, and engineering; such as partial differential equations [31], information theory [5, 32], signal and image processing [4, 9, 23], smoothing techniques [29], dynamical systems [8], and various topics in optimization [7, 12–14, 18]. In the context

Keywords. Interior-point methods, primal-dual entropy, central path, homogeneous and self-dual embedding, search direction.

* Research of this author was supported in part by OGS scholarships from the government of Ontario, a Discovery Grant from NSERC, and by ONR Research Grant N00014-12-1-0049.
** Research of this author was supported in part by an NSERC Discovery Grant.
*** Research of this author was supported in part by Discovery Grants from NSERC and by ONR Research Grant N00014-12-1-0049.

[1] Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. m7karimi@uwaterloo.ca; ltuncel@uwaterloo.ca

[2] Toronto, Ontario, Canada. shenluo@alumni.uwaterloo.ca

of primal-dual algorithms, we use the entropy function in determining the search directions as well as measuring centrality of primal-dual iterates.

Consider the following form of LP and its dual problem:

$$\begin{array}{llll} \text{(P)} & \text{minimize} & c^\top x & \\ & \text{subject to} & Ax = b, & x \geq 0, \end{array}$$

$$\begin{array}{llll} \text{(D)} & \text{maximize} & b^\top y & \\ & \text{subject to} & A^\top y + s = c, & s \geq 0, \end{array}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$ are given data. Without loss of generality, we always assume A has full row rank, *i.e.*, $\text{rank}(A) = m$. We assume throughout this paper that positive integers m and n satisfy $n \geq m + 1 \geq 2$. Let us define $\mathcal{F}$ and $\mathcal{F}_+$ as

$$\begin{aligned} \mathcal{F} &:= \{(x, s) : \ Ax = b, \ A^\top y + s = c, \ x \geq 0, s \geq 0, y \in \mathbb{R}^m\}, \\ \mathcal{F}_+ &:= \{(x, s) : \ Ax = b, \ A^\top y + s = c, \ x > 0, s > 0, y \in \mathbb{R}^m\}. \end{aligned}$$

Next, we define the standard primal-dual central path with parameter $\mu > 0$, *i.e.* $\mathcal{C} := \{(x_\mu, s_\mu) : \mu > 0\}$, as the solutions of the following system:

$$\begin{aligned} A^\top y + s &= c, \ s > 0 \\ Ax &= b, \ x > 0 \\ Xs &= \mu e, \end{aligned} \tag{1.1}$$

where e is the all ones vector whose dimension will be clear from the context (in this case n). The above system has a unique solution for each $\mu > 0$. For every pair $(x, s) \in \mathcal{F}$, we define the average duality gap as $\mu := \frac{x^T s}{n}$.

In standard primal-dual algorithms, search direction is found by applying a Newton-like method to the equations in system (1.1) with an appropriate value of μ_+ and the current point as the starting point. Explicitly, the search direction at a point $(x, s) \in \mathcal{F}_+$ is the solution of the following linear system of equations:

$$\begin{bmatrix} 0 & A^\top & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \mu_+ e \end{bmatrix}. \tag{1.2}$$

The first two blocks of equations in (1.1) are linear and as a result, they are perfectly handled by Newton's method. The nonlinear equation $Xs = \mu_+ e$ plays a very critical role in Newton's method. Now, if we apply a continuously differentiable strictly monotone function $f : \mathbb{R}_+ \to \mathbb{R}$ to both sides of $Xs = \mu_+ e$ (element-wise), clearly the set of solutions of (1.1) does not change, but the solutions of Newton system might change dramatically. This reparametrization of the KKT system can potentially give us an infinite number of search directions, but not all of them would have desirable properties. For a diagonal matrix V, let $f(V)$ and $f'(V)$ denote diagonal matrices with the jth diagonal entry equal to $f(v_j)$ and $f'(v_j)$, respectively. Replacing $Xs = \mu_+ e$ with $f(Xs) = f(\mu_+ e)$ and applying Newton's method gives us the same system as (1.2) with the last equation replaced by (see [35]):

$$Sd_x + Xd_s = (f'(XS))^{-1}(f(\mu_+ e) - f(Xs)). \tag{1.3}$$

This kind of reparametrization has connections to *Kernel functions* in interior-point methods (see our discussion in Appendix A).

Every choice of a continuously differentiable strictly monotone function f in (1.3) gives us a search direction. These search directions include some of the previously studied ones. For example, the choice of $f(x) = 1/x$

gives the search direction proposed in [24] (also see [25] for another connection to the entropy function), and the choice of $f(x) = \sqrt{x}$ leads to the work in [6]. A natural choice for f is $\ln(\cdot)$, which has been studied in [28,35,38]. Substituting $f(x) = \ln(x)$ in (1.3) results in

$$Sd_x + Xd_s = -(XS)\ln\left(\frac{Xs}{\mu_+}\right)\cdot \tag{1.4}$$

This is the place where entropy function comes into play. In this paper, we study the behaviour of the search direction derived by using (1.4), with an appropriate choice of μ_+. This search direction corresponds to the gradient of the primal-dual entropy-based potential function $\psi(x,s) := \frac{1}{\mu}\sum_{j=1}^{n} x_j s_j \ln(x_j s_j)$ (see [35]). As in [35], we define a proximity measure $\delta(x,s)$ as:

$$\delta(x,s) := \sum_{j=1}^{n} \frac{x_j s_j}{n\mu} \ln\left(\frac{x_j s_j}{\mu}\right)\cdot \tag{1.5}$$

We sometimes drop (x,s) in $\delta(x,s)$ when the argument of δ is clear from the context. If we choose μ_+ such that $\ln\left(\frac{\mu}{\mu_+}\right) = 1 - \delta(x,s)$, then (1.4) is reduced to

$$Sd_x + Xd_s = -Xs + \left[\delta Xs - (XS)\ln\left(\frac{Xs}{\mu}\right)\right]\cdot \tag{1.6}$$

This is exactly the search direction studied in [35] for the following neighborhood (of the central path)

$$\mathcal{N}_E(\beta) := \left\{(x,s) \in \mathcal{F}_+ : \frac{1}{2} - \beta \leq \ln\left(\frac{x_j s_j}{\mu}\right) \leq \frac{1}{2} + \beta,\ \text{for all } j\right\},$$

where $\beta \geq \frac{1}{2}$. It is proved in [35] that we can obtain the iteration complexity bound of $O\left(n \ln\left(\frac{1}{\epsilon}\right)\right)$ for $\mathcal{N}_E(3/2)$. We will generalize (1.6) to define our family of entropy-based search directions.

In the vast literature on primal-dual interior-point methods, two of the closest treatments to ours are [35,38]. Our search directions unify and generalize the search directions introduced in [35,38]. Besides that, for infeasible start algorithms, we use homogeneous self-dual embedding proposed in [37]. In this approach, we combine the primal and dual problems into an equivalent homogeneous self-dual LP with an available starting point of our choice. It is proved in [37] that we can achieve the current best iteration complexity bound of $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$ by using this approach. See Appendix B for a definition of homogeneous self-dual embedding and the properties of it that we need.

In Section 2, we introduce our family of search directions that generalizes and unifies those proposed in [35,38], and prove some basic properties. In Section 3, we use the entropy-based search direction in the predictor step of a predictor-corrector algorithm for the *narrow neighborhood* of the central path

$$\mathcal{N}_2(\beta) := \left\{(x,s) \in \mathcal{F}_+ : \left\|\frac{Xs}{\mu} - e\right\|_2 \leq \beta\right\},$$

and prove that we can obtain the current best iteration complexity bound of $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$. After that, we focus on the *wide neighborhood*

$$\mathcal{N}_\infty^-(\beta) := \left\{(x,s) \in \mathcal{F}_+ : \frac{x_j s_j}{\mu} \geq 1 - \beta,\ \text{for all } j\right\},$$

and work with our new family of search directions, parameterized by η (which indicates the weight of a component of the search direction that is based on primal-dual entropy). For various primal-dual interior-point algorithms utilizing the wide neighborhood, see [21,26,30,34] and the references therein. In Section 4, we derive some theoretical results for the wide neighborhood. However, our main goal in the context of wide neighborhood algorithms is to investigate the best practical performance for this class of search directions, in terms of total number of iterations. At each iteration, to find the best search direction in the family (*i.e.* the best value of η) that gives us the longest step (and hence the largest decrease in the duality gap), we perform a plane search. For this purpose, we propose a heuristic plane search algorithm as well as an exact one in Section 5. Then, in Section 6, we perform computational experiments to study the performance of entropy-based search directions with and without utilizing the plane search. Our computational experiments are on a class of classical small dimensional problems from NETLIB library [27]. Section 7 is the conclusion of this paper.

2. Entropy-based search directions and basic properties

In this section, we derive some useful properties for analyzing our algorithms. It is more convenient to work in the scaled v-space. Let us define

$$\begin{aligned} v &:= X^{1/2}S^{1/2}e, \\ u &:= \frac{1}{\mu}Xs = \frac{1}{\mu}Vv. \end{aligned} \tag{2.1}$$

We define the scaled right-hand-side vector with parameter $\eta \in \mathbb{R}_+$ as

$$w(\eta) := -v + \eta\left[\delta v - V\ln\left(\frac{Vv}{\mu}\right)\right]. \tag{2.2}$$

This definition generalizes and unifies the search directions proposed in [35] ($\eta = 1$) and [38] ($\eta = \frac{1}{\sigma}$ with $\sigma \in (0.5, 1)$ and $\sigma < \min\left\{1, \ln\left(\frac{1}{1-\beta}\right)\right\}$). For simplicity, we write $w := w(1)$, which is the scaled right-hand-side vector of (1.6). By using (2.1), we can also write $\delta = \frac{1}{n}\sum_{j=1}^{n} u_j \ln(u_j)$. If we define $[w(\eta)]_p$ as the projection of $w(\eta)$ on the null space of the scaled matrix $\bar{A} := AD$, where $D := X^{1/2}S^{-1/2}$, and define $[w(\eta)]_q := w(\eta) - [w(\eta)]_p$, then in the original space, the primal and dual search directions are $d_x = D\bar{d}_x$ and $d_s = D^{-1}\bar{d}_s$, respectively, where $\bar{d}_x := [w(\eta)]_p$ and $\bar{d}_s := [w(\eta)]_q$. In other words, the scaled search directions, *i.e.*, $\bar{d}_x$ and $\bar{d}_s$, can be obtained from the unique solution of the following system:

$$\begin{bmatrix} 0 & \bar{A}^\top & I \\ \bar{A} & 0 & 0 \\ I & 0 & I \end{bmatrix}\begin{bmatrix} \bar{d}_x \\ d_y \\ \bar{d}_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ w(\eta) \end{bmatrix}. \tag{2.3}$$

Note that d_y in the above system (2.3) is the same d_y as in (1.2) since $A^\top d_y + d_s = 0$ if and only if $0 = DA^\top d_y + Dd_s = \bar{A}^\top d_y + \bar{d}_s$. Most of the upcoming results in this section are for the neighborhood $\mathcal{N}_\infty$ defined as:

$$\mathcal{N}_\infty(\beta) := \left\{(x,s) \in \mathcal{F}_+ : \left\|\frac{Xs}{\mu} - e\right\|_\infty \le \beta\right\}.$$

We also use some of these results for $\mathcal{N}_2$ (since $\mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta)$ for all $\beta > 0$, this is valid). Let us start with the following lemma (see [35]):

Lemma 2.1. *For every $x > 0$, $s > 0$, we have:*

1. $\delta \ge 0$*;*
2. *equality holds above if and only if $Xs = \mu e$.*

The following lemma is well-known and is commonly used in the interior-point literature and elsewhere. See, for instance, Lemma 4.1 in [16] and Lemma 1 in [36].

Lemma 2.2. *For every $\alpha \in \mathbb{R}$ such that $|\alpha| \le 1$, we have:*

$$\alpha - \frac{\alpha^2}{2(1-|\alpha|)} \le \ln(1+\alpha) \le \alpha\cdot$$

Remark 2.3. The right-hand-side inequality above holds for every $\alpha \in (-1, +\infty)$.

Next, we relate the primal-dual proximity measure $\delta(x,s)$ to a more commonly used 2-norm proximity measure for the central path.

Lemma 2.4. *Let $\beta \in [0,1)$ such that $(x,s) \in \mathcal{N}_\infty(\beta)$. Then,*

$$\frac{1-3\beta}{2(1-\beta)n}\left\|\frac{Xs}{\mu}-e\right\|_2^2 \le \delta(x,s) \le \frac{1}{n}\left\|\frac{Xs}{\mu}-e\right\|_2^2.$$

Proof. The right-hand-side inequality was proved in [35]. We prove the left-hand-side inequality here. Let $\beta \in [0,1)$, such that $(x,s) \in \mathcal{N}_\infty(\beta)$. Then, we have (estimations are done in the u-space):

$$\begin{aligned}
\delta(u) = \frac{1}{n}\sum_{j=1}^n u_j \ln(u_j) &\ge \frac{1}{n}\sum_{j=1}^n u_j\left[u_j - 1 - \frac{(u_j-1)^2}{2(1-|u_j-1|)}\right] \\
&= \frac{1}{n}\|u-e\|_2^2 - \frac{1}{2n}\sum_{j=1}^n \frac{u_j}{(1-|u_j-1|)}(u_j-1)^2 \\
&\ge \frac{1}{n}\|u-e\|_2^2 - \frac{(1+\beta)}{2n(1-\beta)}\|u-e\|_2^2 \\
&= \frac{1-3\beta}{2(1-\beta)n}\left\|\frac{Xs}{\mu}-e\right\|_2^2.
\end{aligned}$$

In the above, the first inequality uses Lemma 2.2, the first equality uses $\sum_{j=1}^n u_j = n$, and the second inequality follows from the fact that $(x,s) \in \mathcal{N}_\infty(\beta)$. □

Corollary 2.5. *For every $(x,s) \in \mathcal{N}_\infty\left(\frac{1}{4}\right)$, $\delta \ge \frac{1}{6n}\left\|\frac{Xs}{\mu}-e\right\|_2^2$. Moreover, for every $(x,s) \in \mathcal{N}_\infty\left(\frac{1}{10}\right)$, $\delta \ge \frac{7}{18n}\left\|\frac{Xs}{\mu}-e\right\|_2^2$.*

Next, we want to study the behaviour of the search direction $w = -v + \delta v - V\ln(\frac{Vv}{\mu})$. We already have upper and lower bounds on δ, so we can easily estimate $-v+\delta v$. Next, we estimate $V\ln(\frac{Vv}{\mu})$ within the neighborhood $\mathcal{N}_\infty(\beta)$.

Lemma 2.6. *Let $\beta \in [0, \frac{1}{2})$. Then, for every $(x,s) \in \mathcal{N}_\infty(\beta)$, we have:*

$$\left(\delta(u) - 2 - \frac{\beta^2}{4\beta^2 - 6\beta + 2}\right)v + \mu V^{-1}e \le w \le (\delta(u)-2)v + \mu V^{-1}e.$$

Proof. Let $(x,s) \in \mathcal{N}_\infty(\beta)$ for some $\beta \in [0,\frac{1}{2})$. Then, $(1-\beta)e \le \frac{Vv}{\mu} \le (1+\beta)e$.
On the one hand, using Lemma 2.2, we have

$$-V\ln\left(\frac{Vv}{\mu}\right) = V\ln(\mu V^{-2}e) = V\ln(e + \mu V^{-2}e - e) \le V(\mu V^{-2}e - e) = \mu V^{-1}e - v.$$

On the other hand, using Lemma 2.2 again and the fact that $(x,s) \in \mathcal{N}_\infty(\beta)$, $\beta \in [0, \frac{1}{2})$, for every $j \in \{1, 2, \dots, n\}$, we have

$$v_j \ln\left(\frac{\mu}{v_j^2}\right) = \sqrt{\mu u_j} \ln\left(\frac{1}{u_j}\right) \geq \sqrt{\frac{\mu}{u_j}} - \sqrt{\mu u_j}\left[1 + \frac{\left(\frac{1}{u_j} - 1\right)^2}{2\left(1 - \left|\frac{1}{u_j} - 1\right|\right)}\right].$$

To justify some of the remarks following this proof, we focus on the cases

- $u_j \in [1-\beta, 1]$,
- $u_j \in [1, 1+\beta]$.

Case 1. ($u_j \in [1-\beta, 1]$): Using the derivation above, we further compute

$$\begin{aligned}
v_j \ln\left(\frac{\mu}{v_j^2}\right) &\geq \sqrt{\frac{\mu}{u_j}} - \sqrt{\mu u_j}\left[1 + \frac{\left(\frac{1}{u_j} - 1\right)^2}{2\left(1 - \left|\frac{1}{u_j} - 1\right|\right)}\right] \\
&= \frac{\mu}{v_j} - v_j\left[1 + \frac{(1-u_j)^2}{2u_j^2\left(2 - \frac{1}{u_j}\right)}\right] \\
&= \frac{\mu}{v_j} - v_j\left[1 + \frac{(1-u_j)^2}{2u_j\left(2u_j - 1\right)}\right] \\
&\geq \frac{\mu}{v_j} - v_j\left[1 + \frac{\beta^2}{2(1-\beta)(1-2\beta)}\right].
\end{aligned}$$

Case 2. ($u_j \in [1, 1+\beta]$): Again, using the derivation before this case analysis, we further compute

$$\begin{aligned}
v_j \ln\left(\frac{\mu}{v_j^2}\right) &\geq \sqrt{\frac{\mu}{u_j}} - \sqrt{\mu u_j}\left[1 + \frac{\left(\frac{1}{u_j} - 1\right)^2}{2\left(1 - \left|\frac{1}{u_j} - 1\right|\right)}\right] \\
&= \frac{\mu}{v_j} - v_j\left[1 + \frac{(1-u_j)^2}{2u_j}\right] \\
&= \frac{\mu}{v_j} - v_j\left(\frac{u_j^2 + 1}{2u_j}\right) \\
&\geq \frac{\mu}{v_j} - v_j\left[1 + \frac{\beta^2}{2(1+\beta)}\right] \\
&\geq \frac{\mu}{v_j} - v_j\left[1 + \frac{\beta^2}{2(1-\beta)(1-2\beta)}\right],
\end{aligned}$$

where the last inequality uses the fact that $2\beta^2 - 3\beta + 1 \leq 1 + \beta$ for $\beta \in [0, \frac{1}{2})$. Therefore, within the neighborhood $\mathcal{N}_\infty(\beta)$, for $\beta \in [0, \frac{1}{2})$, we conclude that the claimed relation holds. □

Remark 2.7. Focusing on the case analysis in the last proof, we see that for those j with $x_j s_j \geq \mu$ (Case 2), the corresponding component w_j of w is very close to the corresponding component computed for a generic primal-dual search direction. For example, for $\beta \in [0, 1/4]$,

$$\left(\delta(u) - 2 - \frac{1}{40}\right) v_j + \frac{\mu}{v_j} \leq w_j \leq (\delta(u) - 2)\, v_j + \frac{\mu}{v_j}.$$

Corollary 2.8. *For every* $(x, s) \in \mathcal{N}_\infty\left(\frac{1}{4}\right)$,

$$\left(\delta(u) - 2 - \frac{1}{12}\right) v + \mu V^{-1} e \leq w \leq (\delta(u) - 2) v + \mu V^{-1} e.$$

Remark 2.9. Recall that in a generic primal-dual search direction, w is replaced by $\left[-v + \gamma\mu V^{-1} e\right]$, $\gamma \in [0, 1]$ being the centering parameter. The above corollary shows that inside the neighborhood $\mathcal{N}_\infty(\frac{1}{4})$,

$$\left[-1 - \frac{1}{12(2 - \delta(u))}\right] v + \frac{1}{2 - \delta(u)} \mu V^{-1} e \leq \frac{w}{2 - \delta(u)} \leq -v + \frac{1}{2 - \delta(u)} \mu V^{-1} e.$$

Since by Lemma 2.4, inside the neighborhood $\mathcal{N}_\infty(1/4)$ we have $\delta(u) \leq 1/16$, working with w is close to setting the centrality parameter $\gamma :\approx \frac{1}{2}$.

Let us define the following quantities which play an important role in analysis of our algorithms:

$$\Delta_{21}(u) := \sum_{j=1}^{n} u_j^2 \ln(u_j), \quad \Delta_{12}(u) := \sum_{j=1}^{n} u_j \ln^2(u_j), \quad \Delta_{22}(u) := \sum_{j=1}^{n} u_j^2 \ln^2(u_j).$$

We drop the argument u, (*e.g.* we write Δ_{ij} instead of $\Delta_{ij}(u)$) when u is clear from the context. The next few results provide bounds on the above quantities.

Lemma 2.10. *Let* $\beta \in [0, \frac{1}{4}]$ *and assume that* $(x, s) \in \mathcal{N}_\infty(\beta)$. *Then,*

$$\xi_{ij} n\delta(u) \leq \Delta_{ij} \leq \zeta_{ij} n\delta(u), \quad \forall ij \in \{21, 22\}, \tag{2.4}$$

where
$\xi_{21} := 3(1 - \beta) + 2(1 - \beta)\ln(1 - \beta)$,
$\zeta_{21} := 3(1 + \beta) + 2(1 + \beta)\ln(1 + \beta)$,
$\xi_{22} := 2(1 - \beta) + 6(1 - \beta)\ln(1 - \beta) + 6(1 - \beta)\ln^2(1 - \beta)$,
$\zeta_{22} := 2(1 + \beta) + 6(1 + \beta)\ln(1 + \beta) + 6(1 + \beta)\ln^2(1 + \beta)$.

Proof. See Appendix C. □

Corollary 2.11. *For every* $(x, s) \in \mathcal{N}_\infty\left(\frac{1}{4}\right)$, *we have*

$$1.8 n\delta(u) \leq \Delta_{21} \leq \frac{9}{2} n\delta(u), \quad \textit{and} \quad \Delta_{22} < 5 n\delta(u).$$

Lemma 2.12. *Let* $\beta \in [0, \frac{1}{2}]$ *and assume that* $(x, s) \in \mathcal{N}_\infty^-(\beta)$. *Then,*

$$0 \leq \Delta_{12} \leq \zeta_{12} n\delta(u),$$

where $\zeta_{12} := 2(\ln(n) + 1)$. *Furthermore, the upper bound is tight within a constant factor for large* n.

Proof. The left-hand-side inequality obviously holds due to the nonnegativity of the vectors x, s, u and $\ln^2(Uu)$. For $\zeta_{12} = 2(\ln(n)+1)$, let us define $F_{12} := \zeta_{12} n\delta(u) - \varDelta_{12}$, then

$$\begin{aligned}\nabla F_{12}(u) &= 2(\ln(n)+1)e + 2(\ln(n)+1)\ln(u) - \mathrm{Diag}(\ln(u))\ln(u) - 2\ln(u),\\ \nabla^2 F_{12}(u) &= 2\ln(n)U^{-1} - 2\mathrm{Diag}(\ln(u))U^{-1}.\end{aligned}$$

We consider the constrained optimization problem

$$\mathrm{minimize}_{u\in\mathbb{R}^n} \quad F_{12}(u) \text{ subject to } \mathrm{e}^\top u - n = 0, u - \frac{1}{2}e \geq 0.$$

The Lagrangian has the form $\mathcal{L}_{12}(u,\lambda) = F_{12}(u) - \lambda_1(\mathrm{e}^\top u - n) - \lambda_2^\top(u - \frac{1}{2}e)$. Then, $\nabla^2 F_{12}$ is positive definite if $u < ne$. Since we know that $u \leq \frac{n+1}{2}e$ within $\mathcal{N}_\infty^-(\frac{1}{2})$, we conclude that F_{12} is strictly convex here. Moreover, for $u^* = e$, the Lagrange multipliers $\lambda_1^* = \zeta_{12}$ and $\lambda_2^* = 0$ satisfy the KKT conditions. Therefore, u^* is the global minimizer of the optimization problem. We notice that $F_{12}(u^*) = 0$ which implies the desired conclusion.

Let $u \in \mathbb{R}^n_{++}$ be a vector with $(n-1)$ entries equal to $1/2$ and one entry equal to $(n+1)/2$. Then, for large n, we have:

$$\frac{\varDelta_{12}}{n\delta(u)} = \frac{\frac{n-1}{2}\ln^2(1/2) + \frac{n+1}{2}\ln^2\left(\frac{n+1}{2}\right)}{\frac{n-1}{2}\ln(1/2) + \frac{n+1}{2}\ln\left(\frac{n+1}{2}\right)} \approx \ln\left(\frac{n+1}{2}\right) = \ln(n+1) - \ln(2).$$

Thus, the upper bound is tight within a constant factor, for large n. □

Lemma 2.13. *Let $x > 0$, $s > 0$. Then $\varDelta_{12} \geq n\delta^2$. Moreover, equality holds if and only if $Xs = \mu e$.*

Proof. Let $x > 0$, $s > 0$. Since $u_j > 0$, $\sqrt{u_j} > 0$ and $\sqrt{u_j}|\ln(u_j)| \geq 0$. Using Cauchy−Schwarz inequality, we have

$$\sum_{j=1}^n u_j \sum_{j=1}^n u_j \ln^2(u_j) \geq \left(\sum_{j=1}^n u_j|\ln(u_j)|\right)^2 \geq \left(\sum_{j=1}^n u_j\ln(u_j)\right)^2.$$

Then, the claimed inequality follows. Moreover, by utilizing $\sum_{j=1}^n u_j = n$, we have equality if and only if $u = e$ (we used Cauchy−Schwarz inequality), or equivalently $\frac{Xs}{\mu} = e$. □

Now, we have all the tools to state and analyze our algorithms.

3. Iteration complexity analysis for predictor-corrector algorithm

As stated in previous section, our search directions are the solutions of system (2.3), where $w(\eta) := -v + \eta\left[\delta v - V\ln\left(\frac{Vv}{\mu}\right)\right]$. Here, $\eta \in \mathbb{R}_+$ parameterizes the family of search directions. [35, 38] studied these search directions for special η from iteration complexity point of view. It is proved in [35] that, using $w(1)$ as the search direction (*i.e.*, $\eta = 1$), we can obtain the iteration complexity bound of $O\left(n\ln\left(\frac{1}{\epsilon}\right)\right)$ for $\mathcal{N}_E(3/2)$, for feasible start algorithms. These search directions have also been studied in [38], in the wide neighborhood, for the special case that $\eta = \frac{1}{\sigma}$ with $\sigma \in (0.5, 1)$ and $\sigma < \min\left\{1, \ln\left(\frac{1}{1-\beta}\right)\right\}$. It was shown in [38] that the underlying infeasible-start algorithm, utilizing a wide neighborhood, has iteration complexity of $O\left(n^2\ln\left(\frac{1}{\epsilon}\right)\right)$.

In this section, we show that the current best iteration complexity bound $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$ can be achieved if we use the entropy-based search direction in the predictor step of the standard predictor-corrector algorithm proposed by Mizuno *et al.* Ye [21], together with homogeneous self-dual embedding. Here is the algorithm:

Algorithm 3.1.
Input: $(A, x^{(0)}, s^{(0)}, b, c, \epsilon)$, where $(x^{(0)}, s^{(0)}) \in \mathcal{N}_2(\frac{1}{4})$, and $\epsilon > 0$ is the desired tolerance.
$(x, s) \leftarrow (x^{(0)}, s^{(0)})$,
while $x^\top s > \epsilon$,
  predictor step: solve (2.3) with $\eta = 1$ for $\bar{d}_x$ and $\bar{d}_s$.
    $x(\alpha) := x + \alpha D \bar{d}_x$,
    $s(\alpha) := s + \alpha D^{-1} \bar{d}_s$, where $D = X^{1/2} S^{-1/2}$.
    $\alpha^* := \max\{\alpha : (x(\alpha), s(\alpha)) \in \mathcal{N}_2(\frac{1}{2})\}$.
  Let $x \leftarrow x(\alpha^*)$, and $s \leftarrow s(\alpha^*)$
  corrector step: solve (2.3) for $\bar{d}_x$ and $\bar{d}_s$, where $w(\eta)$ is replaced by $-v + \mu V^{-1} e$,
  Let $x \leftarrow x + D\bar{d}_x$, $s \leftarrow s + D^{-1}\bar{d}_s$.
end {while}.

The $O\left(\sqrt{n} \ln\left(\frac{1}{\epsilon}\right)\right)$ iteration complexity bound is the conclusion of a series of lemmas.

Lemma 3.1. *For every point $(x, s) \in \mathcal{N}_2(\frac{1}{4})$, the following condition on α guarantees that $(x(\alpha), s(\alpha)) \in \mathcal{N}_2(\frac{1}{2})$:*

$$d_4\alpha^4 + d_3\alpha^3 + d_2\alpha^2 + d_1\alpha + d_0 \leq 0,$$

where

$$\begin{aligned}
d_0 &:= -3\mu^2 \leq 0, \\
d_1 &:= 32\left(\delta \sum_{j=1}^n x_j^2 s_j^2 - n\delta\mu^2 - \varDelta_{21}\mu^2 + n\delta\mu^2\right) + 6\mu^2 = 32\left(\delta \sum_{j=1}^n x_j^2 s_j^2 - \varDelta_{21}\mu^2\right) + 6\mu^2, \\
d_2 &:= 16\left(\delta^2 \sum_{j=1}^n (x_j s_j)^2 + \varDelta_{22}\mu^2 - 2\delta\varDelta_{21}\mu^2 + 2C\right) - d_1 + 3\mu^2, \\
d_3 &:= 32(\delta - 1)C - 32B, \\
d_4 &:= 16\sum_{j=1}^n (w_p)_j^2 (w_q)_j^2, \\
B &:= \sum_{j=1}^n x_j s_j \ln(u_j)\,(w_p)_j (w_q)_j, \\
C &:= \sum_{j=1}^n x_j s_j (w_p)_j (w_q)_j.
\end{aligned}$$

Proof. See Appendix C. □

Lemma 3.2. *For every point $(x, s) \in \mathcal{N}_2(\frac{1}{4})$, we have the following bounds on d_1, d_2, d_3 and d_4 defined in Lemma 3.1.*

$$d_1 \leq 10\mu^2, \quad d_2 \leq 34n\mu^2, \quad d_3 \leq 64n^{\frac{3}{2}}\mu^2, \quad d_4 \leq 5n^2\mu^2.$$

Proof. See Appendix C. □

We state the following well-known lemma without proof.

Lemma 3.3 [21]**.** *For every point $(x, s) \in \mathcal{N}_2(\frac{1}{2})$, the corrector step of Algorithm 3.1 returns a point in the neighborhood $\mathcal{N}_2(\frac{1}{4})$.*

Now, we can prove the iteration complexity bound for Algorithm 3.1.

Theorem 3.4. *Algorithm 3.1 gives an ϵ-solution in $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$ iterations.*

Proof. By Lemmas 3.1 and 3.2, in the predictor step, it is sufficient for α to satisfy

$$5n^2\alpha^4 + 64n^{\frac{3}{2}}\alpha^3 + 34n\alpha^2 + 10\alpha \leq 3. \tag{3.1}$$

It is easy to check that $\alpha = \frac{1}{50\sqrt{n}}$ satisfies this inequality. Lemma 3.3 shows that we have a point $(x, s) \in \mathcal{N}_2(\frac{1}{4})$ at the beginning of each predictor step and the algorithm is consistent. Since $x(\alpha)^\top s(\alpha) = (1-\alpha)x^\top s$ by part (b) of Lemma 3.1 of [35], we deduce that the algorithm will reach an ϵ-solution in $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$ iterations. □

4. Algorithm for the wide neighborhoods

In the rest of the paper, we study the behaviour of the entropy-based search directions in a wide neighborhood. As mentioned before, for each η, our search direction is derived from the solution of system (2.3), where $w(\eta) := -v + \eta\left[\delta v - V\ln\left(\frac{Vv}{\mu}\right)\right]$. These search directions have been studied in [38] in the wide neighborhood for the special case that $\eta = \frac{1}{\sigma}$ with $\sigma \in (0.5, 1)$ and $\sigma < \min\{1, \ln(\frac{1}{1-\beta})\}$. In this paper, we study these search directions for a wider range of η. We prove some results on iteration complexity bounds in this section. However, in the rest of the paper, we mainly focus on the practical performance of our search directions in the wide neighborhood.

The algorithm in a wide neighborhood (with a value of $\eta \geq 0$ fixed by the user) is:

Algorithm 4.1.
Input $(A, x^{(0)}, s^{(0)}, b, c, \epsilon, \eta)$, $\epsilon > 0$ is the desired tolerance.
$(x, s) \leftarrow (x^{(0)}, s^{(0)})$,
while $x^\top s > \epsilon$
  solve (2.3) for $\bar{d}_x$ and $\bar{d}_s$,
    $x(\alpha) := x + \alpha D\bar{d}_x$,
    $s(\alpha) := s + \alpha D^{-1}\bar{d}_s$, where $D = X^{1/2}S^{-1/2}$.
    $\alpha^* := \max\{\alpha : (x(\alpha), s(\alpha)) \in \mathcal{N}_\infty^-(\beta)\}$.
  Let $x \leftarrow x(\alpha^*)$; $s \leftarrow s(\alpha^*)$
end {while}.

Lemma 4.1. *In Algorithm 4.1, for every choice of $\eta \in \mathbb{R}_+$, we have $x(\alpha)^\top s(\alpha) = (1-\alpha)\, n\mu$.*

Proof. We proceed as in the proof of Lemma 3.1 of [35], part (b):

$$x(\alpha)^\top s(\alpha) = x^\top s + \alpha v^\top(\bar{d}_x + \bar{d}_s) = x^\top s + \alpha v^\top w(\eta) = (1-\alpha)x^\top s.$$

For the last equation, we used the facts that $v^\top v = x^\top s$, and v and $\delta v - V\ln\left(\frac{Vv}{\mu}\right)$ are orthogonal. □

This lemma shows that the reduction in the duality gap is independent of η and is exactly the same as in the primal-dual affine scaling algorithm. So, Lemma 4.1 includes part (b) of Lemma 3.1 of [35] and part (c) of Theorem 3.2 of [22] as special cases. We show later that by performing a plane search, we can find an η that gives the largest value of α in the algorithm (and hence the largest possible reduction in duality gap, per iteration).

Lemma 4.2. *Let $x > 0$, $s > 0$. Then, for $\eta \geq 0$, we have $\|w(\eta)\|_2^2 = n\mu + \eta^2\mu\Delta_{12} - n\mu\eta^2\delta^2$.*

Proof. Let $x > 0$, $s > 0$, and $\eta \geq 0$. Then,

$$\begin{aligned}
\|w(\eta)\|_2^2 &= \sum_{j=1}^{n} v_j^2 \left(\delta\eta - 1 - \eta \ln\left(\frac{x_j s_j}{\mu}\right)\right)^2 \\
&= \sum_{j=1}^{n} x_j s_j \left(\delta^2\eta^2 + 1 + \eta^2 \ln^2\left(\frac{x_j s_j}{\mu}\right) + 2\eta \ln\left(\frac{x_j s_j}{\mu}\right) - 2\delta\eta - 2\delta\eta^2 \ln\left(\frac{x_j s_j}{\mu}\right)\right) \\
&= n\mu\delta^2\eta^2 + n\mu + \eta^2\mu\Delta_{12} + 2n\eta\mu\delta - 2\delta\eta n\mu - 2n\mu\delta\eta^2\delta \\
&= n\mu + \eta^2\mu\Delta_{12} - n\mu\eta^2\delta^2. \qquad \square
\end{aligned}$$

Theorem 4.3. *If we apply Algorithm 4.1 with $\mathcal{N}_\infty^-(\frac{1}{2})$, then the algorithm converges to an ϵ-solution in at most $O\left(n\ln(n)\ln\left(\frac{1}{\epsilon}\right)\right)$ iterations for every nonnegative $\eta = O(1)$.*

Proof. See Appendix C. □

In the above algorithm, the value of η is constant for all values of j. In the following, we show that if η is allowed to take one of two constant values for each j (one of the values being zero), we get a better iteration complexity bound. For each $j \in \{1, \ldots, n\}$, let us define

$$[w(\eta)]_j := \begin{cases} -v_j, & \text{if } u_j > \frac{3}{4}, \\ -v_j + \eta\left[\delta v_j - v_j \ln\left(\frac{v_j^2}{\mu}\right)\right], & \text{if } \frac{1}{2} \leq u_j \leq \frac{3}{4}, \end{cases} \tag{4.1}$$

where $\eta := \frac{1}{(\delta + \ln(2))}$. Now we have the following theorem:

Theorem 4.4. *If we apply the Algorithm 4.1 with $w(\eta)$ defined in (4.1) to $\mathcal{N}_\infty^-(\frac{1}{2})$, the algorithm converges to an ϵ-solution in at most $O(n\ln\left(\frac{1}{\epsilon}\right))$ iterations.*

Proof. See Appendix C. □

5. Plane search algorithms

In the previous section, we showed how to fix two parameters α and η to achieve iteration complexity bounds. However, in practice we may consider performing a plane search to choose the best α and η in each iteration. Here, our goal is to choose a direction in our family of directions that gives the most reduction in the duality gap. As before, we have $w(\eta) = -v + \eta\left[\delta v - V\ln(\frac{Vv}{\mu})\right]$. For simplicity, in this section we drop parameter η and write $w = w(\eta)$, so $w_p = P_{AD}w$, and $w_q = w - w_p$, where P_{AD} is the projection operator onto the null space of AD. Our goal is to solve the following optimization problem.

$$\begin{aligned}
&\text{maximize } \alpha \\
&\text{subject to } 0 < \alpha < 1, \\
&\qquad \eta \geq 0, \\
&\qquad \frac{(w_p)_j (w_q)_j}{\mu}\alpha^2 + \alpha\left(u_j\delta\eta - u_j\ln(u_j)\eta - u_j + \frac{1}{2}\right) + \left(u_j - \frac{1}{2}\right) \geq 0, \quad \forall j \in \{1, \ldots, n\}.
\end{aligned} \tag{5.1}$$

In the above optimization problem, the objective function is linear and the main constraints are quadratic. Let us define

$$t_p := P_{AD}\left(\delta v - V\ln\left(\frac{Vv}{\mu}\right)\right), \quad t_q := \left(\delta v - V\ln\left(\frac{Vv}{\mu}\right)\right) - t_p, \quad v_p := P_{AD}(-v), \quad v_q := -v - v_p.$$

By these definitions, the quadratic inequalities in formulation (5.1) become

$$a_j\eta^2\alpha^2 + b_j\eta\alpha + c_j\eta\alpha^2 + d_j(1-\alpha) + e_j\alpha^2 \geq 0, \quad \text{where}$$
$$a_j := \frac{(t_p)_j(t_q)_j}{\mu},\ b_j := u_j\delta - u_j\ln(u_j),\ c_j := \frac{(v_p)_j(t_p)_j + (v_q)_j(t_p)_j}{\mu},$$
$$e_j := \frac{(v_p)_j(v_q)_j}{\mu},\ d_j := u_j - \frac{1}{2}\cdot$$

In this section, we propose two algorithms, an exact one and a heuristic one, to solve the two-variable optimization problem (5.1).

5.1. Exact plane search algorithm

We define a new variable $z := \alpha\eta$. Then, the quadratic form can be written as:

$$g_j(z,\alpha) := a_jz^2 + b_jz + c_jz\alpha + d_j(1-\alpha) + e_j\alpha^2, \quad \forall j \in \{1,\ldots,n\}.$$

We are optimizing in the plane of α and z, actually working in the one-sided strip in $\mathbb{R}^2$, defined by $0 \leq \alpha \leq 1$ and $z \geq 0$. The following proposition establishes that it suffices to check $O(n^2)$ points to find an optimal solution:

Proposition 5.1. *Let* (α^*, η^*) *be an optimal solution of* (5.1)*. Then, one of the following is true:*

(1) $\alpha^* = 1$;
(2) *there exists* $z^* \geq 0$ *such that* (z^*, α^*) *is a solution of system* $(g_j(z,\alpha), g_i(z,\alpha)) = (0,0)$ *for some pair* $i,j \in \{1,\ldots,n\}$;
(3) α^* *is a solution of* $\Delta_j(\alpha) := (b_j + \alpha c_j)^2 - 4a_j(d_j(1-\alpha) + e_j\alpha^2) = 0$, $j \in \{1,\ldots,n\}$, *where* $\Delta_j(\alpha)$ *is the discriminant of* $g_j(z,\alpha)$ *with respect to* z.

Proof. Assume that $(\alpha^* \neq 1, \eta^*)$ is a solution to (5.1), and $z^* := \alpha^*\eta^*$. Therefore, we have $g_j(z^*,\alpha^*) \geq 0$, $\forall j \in \{1,\ldots,n\}$. By continuity, we must have $g_j(z^*,\alpha^*) = 0$ for at least one j, and because z^* is real, we have $\Delta_j(\alpha^*) \geq 0$. If $\Delta_j(\alpha^*) = 0$, then condition (3) is satisfied; otherwise, by continuity, we can increase α so that Δ_j remains positive. In this case, if there does not exist another $i \in \{1,\ldots,n\}$ such that $g_i(z^*,\alpha^*) = 0$, continuity gives us another point $(\bar{\alpha}, \bar{\eta})$ that is feasible to (5.1) and $\bar{\alpha} > \alpha^*$, which is a contradiction. Hence, condition (2) must hold. □

The above proposition tells us that to find a solution for (5.1), it suffices to check $O(n^2)$ values for α. For calculating each of these values, we find the roots of a quartic equation.

5.2. Heuristic plane search algorithm

The idea of the heuristic algorithm is that we start with $\alpha = 1$ and see if there exists η such that (η, α) is feasible for (5.1). If not, we keep reducing α and repeat this process. We can reduce α by a small amount (for example 0.01) if α is close to 1 (for example $\alpha \geq 0.95$), and by a larger amount (for example 0.05) otherwise. This approach tries to favor the larger α values over the smaller ones.

The difficult part is checking if there exists η for the current α in the algorithm. To do that, we need to verify if there exists a positive η which satisfies the n inequalities in the third group of constraints of (5.1). Each constraint is a quadratic form in η and can induce a feasible interval for η. If the intersection of all the intervals corresponding to these n inequality constraints is not empty, we then find the η corresponding to a step length α. We use the following procedure to determine the feasible interval of η for a given step length α.

Assume that we fix α. For each quadratic constraint of (5.1), we can solve for η and find the feasible interval. One form is the union of two open intervals, *i.e.*, $(-\infty, r_1(j)]$ and $[r_2(j), \infty)$; denote the indexes in this class as K_1. Another is the convex interval $[r_3(j), r_4(j)]$; denote the indexes in this class by K_2. It is easy to find the intersection of the convex intervals:

$$[t_1, t_2] := [\max_{j \in K_2} r_3(j), \min_{j \in K_2} r_4(j)].$$

Now we have to intersect $[t_1, t_2]$ with the intervals in class K_1. First, we handle the intervals $[t_1, t_2]$ that intersect only one of $(-\infty, r_1(j)]$ and $[r_2(j), \infty)$; in that case we can update $[t_1, t_2] \leftarrow [t_1, r_1(j)]$ or $[t_1, t_2] \leftarrow [r_2(j), t_2]$ for each of these intervals. At the end of this step, we can assume that for the rest of the intervals in K_2 (we denote them by $\bar{K}_2$), $[t_1, t_2]$ intersects both $(-\infty, r_1(j)]$ and $[r_2(j), \infty)$. Then, we can define two intervals:

$$[t_1, t_3 := \min_{j \in \bar{K}_2} (r_1(j))], \quad [t_4 := \max_{j \in \bar{K}_2} (r_2(j)), t_2].$$

If one of these intervals is non-empty, then there exists η such that (η, α) is feasible for (5.1), and we return α. For a more detailed introduction to this heuristic see [19].

To evaluate the performance of our heuristic algorithm, note that the set of feasible points (α, η) of (5.1) in $\mathbb{R}^2$ is not necessarily a connected region. We can think of it as the union of many connected components. In our heuristic algorithm, we check a few discrete values of $\alpha = \bar{\alpha}$. However, for each value we check, we can precisely decide if there exists a feasible η for that value of α. If one of the lines $\alpha = \bar{\alpha}$ intersects a component of feasible region that contains a point with maximum α, then our heuristic algorithm returns an α that is close the optimal value. However, if none of the lines $\alpha = \bar{\alpha}$ that we check for large values of $\bar{\alpha}$ intersects the right component, the heuristic algorithm may return a very bad estimate of the optimal value. In the next section, we observe that (see Figs. 5–8) our heuristic algorithm in the worst-case may return values for α very close to zero while the optimal value is close to 1.

6. Computational experiments with the entropic search direction family

We performed some computational experiments using the software MATLAB R2014a, on a 48-core AMD Opteron 6176 machine with 256GB of memory. The test LP problems are well-known among those in the problem set of NETLIB [27].

We implemented Algorithm 4.1 for a fixed value of η and then ran it for each fixed $\eta \in \{1, 2, 3, 4\}$. We also implemented Algorithm 4.1 with η being calculated using the exact and heuristic plane search algorithms. $\beta = 1/2$ was set for the algorithm, therefore our results are for the wide neighborhood $\mathcal{N}_\infty^-(1/2)$. We used homogeneous self-dual embedding for the LP problems as shown in Appendix B. The initial feasible solution is $y^{(0)} := 0$, $x^{(0)} := e$, $s^{(0)} := e$, $\theta := 1$, $t := 1$ and $\kappa := 1$. In the statements of Algorithms 3.1 and 4.1, we used the stopping criterion $x^T s \leq \epsilon$, which is an abstract criterion assuming exact arithmetic computation. In practice, we may encounter numerical inaccuracies and we need to take that into account for our stopping criterion. We used the stopping criterion proposed and studied in [15], which is very closely related to the stopping criterion in SeDuMi [33]. Let us define $(\bar{x}, \bar{y}, \bar{s}) := (\frac{x}{\tau}, \frac{y}{\tau}, \frac{s}{\tau})$, and their residuals:

$$\begin{aligned} r_p &:= b - A\bar{x}, \\ r_d &:= A^\top \bar{y} + \bar{z} - c, \\ r_g &:= c^\top \bar{x} - b^\top \bar{y}. \end{aligned}$$

The following stopping criterion for general convex optimization problems using homogeneous self-dual embedding was proposed in [15]:

$$2\frac{\|r_p\|_\infty}{1 + \|b\|_\infty} + 2\frac{\|r_d\|_\infty}{1 + \|c\|_\infty} + \frac{\max\{0, r_g\}}{\max\{|c^\top \bar{x}|, |b^\top \bar{y}|, 1\}} \leq r_{\max}.$$

In our algorithm, we used the above stopping criterion for $r_{\max} := 10^{-9}$.

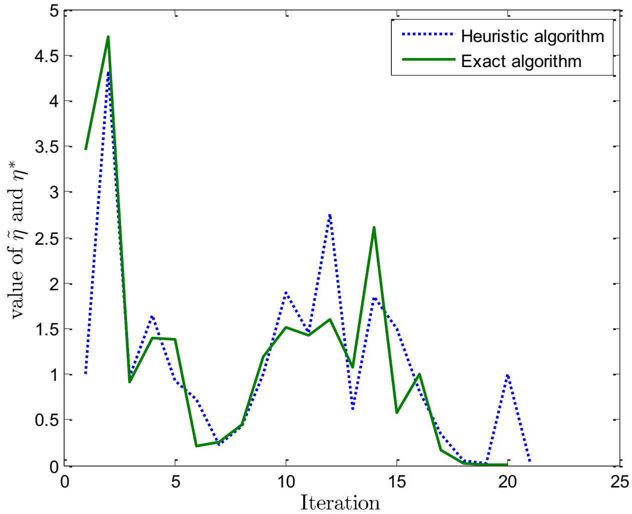

FIGURE 1. Values of $\tilde{\eta}$ (for the heuristic algorithm) and η^* (for the exact algorithm) in each iteration for problem *beaconfd*.

Table 1 shows the number of iterations for each problem. The first four columns show the number of iterations of Algorithm 4.1 with a fixed value of $\eta \in \{1, 2, 3, 4\}$. Let use define $\tilde{\eta}$ and η^* as the η found at each iteration of the plane search algorithm using the heuristic and exact plane search algorithms, respectively. The fifth and sixth columns of the table are the number of iterations when we perform a plane search, using the heuristic plane search and exact plane search algorithms, respectively. The problems in the table are sorted based on the value of $\eta \in \{1, \dots, 4\}$ that gives the smallest number of iterations. For each η, the problems are sorted alphabetically.

As we mentioned above, our family of search directions is a common generalization of the search direction in [35] that uses $\eta = 1$ and the search directions in [28, 38] that use $\eta = \frac{1}{\sigma}$ with $\sigma \in (0.5, 1)$ and $\sigma < \min\{1, \ln(\frac{1}{1-\beta})\}$, so $1 \leq \eta \leq 2$. As we observe from Table 1, our generalization to consider using larger values of η is justified. Among the problems solved and among the fixed values for $\eta \in \{1, 2, 3, 4\}$, $\eta = 1$ had the smallest iteration count for 15 problems, $\eta = 2$ won for 24 problems, $\eta = 3$ won for 13 problems, and $\eta = 4$ had the smallest iteration count for 18 problems (ties counted as wins for both winning η's). Table 1 also shows that using plane search algorithms can be crucial in reducing the number of iterations in addition to making the behaviour of the underlying algorithms more robust; as (1) for most of the problems, there is a large gap between the number of iterations of the plane search and the best constant η algorithms, and (2) we do not know which η is the best one before solving the problem.

The exact plane search algorithm gives a lower bound for our heuristic plane search algorithm. As we observe from Table 1, for most of the problems, exact and heuristic plane search algorithms have similar performances in terms of the number of iterations. In Figures 1–4, we plot the value of η at each iteration for four of the problems of NETLIB, for both exact and heuristic plane search algorithms. For *beaconfd* and *capri* the performances are close and for *degen2* and *ship08s* there is a large gap. An interesting point is that the plane search algorithms sometimes lead to values of η as large as 10 or 20 as can be seen in Figures 3 and 4.

Figures 5–8 provide a more reasonable comparison between the exact and heuristic plane search algorithms for problems *degen2* and *ship08s*. In Figure 5 (for *degen2*) and Figure 7 (for *ship08s*), we plot the values of η and α for the heuristic algorithm, as well as the corresponding values that would have been computed by the exact algorithm at each iteration (for the same current iterates $(x^{(k)}, s^{(k)})$). In Figure 6 (for *degen2*) and Figure 8

TABLE 1. The number of iterations of Algorithm 4.1.

NETLIB-Name	Dimensions	Nonzeros	$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\tilde{\eta}$	η^*
afiro	28*32	88	31	37	45	55	19	18
beaconfd	174*262	3170	31	37	46	56	22	21
blend	75*83	522	35	37	42	48	23	19
grow7	141*301	2633	47	47	55	65	36	31
grow15	301*645	5665	42	47	54	66	37	32
sc105	106*103	281	28	36	43	54	23	19
sc205	206*203	552	28	37	41	53	25	21
sc50a	51*48	131	28	35	43	51	21	18
sc50b	51*48	119	26	33	43	50	19	16
scagr7	130*140	533	34	39	47	55	28	25
scsd1	78*760	3148	31	35	43	52	26	18
scsd8	398*2750	11334	32	34	42	48	25	19
share2b	97*79	730	35	40	45	54	25	23
adlittle	57*97	465	40	40	49	56	27	24
kb2	44*41	291	43	43	49	58	32	30
agg	489*163	2541	58	50	58	68	46	39
agg2	517*302	4515	61	51	56	66	40	35
agg3	517*302	4531	70	55	61	68	42	37
boeing2	167*143	1339	72	53	54	62	39	37
brandy	221*249	21506	72	56	56	57	43	36
capri	272*353	1786	60	49	53	57	39	37
degen2	445*534	4449	41	38	42	48	37	24
degen3	1504*1818	26230	40	38	40	45	30	26
fit1d	25*1026	14430	68	53	64	64	38	37
forplan	162*421	4916	108	77	79	119	60	50
ganges	1310*1681	70216	52	48	54	63	41	38
gfrd-pnc	617*1092	3467	50	47	53	63	35	30
grow22	441*946	8318	51	50	55	67	38	34
lotfi	154*308	1086	54	46	50	58	35	32
scagr25	472*500	2029	43	42	51	58	33	31
scsd6	148*1350	5666	35	36	44	51	25	22
sctap2	1091*1880	8124	56	48	49	52	34	23
ship04s	403*1458	5910	53	45	52	54	34	30
ship04l	403*2118	8450	53	49	56	58	35	29
stocfor1	118*111	474	51	48	55	61	30	26
wood1p	245*2594	70216	120	62	104	65	53	52
fit1p	628*1677	10894	63	54	54	59	37	35
bandm	305*472	2659	67	54	51	58	40	35
boeing1	351*384	3865	91	68	65	68	51	46
e226	224*282	2767	66	53	52	55	40	36
israel	175*142	2358	96	69	67	73	46	40
d6cube	404*6184	37704	76	58	55	56	39	32
modszk1	686*1622	3170	110	87	77	82	75	53
scfxm1	331*457	2612	123	81	73	74	53	42

TABLE 1. Continued.

NETLIB-Name	Dimensions	Nonzeros	$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\tilde{\eta}$	η^*
scrs8	491*1169	4029	143	105	95	100	50	44
sctap3	1481*2480	10734	64	53	53	57	37	25
ship08s	779*2387	9501	82	62	62	63	54	31
vtp-base	199*203	914	117	84	76	77	50	36
scfxm3	991*1371	7846	143	98	84	84	70	47
25fv45	822*1571	11127	160	111	91	79	77	57
bnl1	644*1175	6129	183	122	103	95	87	64
bnl2	2325*3489	16124	197	137	110	100	78	71
czprob	930*3523	14173	236	156	129	119	70	61
etamacro	401*688	2489	218	134	109	98	66	59
pilot4	411*1000	5145	150	107	91	88	87	77
pilot-we	723*2789	9218	234	164	139	125	119	118
perold	626 *1376	6026	190	123	101	95	94	93
scfxm2	661*914	5229	146	97	83	81	69	47
sctap1	301*480	2052	119	81	75	67	37	35
seba	516*1028	4874	170	120	103	94	78	50
share1b	118*225	1182	128	84	74	73	58	51
ship12l	1152*5437	21597	272	164	133	120	75	50
ship12s	1152*2763	10941	218	134	106	97	74	42
stocfor2	2158*2031	9492	129	95	83	82	63	56
standata	360*1075	3038	108	71	67	66	33	27
standmps	468*1075	3686	125	85	73	70	63	35

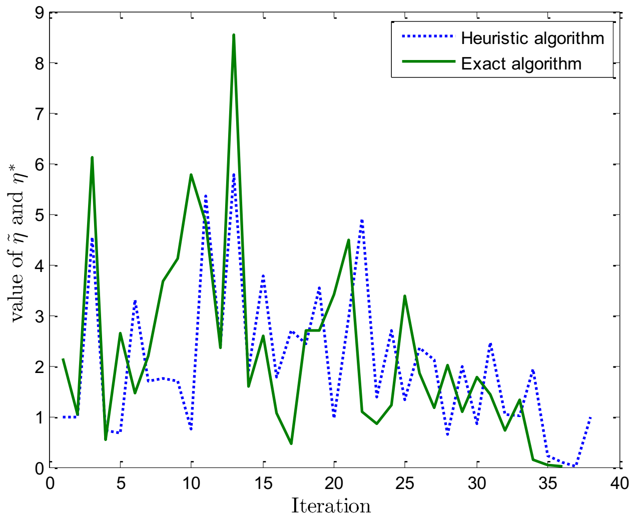

FIGURE 2. Values of $\tilde{\eta}$ (for the heuristic algorithm) and η^* (for the exact algorithm) in each iteration for problem *capri*.

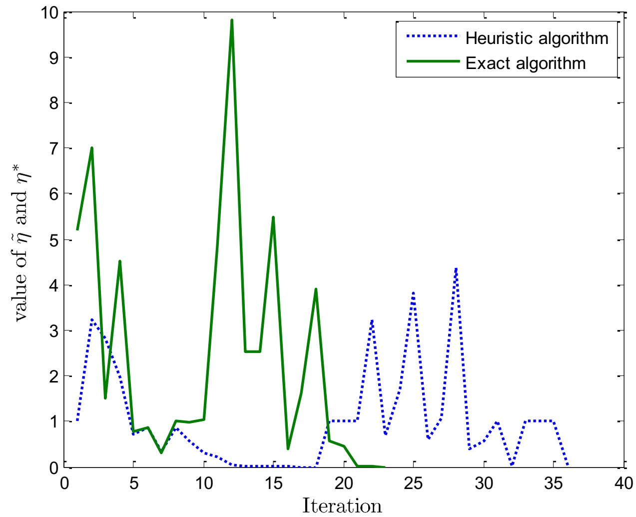

FIGURE 3. Values of $\tilde{\eta}$ (for the heuristic algorithm) and η^* (for the exact algorithm) in each iteration for problem *degen2*.

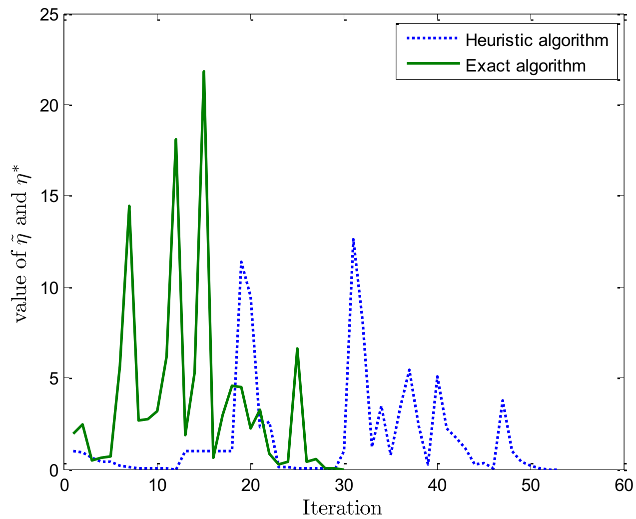

FIGURE 4. Values of $\tilde{\eta}$ (for the heuristic algorithm) and η^* (for the exact algorithm) in each iteration for problem *ship08s*.

(for *ship08s*), we plot the values of η and α for the exact algorithm, as well as the corresponding values that would have been given by the heuristic algorithm at each iteration. Note that in Figures 5–8, the comparison is iteration-wise. The plot in solid line is the main algorithm and the plot in dotted line is the value that would have been returned by the other algorithm using the iterates generated by the main algorithm. We observe from the figures that when the optimal value of α is close to 1 or 0, the heuristic algorithm cannot keep up with the exact algorithm.

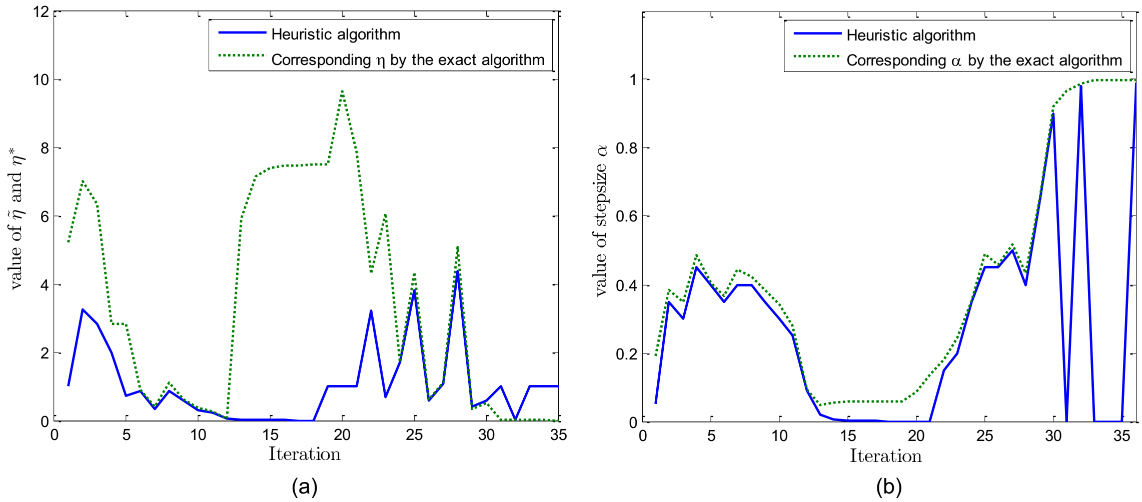

FIGURE 5. Values of (a) η (b) α for the heuristic algorithm, and the corresponding values calculated by the exact algorithm at each iteration of it, for problem *degen2*.

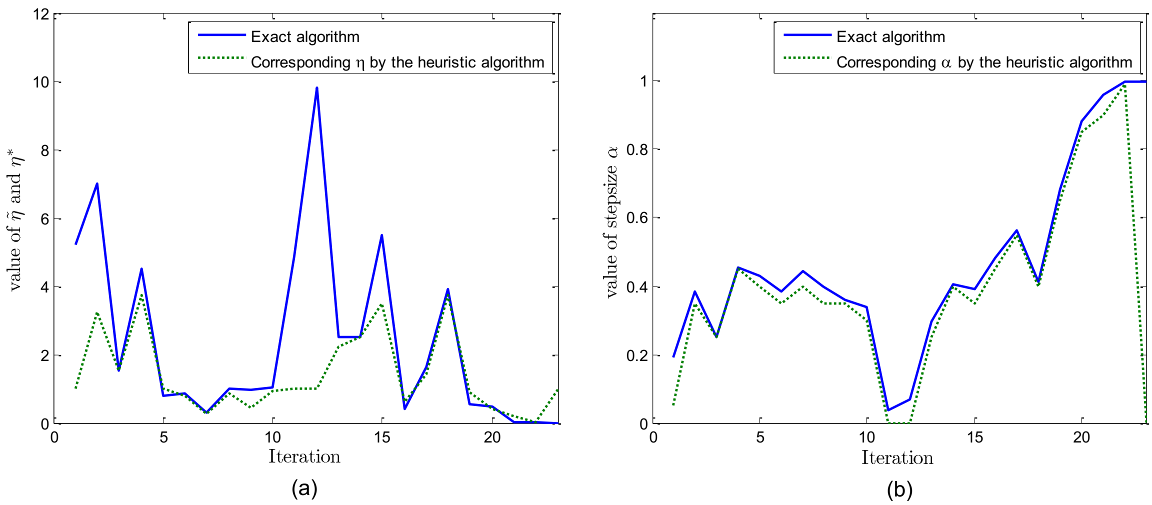

FIGURE 6. Values of (a) η (b) α for the exact algorithm, and the corresponding values calculated by the heuristic algorithm at each iteration of it, for problem *degen2*.

A conclusion of the above discussion is that utilization of plane search algorithms improves the number of iterations significantly. If the plane search algorithm is fast enough, then we can also improve the running time. Our heuristic plane search algorithm is much faster than the exact one. For the exact plane search algorithm, we solve $O(n^2)$ quartic equations, and in each iteration of the primal-dual algorithm, we perform $O(n^3)$ operations. Therefore, if we can speed up our exact plane search algorithm, this would have a potential impact on practical performance of algorithms in this paper as well as some other related algorithms. Note that our main focus in these preliminary computational experiments is on the number of iterations. To speed up the plane search algorithms, one may even use tools from computational geometry, analogous to those used for solving two-dimensional (or $O(1)$-dimensional) LP problems with n constraints in $O(n)$ time (see [10, 20], and the book [11]).

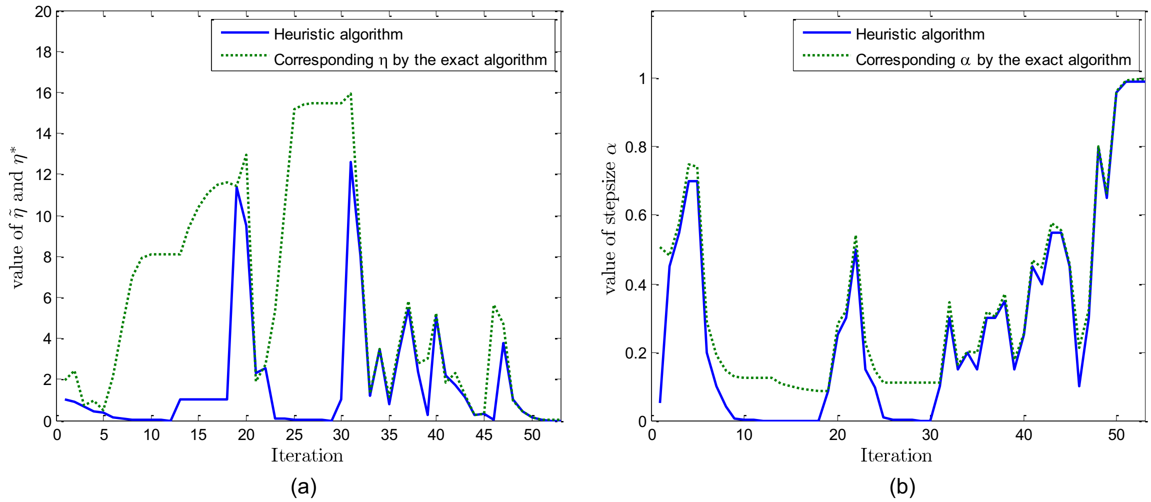

FIGURE 7. Values of (a) η (b) α for the heuristic algorithm, and the corresponding values calculated by the exact algorithm at each iteration of it, for problem *ship08s*.

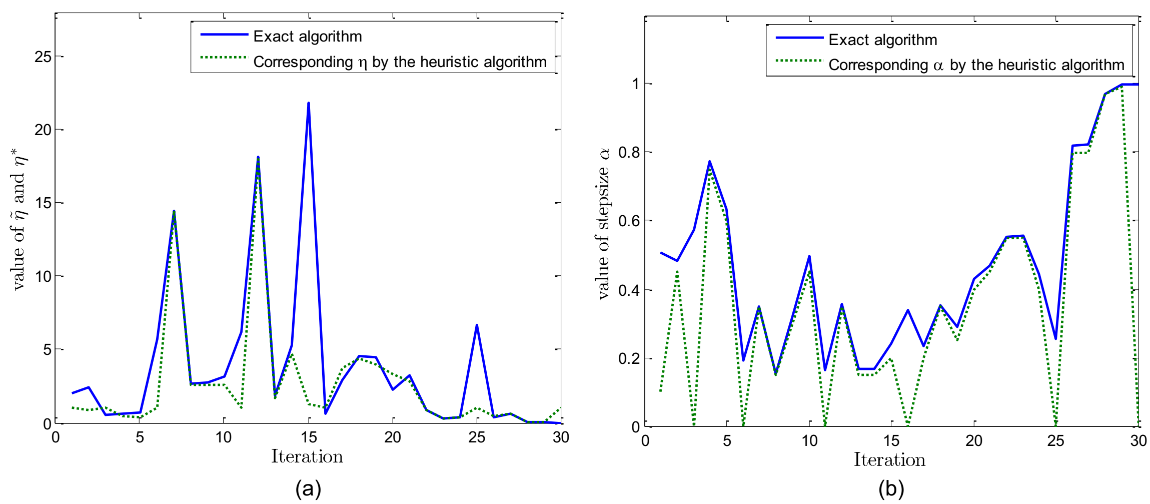

FIGURE 8. Values of (a) η (b) α for the exact algorithm, and the corresponding values calculated by the heuristic algorithm at each iteration of it, for problem *ship08s*.

7. CONCLUSION

In this paper, we introduced a family of search directions parameterized by η. We proved that if we use our search direction with $\eta = 1$ in the predictor step of standard predictor-corrector algorithm, we can achieve the current best iteration complexity bound. Then, we focused on the wide neighborhoods, and after the derivation of some theoretical results, we studied the practical performance of our family of search directions. To find the best search direction in our family, which gives the largest decrease in the duality gap, we proposed a heuristic plane search algorithm as well as an exact one. Our experimental results showed that using plane search algorithms improves the performance of the primal-dual algorithm significantly in terms of the number of iterations. Although our heuristic algorithm works efficiently, there is more room here to work on other heuristic plane search algorithms or improving the practical performance of the exact one, so that we also obtain a significant improvement in the overall running time of the primal-dual algorithm.

The idea of using a plane search in each iteration of a primal-dual algorithm has been used by many other researchers. For example, relatively recently, Ai and Zhang [1] defined a new wide neighborhood (which contains the conventional wide neighborhood for suitable choices of parameter values) and introduced a new search direction by decomposing the right-hand-side vector of (1.2) into positive and negative parts and performing a plane search to find the step size for each vector. By this approach, they obtained the current best iteration complexity bound for their wide neighborhood. Their approach together with ours inspires the following question: are there other efficient decompositions which in combination with a plane search, give good theoretical as well as computational performances in the wide neighborhoods of the central path? This is an interesting question left for future work.

Appendix A. Connection with Kernel functions

In this section, we introduce the Kernel function approach for interior-point methods [2,3,17] and discuss its connection with our approach. Let $\Psi(v) : \mathbb{R}^n_{++} \to \mathbb{R}$ be a strictly convex function such that $\Psi(v)$ is minimal at $v = e$ and $\Psi(e) = 0$. In the Kernel function approach we replace the last equation of (1.2) with

$$Sd_x + Xd_s = -\sqrt{\mu} V \nabla\Psi\left(\frac{v}{\sqrt{\mu}}\right), \tag{A.1}$$

where $v := X^{1/2}S^{1/2}e$ [2]. Note that by the definition of Ψ, $\nabla\Psi\left(\frac{v}{\sqrt{\mu}}\right) = 0$ if and only if (x, s) is on the central path. To simplify the matters, we assume that

$$\Psi(v) = \sum_{j=1}^{n} \psi(v_j),$$

where $\psi(t) : \mathbb{R}_{++} \to \mathbb{R}$ is an strictly convex function with unique minimizer at $t = 1$ and $\psi(1) = 0$. We call the univariate function $\psi(t)$ the Kernel function of $\Psi(v)$. It has been shown that the short update primal-dual path following algorithms using special Kernel functions obtain the current best iteration complexity bound [2].

Comparing (1.3) and (A.1), we observe that the two approaches are similar in the sense that the left-hand-side of the last equation in (1.2) is replaced by a nonlinear function of Xs. The question here is whether there exists a continuously differentiable strictly monotone function f for each Kernel function ψ or *vice versa* so that (1.2) and (A.1) give the same search direction. In other words, can we solve

$$-\sqrt{\mu}t\psi'\left(\frac{t}{\sqrt{\mu}}\right) = K\frac{f(\mu) - f(t^2)}{f'(t^2)}, \tag{A.2}$$

for f or ψ, for a constant scalar K? For $t = \sqrt{\mu}$, both sides of (A.2) are equal to zero, so the equation is consistent in that sense. $\psi(t)$ is a strictly convex function with minimum at $t = 1$, so $\psi'\left(\frac{t}{\sqrt{\mu}}\right) < 0$ for $t < \sqrt{\mu}$ and $\psi'(\frac{t}{\sqrt{\mu}}) > 0$ for $t > \sqrt{\mu}$. This makes both sides of (A.2) consistent for a strictly monotone function f. Hence, (A.2) may be solved for f or ψ, however the result depends on μ in general. Table A.1 shows five pairs of functions. Some of the Kernel functions in the table are from the set of functions studied in [2], and we solved (A.2) for the corresponding $f(x)$. In the last two ones, we picked $f(x) = \ln(x)$ and $f(x) = \sqrt{x}$ and derived the corresponding $\psi(t)$.

TABLE A.1. Some $\psi(t)$ and their corresponding $f(x)$ in view of (A.2).

$\psi(t)$	$f(x)$
$\frac{t^2-1}{2} - \ln(t)$	x
$\frac{1}{2}\left(t - \frac{1}{t}\right)^2$	x^2
$\frac{1}{2}(t^2-1) + \frac{t^{-2q+2}-1}{-2q+2}, \quad q > 1$	x^q
$\frac{1}{2}\left(t^2 + \frac{1}{t^2}\right) - 1$	$\frac{1}{x}$
$\frac{t^2-1}{2} + \frac{t^{1-q}-1}{q-1}, \quad q > 1$	$x^{\frac{-q+1}{2}}$
$(t-1)^2$	$\sqrt{x}$
$t^2\ln(t) - \frac{1}{2}t^2 + \frac{1}{2}$	$\ln(x)$

As an example, we see the derivation of $f(x)$ for the third $\psi(t)$: we have $\psi'(t) = t - t^{-q}$, then

$$-\sqrt{\mu}t\psi'\left(\frac{t}{\sqrt{\mu}}\right) = -\sqrt{\mu}t\left(\frac{t}{\sqrt{\mu}} - \frac{\mu^{-q/2}}{t^{-q}}\right) = -t^2 + \frac{\mu^{\frac{-q+1}{2}}}{t^{-q-1}}$$

$$= \frac{\mu^{\frac{-q+1}{2}} - t^{-q+1}}{t^{-q-1}} = 2\frac{f(\mu) - f(t^2)}{f'(t^2)}, \quad f(x) = x^{\frac{-q+1}{2}}.$$

For the fourth pair, the function $\psi(t) = t^2\ln(t) - \frac{1}{2}t^2 + \frac{1}{2}$ obtains its minimum at $t = 1$ with $\psi(1) = 0$, and is decreasing before $t = 1$ and increasing after that. The function is also convex around $t = 1$, but it is not convex on the whole range of $t > 0$.

As mentioned above, for each Kernel function $\psi(t)$, solving (A.2) for $f(x)$ may result in a function depending on μ. We can cover that by generalizing our method as follows. At each iteration, instead of applying $f(\cdot)$ to both sides of $Xs = \mu e$, we apply a function of μ, *i.e.* $f(\mu;\cdot)$. The rationale behind it is that we expect different behaviours from the algorithm when $\mu > 1$ and $\mu \ll 1$; *e.g.*, we expect quadratic or at least super-linear convergence when $\mu \ll 1$. Hence, it is reasonable to apply a function $f(\cdot)$ that depends on μ. We saw above that (A.2) gives a non-convex function $\psi(t)$ for $f(x) = \ln(x)$ and the Kernel function approach does not cover our approach. However, our generalized method contains the Kernel function approach and is strictly more general in that sense.

Consider (A.2) for $K = \sqrt{\mu}/2$ and assume, without loss of generality, that $f(\sqrt{\mu}) = 0$. Then, from (A.2), for $t \neq \sqrt{\mu}$ we have:

$$\frac{2tf'(t^2)}{f(t^2) - f(\mu)} = \left(\psi'\left(\frac{t}{\sqrt{\mu}}\right)\right)^{-1} \quad \Rightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}[\ln(|f(t^2) - f(\mu)|)] = \left(\psi'\left(\frac{t}{\sqrt{\mu}}\right)\right)^{-1}$$

$$\Rightarrow |f(t^2)| = \exp\left[\int\left(\psi'\left(\frac{t}{\sqrt{\mu}}\right)\right)^{-1}\mathrm{d}t\right], \tag{A.3}$$

where we have $f(t^2) < 0$ for $t < \sqrt{\mu}$ and $f(t^2) > 0$ for $t > \sqrt{\mu}$. As an example, consider the kernel function $\psi(t) := t - 1 + \frac{t^{1-q}-1}{q-1}$ (see [2]) for the special case of $q = 2$. Then we have $\psi'(t) = 1 - t^{-2}$. Substituting this in (A.3), we have:

$$\begin{aligned}|f(t^2)| &= \exp\left[\int \frac{t^2}{t^2-\mu}\mathrm{d}t\right] = \exp\left[\int 1+\frac{\mu}{t^2-\mu}\mathrm{d}t\right]\\ &= \exp\left[\int 1+\frac{\sqrt{\mu}}{2}\left(\frac{1}{t-\sqrt{\mu}}-\frac{1}{t+\sqrt{\mu}}\right)\mathrm{d}t\right]\\ &= \mathrm{e}^t\left(\frac{|t-\sqrt{\mu}|}{t+\sqrt{\mu}}\right)^{\frac{\sqrt{\mu}}{2}}. \end{aligned}\tag{A.4}$$

As can be seen, the concluded function $f(\cdot)$ is a function of μ.

Appendix B. Homogeneous self-dual embedding

In this section, we introduce the homogeneous self-dual embedding [37]. We can construct a homogeneous and self-dual artificial LP problem (HLP) related to (P) and (D) as follows: given any $x^{(0)} > 0$, $s^{(0)} > 0$, and $y^{(0)}$ free,

$$\text{minimize} \quad ((x^{(0)})^\top s^{(0)}+1)\theta$$

$$\text{subject to} \quad Ax - bt + \bar{b}\theta = 0 \tag{B.1a}$$

$$-A^\top y + ct - \bar{c}\theta \geq 0 \tag{B.1b}$$

$$b^\top y - c^\top x + \bar{z}\theta \geq 0 \tag{B.1c}$$

$$-\bar{b}^\top y + \bar{c}^\top x - \bar{z}t = -((x^{(0)})^\top s^{(0)}+1) \tag{B.1d}$$

$$y \text{ free}, x \geq 0, t \geq 0, \theta \text{ free}, \tag{B.1e}$$

where $\bar{b} := b - Ax^{(0)}$, $\bar{c} := c - A^\top y^{(0)} - s^{(0)}$, and $\bar{z} := c^\top x^{(0)} + 1 - b^\top y^{(0)}$.

The relationships (B.1a)$-$(B.1c), with $t = 1$ and $\theta = 0$, represent primal and dual feasibility (with $x \geq 0$) and reversed weak duality, so that all together they define the set of primal and dual optimal solutions. To achieve feasibility for $x = x^{(0)}$ and $(y, s) = (y^{(0)}, s^{(0)})$, the artificial variable θ is added with appropriate coefficients and constraint (B.1d) is added to achieve self duality. Denote by s the slack vector for the inequality constraint (B.1b) and by κ the slack scalar for the inequality constraint (B.1c). We can see that (HLP) is homogeneous and self-dual.

The following are the properties of the (HLP) model [37].

- The Dual of (HLP), denoted by(HLD), has the same form as (HLP), *i.e.*, (HLD) is simply (HLP) with (y, x, t, θ) being replaced by (y', x', t', θ'). Here y', x', t', θ' make up the dual multiplier vector for constraint (B.1a), (B.1b), (B.1c), and (B.1d), respectively.
- (HLP) has a strictly feasible point for every choice of $x^{(0)} > 0, s^{(0)} > 0$.
- (HLP) has an optimal solution and its optimal solution set is bounded.
- The optimal value of (HLP) is zero, and for every feasible point $(y, x, t, \theta, s, \kappa)$ we have:
$$((x^{(0)})^\top s^{(0)}+1)\theta = x^\top s + t\kappa.$$
- There is an optimal solution $(y^*, x^*, t^*, \theta^* = 0, s^*, \kappa^*)$, such that:
$$\begin{pmatrix} x^* + s^* \\ t^* + \kappa^* \end{pmatrix} > 0,$$
which we call a *strictly self-complementary solution*.

If we choose $y^{(0)} := 0$, $x^{(0)} := e$, and $s^{(0)} := e$, then (HLP) becomes:

$$\begin{aligned} &\text{minimize} \quad (n+1)\theta \\ &\text{subject to} \quad Ax - bt + \overline{b}\theta = 0 \\ &-A^\top y + ct - \overline{c}\theta \geq 0 \\ &b^\top y - c^\top x + \overline{z}\theta \geq 0 \\ &-\overline{b}^\top y + \overline{c}^\top x - \overline{z}t = -(n+1) \\ &x \geq 0, t \geq 0, \end{aligned}$$

where $\overline{b} := b - Ae$, $\quad \overline{c} := c - e$, and $\overline{z} := c^\top e + 1$.

If we look at the solution of (HLP), we can solve the initial (LP) by using the theorem below.

Theorem B.1. [37] *Let $(y^*, x^*, t^*, \theta^* = 0, s^*, \kappa^*)$ be a strictly-self-complementary solution for* (HLP). *Then:*

- (P) *has an optimal solution if and only if $t^* > 0$. In this case, (x^*/t^*) is an optimal solution for* (P) *and $(y^*/t^*, s^*/t^*)$ is an optimal solution for* (D)*;*
- *if $t^* = 0$, then $\kappa^* > 0$, which implies that $c^\top x^* - b^\top y^* < 0$, i.e., at least one of $c^\top x^*$ and $-b^\top y^*$ is strictly less than 0. If $c^\top x^* < 0$ then* (D) *is infeasible; if $-b^\top y^* < 0$ then* (P) *is infeasible; and if both $c^\top x^* < 0$ and $-b^\top y^* < 0$ then both* (P) *and* (D) *are infeasible.*

So, homogeneous and self-dual model can guarantee that we have a strictly feasible solution to start most interior-point algorithms, and a strictly-self-complementary solution of the homogeneous self-dual embedding immediately solves both of the problems (P) and (D). In this context, "solving an LP" means determining exactly which of the three possibilities (given by the Fundamental Theorem of LP) holds and providing a succinct certificate of the claim.

Appendix C. Proofs of some theorems, lemmas, and propositions.

Proof of Lemma 2.10.

Proof. Let $\beta \in [0, \frac{1}{4}]$, $(x, s) \in \mathcal{N}_\infty(\beta)$, and ξ_{ij} and ζ_{ij}, $ij \in \{21, 22\}$, as in the statement of the lemma. We define

$$\begin{aligned} f_{ij}(u) &:= \varDelta_{ij} - \xi_{ij} n\delta(u), \qquad F_{ij}(u) := \zeta_{ij} n\delta(u) - \varDelta_{ij}, \\ \varOmega &:= \left\{u \in \mathbb{R}^n : \mathrm{e}^\top u = n,\ (1-\beta)e \leq u \leq (1+\beta)e\right\}. \end{aligned} \tag{C.1}$$

Consider the following four optimization problems for $ij \in \{21, 22\}$:

$$\text{minimize}_{u\in\varOmega} \quad F_{ij}(u) \qquad\qquad \text{and} \qquad\qquad \text{minimize}_{u\in\varOmega} \quad f_{ij}(u).$$

To prove the lemma, it is sufficient to prove that the optimal objective values of these four problems are at least 0. We prove it for $\min_{u\in\varOmega} f_{21}(u)$ and the proofs for the rest of them are similar. We have

$$\nabla f_{21}(u) = -\xi_{21} e + u + (2U - \xi_{21} I)\ln(u), \quad \nabla^2 f_{21}(u) = 3I - \xi_{21} U^{-1} + 2\mathrm{Diag}(\ln(u)). \tag{C.2}$$

For $u > 0$, $2\ln(u_j) + 3 - \frac{\xi_{21}}{u_j}$ is an increasing function of u_j for every $j \in \{1, \ldots, n\}$, and by our definition of ξ_{21} we have

$$2\ln(1-\beta) + 3 - \frac{\xi_{21}}{1-\beta} = 0.$$

Hence, by (C.2), $\nabla^2 f_{21}(u)$ is positive semidefinite over Ω, which implies $f_{21}(u)$ is a convex function over Ω. Let us write the Lagrangian function of the optimization problem $\min_{u\in\Omega} f_{21}(u)$:

$$\mathcal{L}_{21}(u,\lambda_1,\lambda_2,\lambda_3)=f_{21}(u)-\lambda_1(n-\mathrm{e}^\top u)-\lambda_2^\top(u-(1-\beta)e)-\lambda_3^\top((1-\beta)e-u),$$

where $\lambda_1\in\mathbb{R}^n$, $\lambda_2\in\mathbb{R}^n_+$, $\lambda_3\in\mathbb{R}^n_+$. Let us define

$$u^*:=e,\quad \lambda_1^*:=-1,\quad \lambda_2^*:=\xi_{21}e,\quad \lambda_3^*:=0.$$

Then, we have $\nabla\mathcal{L}_{21}(u^*,\lambda_1^*,\lambda_2^*,\lambda_3^*)=0$ and $\nabla^2\mathcal{L}_{21}(u^*,\lambda_1^*,\lambda_2^*,\lambda_3^*)=\nabla^2 f_{21}(u^*)$ is positive definite. Therefore, by second order sufficient conditions for optimality, $u^*=e$ is an optimal solution of $\min_{u\in\Omega} f_{21}(u)$ with optimal objective value of 0. □

Proof of Lemma 3.1

Proof. For $(x,s)\in\mathcal{N}_2(\frac{1}{4})$, the following condition guarantees that $(x(\alpha),s(\alpha))\in\mathcal{N}_2(\frac{1}{2})$.

$$\sum_{j=1}^n\left[\frac{x_j(\alpha)s_j(\alpha)}{(1-\alpha)\mu}-1\right]^2\le\sum_{j=1}^n\left(\frac{x_js_j}{\mu}-1\right)^2+\frac{3}{16}\cdot \tag{C.3}$$

Solving (2.3), we have $(d_x)_j=\sqrt{x_j/s_j}(w_p)_j$ and $(d_s)_j=\sqrt{s_j/x_j}(w_q)_j$, for $j\in\{1,\dots,n\}$. Using these, we have

$$\begin{aligned}x_j(\alpha)s_j(\alpha)&=\left(x_j+\alpha\sqrt{\frac{x_j}{s_j}}(w_p)_j\right)\left(s_j+\alpha\sqrt{\frac{s_j}{x_j}}(w_q)_j\right)\\&=x_js_j+\alpha\sqrt{x_js_j}(w_j)+\alpha^2(w_p)_j(w_q)_j.\end{aligned}$$

Substituting this in the left-hand-side of (C.3) and expanding it, we get

$$\begin{aligned}\sum_{j=1}^n\left(\frac{x_j(\alpha)s_j(\alpha)}{(1-\alpha)\mu}-1\right)^2&=\sum_{j=1}^n\left(u_j-1+\frac{\alpha x_js_j}{(1-\alpha)\mu}(\delta-\ln(u_j)))+\frac{\alpha^2}{(1-\alpha)\mu}(w_p)_j(w_q)_j\right)^2\\&=\sum_{j=1}^n\Bigg[(u_j-1)^2+\frac{\alpha^2(u_j)^2}{(1-\alpha)^2}[\delta^2+\ln^2(u_j)-2\delta\ln(u_j)]\\&+\frac{\alpha^4}{(1-\alpha)^2\mu^2}(w_p)_j^2(w_q)_j^2+2(u_j-1)\left(\frac{\alpha u_j}{(1-\alpha)}(\delta-\ln(u_j))+\frac{\alpha^2}{(1-\alpha)\mu}(w_p)_j(w_q)_j\right)\\&+2\frac{\alpha^3x_js_j}{(1-\alpha)^2\mu^2}(\delta-\ln(u_j))(w_p)_j(w_q)_j\Bigg],\end{aligned}\tag{C.4}$$

where we used $u_j=(x_js_j)/\mu$. By canceling out $\sum_{j=1}^n(u_j-1)^2$ from both sides of (C.3) and multiplying both sides by $(1-\alpha)^2\mu^2$, we obtain the following equivalent inequality:

$$\begin{aligned}&\alpha^2(\delta^2\sum_{j=1}^n(x_js_j)^2+\varDelta_{22}\mu^2-2\delta\varDelta_{21}\mu^2)+\alpha^4\sum_{j=1}^n(w_p)_j^2(w_q)_j^2\\&+2\alpha(1-\alpha)\sum_{j=1}^n[x_j^2s_j^2\delta-\mu\delta x_js_j-x_j^2s_j^2\ln(u_j)+\mu x_js_j\ln(u_j)]\\&+2\alpha^2(1-\alpha)\sum_{j=1}^n x_js_j(w_p)_j(w_q)_j+2\alpha^3\sum_{j=1}^n\delta x_js_j(w_p)_j(w_q)_j\\&-2\alpha^3\sum_{j=1}^n x_js_j(w_p)_j(w_q)_j\ln(u_j)\le\frac{3(1-\alpha)^2\mu^2}{16}\cdot\end{aligned}\tag{C.5}$$

Note that for the coefficient of $\alpha^2(1-\alpha)$ we used $\sum_{j=1}^n (w_p)_j(w_q)_j = 0$. We show how to derive the first term in (C.5) and the rest of them can be done similarly. After multiplying both sides of (C.4) by $(1-\alpha)^2\mu^2$, for the second term in the summation, we have:

$$\begin{aligned}
&\sum_{j=1}^n \alpha^2(u_j)^2\mu^2[\delta^2 + \ln^2(u_j) - 2\delta\ln(u_j)] \\
&= \alpha^2\left(\delta^2\sum_{j=1}^n (u_j\mu)^2 + \mu^2\sum_{j=1}^n (u_j)^2\ln^2(u_j) - 2\delta\mu^2\sum_{j=1}^n (u_j)^2\ln(u_j)\right) \\
&= \alpha^2\left(\delta^2\sum_{j=1}^n (x_js_j)^2 + \mu^2\Delta_{22} - 2\delta\mu^2\Delta_{21}\right),
\end{aligned}$$

which is the first term in (C.5). After expansion of the inequality above, we obtain the following inequality for the predictor step of length α, with the coefficients given in the statement.

$$d_4\alpha^4 + d_3\alpha^3 + d_2\alpha^2 + d_1\alpha + d_0 \le 0,$$ □

Proof of Lemma 3.2

Proof. Let us define $\beta := 1/4$. For every $(x,s) \in \mathcal{N}_2\left(\frac{1}{4}\right)$ we have

$$\frac{3}{4} < u_j = \frac{x_js_j}{\mu} < \frac{5}{4} \quad \forall j \quad \Rightarrow \quad \sum_{j=1}^n (x_js_j)^2 \le \frac{25}{16}n\mu^2. \tag{C.6}$$

We also have $0 \le \delta \le 1$; more precisely, $\frac{1}{96n} = \frac{\beta^2}{6n} \le \delta \le \frac{\beta^2}{n} = \frac{1}{16n}$ on the boundary of $\mathcal{N}_2\left(\frac{1}{4}\right)$ (see Lem. 2.4). By Lemma 1 of [21], we know that

$$\begin{aligned}
&\text{(a) } \|W_pw_q\| \le \frac{\sqrt{2}}{4}\|w\|^2 \quad \equiv \quad \sqrt{\sum_{j=1}^n (w_p)_j^2(w_q)_j^2} \le \frac{\sqrt{2}}{4}\sum_{j=1}^n w_j^2, \\
&\text{(b) } |(w_p)_j(w_q)_j| \le \frac{\|w\|^2}{4}, \\
&\text{(c) } \|(W_pw_q)^+\|_\infty \le \frac{\|w\|_\infty^2}{4}, \quad \text{where } (W_pw_q)_j^+ = \max\{0,(W_pw_q)_j\} \text{ for every } j.
\end{aligned} \tag{C.7}$$

We have $w_j = -v_j + \delta v_j - v_j\ln\left(\frac{v_j^2}{\mu}\right)$, for which we get

$$|w_j^2| = v_j^2\left|-1 + \delta - \ln\left(\frac{v_j^2}{\mu}\right)\right|^2 \le \frac{25}{16}v_j^2 = \frac{25}{16}x_js_j. \tag{C.8}$$

This inequality is due to the facts that $\left|\ln\left(\frac{v_j^2}{\mu}\right)\right| \le \frac{1}{4}$ and $0 \le \delta \le \frac{1}{16n}$ within $\mathcal{N}_2(\frac{1}{4})$. We need to bound d_4, d_3, d_2, and d_1. By using Corollary 2.11 and the above results, we have

$$\begin{aligned}
d_1 &= 32\left(\delta\textstyle\sum_{j=1}^n x_j^2s_j^2 - \Delta_{21}\mu^2\right) + 6\mu^2 \\
&\le 32\left(\delta\textstyle\sum_{j=1}^n x_j^2s_j^2\right) + 6\mu^2, && \text{as } \Delta_{21}\mu^2 \ge 0, \\
&\le 32\delta\cdot\frac{25}{16}n\mu^2 + 6\mu^2, && \text{using (C.6)}, \\
&\le 10\mu^2, && \delta \le \frac{1}{16n},
\end{aligned}$$

and

$$d_4 = 16\sum_{j=1}^{n}(w_p)_j^2(w_q)_j^2 \le 2\left(\sum_{j=1}^{n} w_j^2\right)^2 \le \frac{625}{128}\left(\sum_{j=1}^{n} x_j s_j\right)^2 = \frac{625}{128}n^2\mu^2 \le 5n^2\mu^2.$$

For the first inequality, we used (C.7)-(a) and for the second inequality we used (C.8). Within $\mathcal{N}_2(\frac{1}{4})$, we have $\left|\ln(\frac{x_j s_j}{\mu})\right| \le 1$. Hence, by Cauchy−Schwarz inequality we have:

$$\begin{aligned}
|B| &= \left|\textstyle\sum_{j=1}^{n} x_j s_j \ln\left(\frac{x_j s_j}{\mu}\right)(w_p)_j(w_q)_j\right| \\
&\le \textstyle\sum_{j=1}^{n} x_j s_j \left|\ln\left(\frac{x_j s_j}{\mu}\right)(w_p)_j(w_q)_j\right| \\
&\le \textstyle\sum_{j=1}^{n} x_j s_j |(w_p)_j(w_q)_j|, && \text{using } \left|\ln(\tfrac{x_j s_j}{\mu})\right| \le 1, \\
&\le \sqrt{\textstyle\sum_{j=1}^{n}(w_p)_j^2(w_q)_j^2}\sqrt{\textstyle\sum_{j=1}^{n} x_j^2 s_j^2}, && \text{using Cauchy−Schwarz}, \\
&\le \tfrac{\sqrt{2}}{4}\textstyle\sum_{j=1}^{n} w_j^2\sqrt{\tfrac{25}{16}n\mu^2}, && \text{using (C.7)-(a) and (C.6)}, \\
&\le \tfrac{\sqrt{2}}{4}\cdot\tfrac{25}{16}n\mu\sqrt{\tfrac{25}{16}n\mu^2} \le n^{\frac{3}{2}}\mu^2, && \text{using (C.8)}.
\end{aligned}$$

Similarly,

$$|C| \le \sum_{j=1}^{n} x_j s_j |(w_p)_j(w_q)_j| \le n^{\frac{3}{2}}\mu^2.$$

Moreover,

$$\begin{aligned}
C &\le \textstyle\sum_{\{j:(w_p)_j(w_q)_j \ge 0\}} x_j s_j (w_p)_j(w_q)_j \\
&\le \frac{\|w\|_\infty^2}{4}\textstyle\sum_{j=1}^{n} x_j s_j, && \text{using (C.7)-(c)}, \\
&\le \frac{1}{4}\cdot\frac{25}{16}\cdot\left(\frac{5}{4}\mu\right)n\mu \le \frac{n\mu^2}{2}, && \text{using (C.8) and (C.6)}.
\end{aligned}$$

Since $|\delta - 1| \le 1$, we get:

$$d_3 = 32(\delta-1)C - 32B < 32|C| + 32|B| \le 64n^{\frac{3}{2}}\mu^2.$$

Using Lemma 2.10 and Corollary 2.11, for every $(x,s) \in \mathcal{N}_2(\frac{1}{4})$, we have $\varDelta_{22} \le \frac{5}{16}$ and $\varDelta_{21} \le \frac{9}{32}$. Thus, using the bound we found for C, we have

$$\begin{aligned}
d_2 &= 16\left(\delta^2\sum_{j=1}^{n}(x_j s_j)^2 + \varDelta_{22}\mu^2 - 2\delta\varDelta_{21}\mu^2 + 2C - 2\delta\sum_{j=1}^{n} x_j^2 s_j^2 + 2\varDelta_{21}\mu^2\right) + 3\mu^2 \\
&\le 16\left(\frac{25}{16}\delta^2 n\mu^2 + \frac{5}{16}\mu^2 + 0 + n\mu^2 + 0 + \frac{9}{16}\mu^2\right) + 3\mu^2, \quad \text{using (C.6) and the bound for } C, \\
&\le 34n\mu^2, \quad \text{using } \delta \le \frac{1}{16n} \text{ and } n \ge 1.
\end{aligned}$$

□

Proof of Theorem 4.3.

Proof. For each $j \in \{1, \dots, n\}$, let $u_j := \frac{x_j s_j}{\mu}$. Let us consider the search direction w in the wide neighbourhood $\mathcal{N}_\infty^-(\frac{1}{2})$. For the next iteration, to stay in the same neighbourhood, the step length α should satisfy the following condition for every $j \in \{1, 2, \dots, n\}$.

$$u_j + \frac{\alpha}{1-\alpha} u_j(\delta\eta - \eta \ln(u_j)) + \frac{\alpha^2}{1-\alpha} \frac{(w_p)_j (w_q)_j}{\mu} \geq 1/2. \tag{C.9}$$

Since all of our discussion is within $\mathcal{N}_\infty^-(\frac{1}{2})$, we know that: $\frac{x_j s_j}{\mu} \geq \frac{1}{2}$, $j \in \{1, 2, \dots, n\}$. We also deduce that $\frac{x_j s_j}{\mu} \leq \frac{n+1}{2}$ from the fact $\sum_{j=1}^n x_j s_j = n\mu$. So,

$$0 \leq \varDelta_{12} \leq \max_j \left\{ \ln^2 \left(\frac{x_j s_j}{\mu} \right) \right\} \sum_{j=1}^n \frac{x_j s_j}{\mu} \leq n \ln^2 \left(\frac{n+1}{2} \right), \quad 0 \leq \delta \leq \ln \left(\frac{n+1}{2} \right).$$

By using (C.7)-(b) we have $|(w_p)_j (w_q)_j| \leq \|w(\eta)\|^2/4$. Using this and Lemma 4.2, a sufficient condition for (C.9) to hold is the following inequality:

$$u_j + \frac{\alpha}{1-\alpha} u_j(\delta\eta - \eta \ln(u_j)) - \frac{\alpha^2}{1-\alpha} \frac{n\mu + \eta^2 \mu \varDelta_{12} - n\mu\delta^2\eta^2}{4\mu} \geq 1/2.$$

By multiplying both sides by $4(1-\alpha)$ and rearranging, we get

$$-\alpha^2(n + \eta^2 \varDelta_{12} - n\delta^2\eta^2) + 4\alpha \left(u_j \delta\eta - u_j \eta \ln(u_j) - u_j + \frac{1}{2} \right) + 4 \left(u_j - \frac{1}{2} \right) \geq 0. \tag{C.10}$$

Clearly, $\alpha = 0$ satisfies this inequality; we are going to find the maximum α that does that. Without loss of generality, we can assume $\eta = 1$. We will discuss three cases with respect to the magnitude of u_j.

If we substitute the bound we found above for $\varDelta_{12}$ and remove the negative term in the coefficient of $-\alpha^2$, a sufficient condition for (C.9) to hold becomes

$$-\alpha^2 \left(n + \eta^2 n \ln^2 \left(\frac{n+1}{2} \right) \right) + 4\alpha \left(u_j \delta\eta - u_j \eta \ln(u_j) - u_j + \frac{1}{2} \right) + 4 \left(u_j - \frac{1}{2} \right) \geq 0. \tag{C.11}$$

If $u_j \geq 1$, the third term in (C.11) is at least 2. On the other hand, as $\sum_{j=1}^n u_j = n$ and $u_j \geq 1/2$ for all j, we have $u_j \leq (n+1)/2$ for all j. Using these, we have

$$\begin{aligned} n + \eta^2 n \ln^2 \left(\frac{n+1}{2} \right) &\leq 16\eta^2 n^2 \ln^2(n), \\ u_j \eta \ln(u_j) + u_j &\leq 4\eta n \ln(n), \end{aligned} \tag{C.12}$$

for $\eta = 1$ and $n \geq 2$. Therefore, $\alpha = \frac{1}{4\eta n \ln(n)}$ satisfies (C.11) for $n \geq 2$.
If $1 > u_j \geq \frac{9}{16}$, for $\eta = 1$ we easily have

$$u_j \delta\eta - u_j \eta \ln(u_j) - u_j + \frac{1}{2} \geq -u_j \eta \ln(u_j) - u_j + \frac{1}{2} \geq -\frac{1}{2}.$$

Hence, a sufficient condition for (C.11) to hold is:

$$-\alpha^2 \left(n + \eta^2 n \ln^2 \left(\frac{n+1}{2} \right) \right) - 2\alpha + \frac{1}{4} \geq 0.$$

Let $\alpha = \frac{1}{10\eta n \ln(n)}$, then similar to the previous case, for $\eta = 1$ the above inequality holds for $n \geq 2$.
The case of $\frac{1}{2} \leq u_j \leq \frac{9}{16}$ is the critical case as the third term in (C.10) can be zero. For $\eta = 1$, we have

$$-u_j\eta\ln(u_j) - u_j + \frac{1}{2} \geq -\frac{9}{16}\ln\left(\frac{9}{16}\right) - \frac{9}{16} + \frac{1}{2} \geq \frac{1}{5}\cdot$$

Hence, a sufficient condition for (C.10) to hold is (we again remove the negative term in the coefficient of $-\alpha^2$):

$$-\alpha^2(n + \Delta_{12}\eta^2) + 4\alpha u_j\delta\eta + \frac{4}{5}\alpha \geq 0,$$

Here, we use the result of Lemma 2.12 to upper bound Δ_{12} and the fact that $u_j \geq 1/2$ to make the sufficient condition stronger as (we substitute $\eta = 1$)

$$-\alpha^2(n + 2(\ln(n)+1)n\delta) + 2\alpha\delta + \frac{4}{5}\alpha \geq 0.$$

Now it is easy to check that $\alpha = \frac{1}{10n\ln(n)}$ satisfies the inequality for $n \geq 2$. If we substitute this α we get

$$\begin{aligned}
&-\frac{n + 2(\ln(n)+1)n\delta}{100n^2\ln^2(n)} + \frac{2\delta}{10n\ln(n)} + \frac{4}{50n\ln(n)} \\
&\qquad = 2\left(-\frac{(\ln(n)+1)}{100n\ln^2(n)} + \frac{1}{10n\ln(n)}\right)\delta + \left(-\frac{1}{100n\ln^2(n)} + \frac{4}{50n\ln(n)}\right) \geq 0,
\end{aligned}$$

which holds for $n \geq 2$.

Hence, for the case $\eta = 1$, we see that the constant step length of $\alpha = \frac{1}{10n\ln(n)}$ achieves the iteration complexity bound of $O\left(n\ln(n)\ln\left(\frac{1}{\epsilon}\right)\right)$. Therefore, the same iteration complexity bound holds for Algorithm 4.1. □

Proof of Theorem 4.4.

Proof. As $u_j \geq \frac{1}{2}$ for all j, then for each $u_j \leq 1$ we have $0 \leq \delta - \ln(u_j) \leq \delta + \ln(2)$. Let us define $J \subset \{1,\ldots,n\}$ as the set of indices for which $u_j \leq \frac{3}{4}$. By using this and (4.1), and substituting $\eta = 1/(\delta + \ln(2))$ we have

$$\|w(\eta)\|^2 = \sum_{j\notin J} x_js_j + \sum_{j\in J} x_js_j\left(-1 + \frac{\delta - \ln(u_j)}{\delta + \ln(2)}\right)^2 \leq \sum_{j=1}^{n} x_js_j = n\mu. \tag{C.13}$$

We must show that we still have enough reduction in the duality gap. First note that $\sum_{j=1}^n u_j = n$, and we have $\sum_{j\in J} u_j \leq \frac{3}{4}n$. This means $\sum_{j\in J} x_js_j \leq \frac{3}{4}\sum_{j=1}^n x_js_j$. Thus, we have:

$$\begin{aligned}
x(\alpha)^\top s(\alpha) &= (1-\alpha)x^\top s + \alpha\sum_{j\in J} x_js_j\left(\frac{\delta - \ln(u_j)}{\delta + \ln(2)}\right) \\
&\leq (1-\alpha)x^\top s + \frac{3}{4}\alpha x^\top s = \left[1 - \frac{1}{4}\alpha\right]x^\top s.
\end{aligned} \tag{C.14}$$

We want $\alpha > 0$ as large as possible such that $\frac{x_j(\alpha)s_j(\alpha)}{\mu(\alpha)} \geq \frac{1}{2}$ for all j. Using (C.14) we have

$$\frac{x_j(\alpha)s_j(\alpha)}{\mu(\alpha)} \geq \frac{x_j(\alpha)s_j(\alpha)}{\left(1 - \frac{1}{4}\alpha\right)\mu} \geq \frac{1}{2}\cdot$$

Using $|(w_p)_j(w_q)_j| \leq \|w(\eta)\|^2/4$ and (C.13), it is sufficient for $\alpha > 0$ to satisfy:

$$-2\left(1-\frac{1}{4}\alpha\right)+4(1-\alpha)u_j-\alpha^2 n \geq 0, \quad j \notin J,$$
$$-2\left(1-\frac{1}{4}\alpha\right)+4(1-\alpha)u_j+4\alpha u_j\eta(\delta-\ln(u_j))-\alpha^2 n \geq 0, \quad j \in J. \tag{C.15}$$

Let us multiply both of the inequalities by 2 and rewrite them as

$$-2n\alpha^2+(1-8u_j)\alpha+(-4+8u_j) \geq 0, \quad j \notin J,$$
$$-2n\alpha^2+[1-8u_j+8u_j\eta(\delta-\ln(u_j))]\alpha+(-4+8u_j) \geq 0, \quad j \in J. \tag{C.16}$$

Note that $u_j \geq \frac{1}{2}$ for all j and also $\sum_{j=1}^n u_j = n$, so $u_j \leq \frac{n+1}{2}$ for all j. On the other hand, $u_j > \frac{3}{4}$ for $j \notin J$ and so $-4+8u_j > 2$. Therefore, for $j \notin J$, (C.16) is satisfied if

$$-2n\alpha^2-(3+4n)\alpha+2 \geq 0.$$

Clearly for $\alpha = \frac{1}{5n}$, this inequality is satisfied for every $n \geq 1$.

For $j \in J$, we use the following inequality:

$$\eta(\delta-\ln(u_j)) = \frac{\delta-\ln(u_j)}{\delta+\ln(2)} \geq \frac{-\ln(u_j)}{\ln(2)}, \quad \forall\ u_j \geq \frac{1}{2}. \tag{C.17}$$

Substituting this in (C.16), we get a stronger sufficient condition as:

$$-2n\alpha^2+\left[1-8u_j-8\frac{u_j\ln(u_j)}{\ln(2)}\right]\alpha+(-4+8u_j) \geq 0. \tag{C.18}$$

For $j \in J$, we have $u_j \in \left[\frac{1}{2},\frac{3}{4}\right]$. We further split this case into two cases: $u_j \in \left[\frac{1}{2},0.55\right]$ and $u_j \in \left(0.55,\frac{3}{4}\right]$.

For $u_j \in (0.55, 0.75]$, it is not difficult to see the minimum of the coefficient of α in (C.18) is achieved at 0.75. Therefore, (C.18) is satisfied if

$$-2n\alpha^2+\left[1-8\times 0.75-8\times\frac{0.75\ln(0.75)}{\ln(2)}\right]\alpha+(-4+8\times 0.55) \geq -2n\alpha^2-3\alpha+0.4 \geq 0,$$

which holds for $\alpha = \frac{1}{10n}$.

For $u_j \in \left[\frac{1}{2},0.55\right]$, similarly it is not difficult to see the minimum of the coefficient of α in (C.18) is achieved at 0.55. Therefore, (C.18) is satisfied if

$$-2n\alpha^2+\left[1-8\times 0.55-8\times\frac{0.55\ln(0.55)}{\ln(2)}\right]\alpha+(-4+8\times 0.5) \geq -2n\alpha^2+0.39\alpha \geq 0,$$

which holds for $\alpha = \frac{1}{10n}$. We conclude that $\alpha = \frac{1}{10n}$ satisfies (C.15) for all possible cases, and in a similar way to the proof of Theorem 4.3, we can conclude the desired iteration complexity bound. □

References

[1] 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) 400–417.

[2] Y.Q. Bai, M. El Ghami and C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization. *SIAM J. Optim.* **15** (2004) 101–128.

[3] X.Z. Cai, G.Q. Wang, M. El Ghami and Y.J. Yue, Complexity analysis of primal-dual interior-point methods for linear optimization based on a new parametric kernel function with a trigonometric barrier term. *Abstr. Appl. Anal.* **2014** (2014) 710158.

[4] C.I. Chang, Y. Du, J. Wang, S.M. Guo and P.D. Thouin, Survey and comparative analysis of entropy and relative entropy thresholding techniques. In Vol. 153 of *IEE Proceedings Vision, Image and Signal Processing*. IET (2006) 837–850.

[5] T.M. Cover and J.A. Thomas, Elements of information theory. John Wiley & Sons (2012).

[6] Zs. Darvay, A new algorithm for solving self-dual linear optimization problems. *Stud. Univ. Babęs-Bolyai* **47** (2003) 15–26.

[7] A. Decarreau, D. Hilhorst, C. Lemaréchal and J. Navaza, Dual methods in entropy maximization. application to some problems in crystallography. *SIAM J. Optim.* **2** (1992) 173–197.

[8] T. Downarowicz, *Entropy in dynamical systems*. Cambridge University Press (2011)

[9] N.J. Dusaussoy and I.E. Abdou, The extended ment algorithm: a maximum entropy type algorithm using prior knowledge for computerized tomography. *IEEE Trans. Signal Proces.* **39** (1991) 1164–1180.

[10] M. Dyer, A class of convex programs with applications to computational geometry. In *Proc. of the Eighth Annual Symposium on Computational Geometry*. ACM (1992) 9–15.

[11] H. Edelsbrunner, *Algorithms in Computational Geometry*. Springer New York (1987).

[12] S. Erlander, Accessibility, entropy and the distribution and assignment of traffic. *Transport. Res.* **11** (1977) 149–153.

[13] S. Erlander, Entropy in linear programs. *Math. Program.* **21** (1981) 137–151.

[14] S. Fang, J.R. Rajasekera and H.J. Tsao, Vol. 8 of *Entropy Optimization and Mathematical Programming*. Springer (1997).

[15] R.M. Freund, On the behavior of the homogeneous self-dual model for conic convex optimization. *Math. Program.* **106** (2006) 527–545.

[16] N. Karmarkar, A new polynomial time algorithm for linear programming. *Combinatorica* **4** (1984) 373–395.

[17] Y.H. Lee, J.H. Jin and G.M. Cho, Kernel function based interior-point algorithms for semidefinite optimization. *Math. Inequal. Appl.* **16** (2013) 1279–1294.

[18] X.S. Li, *Entropy and optimization.* Ph.D. thesis, University of Liverpool, UK (1987).

[19] S. Luo, *Interior-Point Algorithms Based on Primal-Dual Entropy.* Master's thesis, University of Waterloo, Canada (2006).

[20] N. Megiddo, Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. In *Foundations of Computer Science. SFCS'08. 23rd Annual Symposium on.* IEEE (1982) 329–338.

[21] S. Mizuno, M.J. Todd and Y. Ye, On adaptive-step primal-dual interior-point algorithms for linear programming. *Math. Oper. Res.* **18** (1993) 964–981.

[22] R.D.C. Monteiro, I. Adler and M.G.C. Resende, A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Math. Operat. Res.* **15** (1990) 191–214.

[23] C. Nadeu and M. Bertran, A flatness-based generalized optimization approach to spectral estimation. *Signal Process.* **19** (1990) 311–320.

[24] J.L. Nazareth, *A reformulation of the central path equations and its algorithmic implications.* Department of Pure and Applied Math., Washington State Univ. (1994).

[25] J.L. Nazareth, Deriving potential functions *via* a symmetry principle for nonlinear equations. *Oper. Res. Lett.* **21** (1997) 147–152.

[26] Y. Nesterov, Parabolic target space and primal–dual interior-point methods. *Discrete Appl. Math.* **156** (2008) 2079–2100.

[27] NETLIB. Available at http://www.netlib.org/lp/data/.

[28] S. Pan, X. Li and S. He, An infeasible primal–dual interior point algorithm for linear programs based on logarithmic equivalent transformation. *J. Math. Anal. Appl.* **314** (2006) 644–660.

[29] M.Ç. Pınar and S.A. Zenios, An entropic approximation of ℓ_1 penalty function. *Trans. Oper. Res.* (1995) 101–120.

[30] F.A. Potra, Primal-dual affine scaling interior point methods for linear complementarity problems. *SIAM J. Optim.* **19** (2008) 114–143.

[31] T. Pynchon, A survey of entropy methods for partial differential equations. *Amer. Math. Soc.* **41** (2004) 409–438.

[32] C. E. Shannon, A mathematical theory of communication. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* **5** (2001) 3–55.

[33] J.F. Sturm, Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optim. Meth. Softw.* **17** (2002) 1105–1154.

[34] L. Tunçel, On the convergence of primal-dual interior-point methods with wide neighborhoods. *Comput. Optim. Appl.* **4** (1995) 139–158.

[35] L. Tunçel and M.J. Todd, On the interplay among entropy, variable metrics and potential functions in interior-point algorithms. *Comput. Optim. Appl.* **8** (1997) 5–19.

[36] Y. Ye, An $O(n^3L)$ potential reduction algorithm for linear programming. *Math. Program.* **50** (1991) 239–258.

[37] Y. Ye, M.J. Todd and S. Mizuno, An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.* **19** (1994) 53–67.

[38] P. Zhang and X. Li, An infeasible-start path-following method for monotone LCPs. *Math. Comput. Model.* **38** (2003) 23–31.