An Asymptotical $O(\sqrt{n}L)$-Iteration Path-Following Linear Programming Algorithm That Uses Wide Neighborhoods *

Pi-Fang Hung † Yinyu Ye ‡

March 1994

Abstract: Path-following linear programming (LP) algorithms generate a sequence of points within certain neighborhoods of a central-path $\mathcal{C}$, which prevent iterates from prematurely getting too close to the boundary of the feasible region. Depending on their norm used, these neighborhoods include $\mathcal{N}_2(\beta)$, $\mathcal{N}_\infty(\beta)$ and $\mathcal{N}_\infty^-(\beta)$, where $\beta \in (0,1)$, and

$$\mathcal{C} \subset \mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta) \subset \mathcal{N}_\infty^-(\beta) \quad \text{for each} \quad \beta \in (0,1).$$

A paradox is that among all existing (infeasible or feasible) path-following algorithms, the theoretical iteration complexity, $O(\sqrt{n}L)$, of small-neighborhood ($\mathcal{N}_2$) algorithms is significantly better than the complexity, $O(nL)$, of wide-neighborhood ($\mathcal{N}_\infty^-$) algorithms, while in practice wide-neighborhood algorithms outperform small-neighborhood ones by a big margin. Here, n is the number of LP variables and L is the LP data length. In this paper, we present an $O(n^{\frac{n+1}{2n}}L)$-iteration (infeasible) primal-dual high-order algorithm that uses wide neighborhoods. Note that this iteration bound is asymptotical $O(\sqrt{n}L)$, i.e., the best bound for small-neighborhood algorithms, as n increases.

Key words: Linear programming, central-path, interior-point algorithm, complexity.

1 Introduction

Consider a linear programming (LP) problem in the standard form:

$$\begin{array}{lll} (LP) & \text{minimize} & c^T x \\ & \text{subject to} & Ax = b, \; x \geq 0, \end{array}$$

*This work was supported in part by NSF Grant DDM-9207347.

†Department of Mathematics, Tunghai University, Taichung, Taiwan 40704, ROC. This work is based on the author's Ph. D. thesis in the Applied Mathematical and Computational Sciences Program, Department of Mathematics, The University of Iowa, Iowa City, Iowa 52242, USA.

‡Department of Management Sciences, The University of Iowa, Iowa City, Iowa 52242, USA.

where $c \in R^n$, $A \in R^{m \times n}$ and $b \in R^m$ are given, $x \in R^n$, and superscript ^{T} denotes transpose. (LP) is said to be feasible if and only if its constraints are consistent; it is called unbounded if there is a sequence $\{x^k\}$ such that x^k is feasible for all k but $c^T x^k \to -\infty$. (LP) has a solution if and only if it is feasible and bounded.

The dual problem of (LP) can be written as

$$\begin{array}{lll} (LD) & \text{maximize} & b^T y \\ & \text{subject to} & A^T y \le c, \end{array}$$

where $y \in R^m$. We call $s := c - A^T y \in R^n$ dual slacks. Denote by $\mathcal{F}$ the set of all (x, s) such that x and (y, z) are feasible for the primal and dual, respectively. Denote by $\mathcal{F}^0$ the set of points with $(x, s) > 0$ in $\mathcal{F}$.

Assuming that the problem has a feasible interior point, Bayer and Lagarias [2], Megiddo [11], and Sonnevend [22] defined the central path for a feasible problem as

$$\mathcal{C} = \left\{ (x, s) \in \mathcal{F}^0 : Xs = \frac{x^T s}{n} e \right\}$$

in primal-dual form, where e denotes the vector of ones and $X = \operatorname{diag}(x)$ denotes the diagonal matrix with diagonal entries equal to the elements of x. As $x^T s \to 0$, this path leads to a strictly complementary solution of (LP) and (LD).

There are two types of polynomial interior-point algorithms. One is called potential reduction algorithm that is not based on the central-path and its neighborhood. This group includes the $O(nL)$-iteration projective-scaling algorithm (e.g., Karmarkar [7]) and the $O(\sqrt{n}L)$-iteration affine-scaling algorithm (e.g., Anstreicher and Bosch and references therein [1]) among others, where each iteration uses $O(n^3)$ arithmetic operations. Here, L represents the data length of rational LP data, or the precision to achieve $(x^T s)/(x^0)^T s^0 \le 2^{-L}$ for a given initial point (x^0, s^0). The practical value of these algorithms relies on an efficient line search technique to select step-size, which could be costly for solving some LP problem. (Several other long-step strategies are recently proposed by Nesterov [17].)

The second type is called path-following algorithm. Path-following algorithms generate a sequence of points within certain neighborhoods of the central path $\mathcal{C}$, which prevent iterates from prematurely getting too close to the boundary of the feasible region. Depending on the norm used in their definition, these neighborhoods include,

$$\mathcal{N}_2(\beta) = \left\{ (x, s) \in \mathcal{F}^0 : \|Xs - \mu e\| \le (1 - \beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}$$

for some $\beta \in (0, 1)$ (where $\|\cdot\|$ without subscript designates the ℓ_2 norm),

$$\mathcal{N}_\infty(\beta) = \left\{ (x, s) \in \mathcal{F}^0 : \|Xs - \mu e\|_\infty \le (1 - \beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\}$$

and

$$\mathcal{N}_\infty^-(\beta) = \left\{ (x, s) \in \mathcal{F}^0 : \|Xs - \mu e\|_\infty^- \le (1 - \beta)\mu \quad \text{where} \quad \mu = \frac{x^T s}{n} \right\},$$

for any $\beta \in (0,1)$. Here, for any $z \in R^n$,

$$\|z\|_\infty^- := \|z^-\|_\infty$$

and

$$\|z\|_\infty^+ := \|z^+\|_\infty,$$

where $(z^-)_j := \min\{z_j, 0\}$ and $(z^+)_j := \max\{z_j, 0\}$ and $\|\cdot\|_\infty$ is the usual ℓ_∞ norm. Note that $\|z\|_\infty = \max\{\|z\|_\infty^+, \|z\|_\infty^-\}$ and that neither $\|\cdot\|_\infty^-$ nor $\|\cdot\|_\infty^+$ is a norm, although they obey the triangle inequality. Hence, $\mathcal{N}_\infty^-(\beta)$ can be written as

$$\mathcal{N}_\infty^-(\beta) = \left\{(x,s) \in \mathcal{F}^0 : Xs \geq \beta\mu e \quad \text{where} \quad \mu = \frac{x^T s}{n}\right\}.$$

We easily see that

$$\mathcal{C} \subset \mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta) \subset \mathcal{N}_\infty^-(\beta) \subset \mathcal{F}^0 \text{ for each } \beta \in (0,1).$$

In particular,

$$\mathcal{N}_\infty^-(0) = \mathcal{F}^0,$$

so when β is close to 0, the neighborhood $\mathcal{N}_\infty^-(\beta)$ spreads almost all over the feasible region $\mathcal{F}$ and the points generated by the algorithm based on $\mathcal{N}_\infty^-(\beta)$ are close to the boundary rather than the central path.

Among all existing path-following (infeasible or feasible) LP algorithms, the theoretical iteration complexity of small-neighborhood ($\mathcal{N}_2$) algorithms is $O(\sqrt{n}L)$ (e.g., Renegar [21], Gonzaga [5], Kojima et al. [9] and Monteiro and Adler [13]), and the complexity of wide-neighborhood ($\mathcal{N}_\infty$ or $\mathcal{N}_\infty^-$) algorithms is at least $O(nL)$ (e.g., Kojima et al. [8], Mizuno et al. [15], Potra [19], and Roos and Vial [20]). In contrast, wide-neighborhood algorithms outperform small-neighborhood ones by a big margin in practice (e.g., Lustig et al. [10] and Mehrotra [12]). It seems that smaller neighborhoods generally restrict all iterates moved by a short step and they might be too conservative for solving real LP problems.

A few results have been developed in attempting to improve the theoretical complexity of wide-neighborhood algorithms that use $\mathcal{N}_\infty$ or $\mathcal{N}_\infty^-$. One result is due to Monteiro et al. [14], who developed a r-order power-series algorithm ($r \in [1,n]$), where iterative points move along a r-order polynomial curve, with iteration complexity $O(n^{\frac{r+1}{2r}} L^{\frac{r+1}{r}})$. Their algorithm requires an initial feasible point in $\mathcal{N}_\infty(1-\beta^0)$ with $\beta^0 \in (0,1)$. Then, it generates a direction and moves the iterate in this direction by a fixed step size $O(n^{\frac{-r-1}{2r}} L^{\frac{-1}{r}})$. They have shown that the new iterate will be in $\mathcal{N}_\infty(1-\beta^1)$ where $\beta^1 > \beta^0$. In general, the kth iterate will be in $\mathcal{N}_\infty(1-\beta^k)$, while β^k increases and approaches to an upper bound $(1+\beta^0)/2$. We can see that all iterates lie in $\mathcal{N}_\infty((1-\beta^0)/2)$. Unfortunately, their pre-planned step size selection still takes a very short step in each iteration.

The second result is due to Zhang and Zhang [27] who proved that a 2-order power-series algorithm, similar to the one introduced by Monteiro et al. [14] and Mehrotra [12], possesses $O(n^{3/4}L)$-iteration complexity under the assumption that the initial point is feasible and in $\mathcal{N}_\infty^-(\beta)$. Subsequently, all iterates lie in $\mathcal{N}_\infty^-(\beta)$, where β is fixed during entire iterations. If the initial positive point is not feasible, then after using a "big-M" technique their algorithm's iteration complexity is $O(n^{3/2}L)$, again greater than $O(nL)$. Tapia et al. [23] also studied the high-order predictor-corrector algorithm and compared it to the composite Newton method.

On the other hand, the high-order power-series algorithm has gained tremendous popularity in practice due to the computational results of Mehrotra [12], Lustig et al. [10], and Carpenter et al. [3] among others. Their high-order algorithms actually use an infeasible initial point, and allow iterative points move in $\mathcal{N}_\infty^\sim(\beta)$ where

$$\mathcal{N}_\infty^\sim(\beta) = \left\{(x \geq 0, s \geq 0) : Xs \geq \beta\mu e \quad \text{where} \quad \mu = \frac{x^T s}{n}\right\},$$

and β is fixed in the course of iterations and it is very close to 0. Obviously,

$$\mathcal{N}_\infty^-(\beta) \subset \mathcal{N}_\infty^\sim(\beta),$$

and $\mathcal{N}_\infty^\sim(0)$ spreads the whole non-negative orthant. To our knowledge, no such an algorithm, which is based on $\mathcal{N}_\infty^\sim(\beta)$ and whose complexity is below $O(nL)$, is ever developed or proved.

Therefore, it seems that there is a paradox between theory and practice. The question is, can we smoothen this conflict?

In this paper, we present a r-order power-series primal-dual path-following algorithm that is based on $\mathcal{N}_\infty^-(\beta)$ where β is any fixed constant in $(0, 1)$. We show that its iteration complexity is $O(n^{\frac{r+1}{2r}}L)$ where $r \in [1, n]$. Again, each iteration uses $O(n^3)$ arithmetic operations. Note that if we let $r = n$, then this iteration bound is asymptotical $O(\sqrt{n}L)$ as n increases. Moreover, our algorithm does not need the initial point to be feasible and does not use any big-M parameter, that is, the algorithm is actually based on $\mathcal{N}_\infty^\sim(\beta)$. It solves LP problems without any regularity assumption concerning the existence of optimal, feasible, or interior feasible solutions. If the LP problem has a solution, the algorithm generates a sequence that approaches feasibility and optimality simultaneously; if the problem is infeasible or unbounded, the algorithm correctly detects infeasibility.

Our algorithm is built upon the results mentioned above, and it is an extension of the homogeneous self-dual (HSD) algorithm studied by Ye et al. [26] and later by Xu et al. [24][25]. We describe our generic algorithm in the next section and develop several general lemmas. Then, we present a special algorithm in Section 3, and analyze its complexity in Section 4. Another special algorithm is also shown in Section 5. Finally, conclusions and implications based on our main result are summarized in Section 6.

2 The r-Order Power-Series and Homogeneous Self-Dual Algorithm

Recently, Xu et al. [25] proposed a simplified version of the HSD Algorithm of Ye et al. [26]. The algorithm deals with a homogeneous and self-dual linear feasibility model of Goldman and Tucker [4]:

$$
\begin{array}{llllr}
(HLF) & & Ax & -b\tau & = 0, \\
 & -A^T y & & +c\tau & \geq 0, \\
 & b^T y & -c^T x & & \geq 0, \\
 & y \text{ free} & x \geq 0 & \tau \geq 0. &
\end{array} \tag{1}
$$

Denote by s the slack vector for the second (inequality) constraint and by κ the slack scalar for the third (inequality) constraint. Then, the problem is to find a strictly complementary point such that

$$x^T s = 0, \ \tau\kappa = 0, \quad \text{and} \quad x + s > 0, \ \tau + \kappa > 0.$$

Given $(y^k, \ x^k > 0, \tau^k > 0, s^k > 0, \kappa^k > 0)$ be an (infeasible) interior point in $\mathcal{N}_\infty^{\sim}(\beta)$ for (HLF), where $\beta \in (0, 1)$, the kth iteration of the HSD algorithm solves the following system of linear equations for the direction $(d_y, d_x, d_\tau, d_s, d_\kappa)$

$$
\begin{array}{rrrrrcr}
 & A \ d_x & -b \ d_\tau & & & = & \eta \ r_P^k, \\
-A^T \ d_y & & +c \ d_\tau & -d_s & & = & -\eta \ r_D^k, \\
b^T \ d_y & -c^T \ d_x & & & -d_\kappa & = & \eta \ r_G^k,
\end{array} \tag{2}
$$

and

$$
\begin{aligned}
X^k d_s + S^k d_x &= \gamma\mu^k e - X^k s^k, \\
\tau^k d_\kappa + \kappa^k d_\tau &= \gamma\mu^k - \tau^k \kappa^k,
\end{aligned} \tag{3}
$$

where non-negative η and γ are two parameters, and

$$\mu^k = ((x^k)^T s^k + \tau^k \kappa^k)/(n + 1), \tag{4}$$

$$r_P^k = b\tau^k - Ax^k, \quad r_D^k = c\tau^k - A^T y^k - s^k, \quad r_G^k = c^T x^k - b^T y^k + \kappa^k. \tag{5}$$

Xu et.al [25] showed that if $\eta = 1 - \gamma$ is chosen in each iteration for some $0 < \gamma < 1$, then the algorithm becomes the HSD algorithm of Ye et al. [26], which generates iterates such that

$$\frac{\mu^k}{\theta^k} = \frac{\mu^0}{\theta^0} \quad \text{for all} \quad k \geq 0,$$

and

$$r_P^k = \theta^k r_P^0, \quad r_D^k = \theta^k r_D^0, \quad r_G^k = \theta^k r_G^0, \quad \text{with} \quad \theta^0 = 1.$$

Here, μ^k represents the complementarity gap and θ^k measures the infeasibility error at the kth iteration. The above relation indicates that both measures tend to zero at a same rate. Furthermore, (HLF) is self-dual and its solution set is unbounded. Thus, based on Theorem 4 of Mizuno et al. [16] (also see Lemma 2 of Güler and Ye [6]), every limit point of iterates is strictly complementary for (HLF) as long as iterates lie in $\mathcal{N}_\infty^\sim(\beta)$. Moreover, if $\tau^k \to 0$, then the original (LP) problem is either infeasible or unbounded.

Now, we describe a generic high-order power-series and homogeneous self-dual algorithm with the same feature:

The r-Order HSD algorithm

Given $(y^k,\ x^k > 0, \tau^k > 0, s^k > 0, \kappa^k > 0)$ be an (infeasible) interior point in $\mathcal{N}_\infty^\sim(\beta)$ for (HLF):

$$\mathcal{N}_\infty^\sim(\beta) = \{(x \geq 0, s \geq 0, \tau \geq 0, \kappa > 0) : Xs \geq \beta\mu e \quad \text{and} \quad \tau\kappa \geq \beta\mu\}\,, \tag{6}$$

where $\beta \in (0,1)$ and it is fixed during iterations, and

$$\mu = (x^T s + \tau\kappa)/(n+1).$$

Given integer $r \geq 1$ and scalar parameters $\gamma \geq 0,\ \eta^{(j)} \geq 0, j = 1, \ldots, r$, the kth iteration of the algorithm consists of the following steps:

1. Solve for the first order direction $(d_y^{(1)}, d_x^{(1)}, d_\tau^{(1)}, d_s^{(1)}, d_\kappa^{(1)})$ from

$$\begin{array}{rrrrrcl} & A\ d_x^{(1)} & -b\ d_\tau^{(1)} & & & = & \eta^{(1)} r_P^k, \\ -A^T\ d_y^{(1)} & & +c\ d_\tau^{(1)} & -d_s^{(1)} & & = & -\eta^{(1)} r_D^k, \\ b^T\ d_y^{(1)} & -c^T\ d_x^{(1)} & & & -d_\kappa^{(1)} & = & \eta^{(1)} r_G^k. \end{array} \tag{7}$$

and

$$\begin{array}{rcl} X^k d_s^{(1)} + S^k d_x^{(1)} & = & \gamma\mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} & = & \gamma\mu^k - \tau^k\kappa^k. \end{array} \tag{8}$$

2. For $j = 2, 3, \ldots . r$, solve for the jth order direction $(d_y^{(j)}, d_x^{(j)}, d_\tau^{(j)}, d_s^{(j)}, d_\kappa^{(j)})$ from

$$\begin{array}{rrrrrcl} & A\ d_x^{(j)} & -b\ d_\tau^{(j)} & & & = & \eta^{(j)}\ r_P^k, \\ -A^T\ d_y^{(j)} & & +c\ d_\tau^{(j)} & -d_s^{(j)} & & = & -\eta^{(j)}\ r_D^k, \\ b^T\ d_y^{(j)} & -c^T\ d_x^{(j)} & & & -d_\kappa^{(j)} & = & \eta^{(j)}\ r_G^k, \end{array} \tag{9}$$

and

$$\begin{array}{rcl} X^k d_s^{(j)} + S^k d_x^{(j)} & = & -\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} & = & -\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{array} \tag{10}$$

3. Calculate $f^{(j)}$ according to

$$f^{(1)} = \eta^{(1)} + \gamma - 1, \quad f^{(j)} = \eta^{(j)} + \sum_{t=1}^{j-1} \eta^{(t)} f^{(j-t)}, \quad 2 \leq j \leq r,$$

and choose a step-size $\alpha^k > 0$ such that the new iterate,

$$\begin{array}{rcl} x^{k+1} & = & \left(x^k + \sum_{j=1}^{r}(\alpha^k)^j d_x^{(j)}\right) / \left(1 + \sum_{j=1}^{r}(\alpha^k)^j f^{(j)}\right), \\ y^{k+1} & = & \left(y^k + \sum_{j=1}^{r}(\alpha^k)^j d_y^{(j)}\right) / \left(1 + \sum_{j=1}^{r}(\alpha^k)^j f^{(j)}\right), \\ s^{k+1} & = & \left(s^k + \sum_{j=1}^{r}(\alpha^k)^j d_s^{(j)}\right) / \left(1 + \sum_{j=1}^{r}(\alpha^k)^j f^{(j)}\right), \\ \tau^{k+1} & = & \left(\tau^k + \sum_{j=1}^{r}(\alpha^k)^j d_\tau^{(j)}\right) / \left(1 + \sum_{j=1}^{r}(\alpha^k)^j f^{(j)}\right), \\ \kappa^{k+1} & = & \left(\kappa^k + \sum_{j=1}^{r}(\alpha^k)^j d_\kappa^{(j)}\right) / \left(1 + \sum_{j=1}^{r}(\alpha^k)^j f^{(j)}\right), \end{array}$$

lies in $\mathcal{N}_\infty^{\sim}(\beta)$.

□

Regarding the algorithm, we have following lemmas which are similar to those proved in Xu et al. [24][25].

Lemma 1 . *The direction defined by (7),(8),(9) and (10) satisfies:*

$$(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} = -(\eta^{(i)} f^{(j)} + \eta^{(j)} f^{(i)})((x^k)^T s^k + \tau^k \kappa^k) \tag{11}$$

and

$$(d_x^{(j)})^T d_s^{(j)} + d_\tau^{(j)} d_\kappa^{(j)} = -\eta^{(j)} f^{(j)}((x^k)^T s^k + \tau^k \kappa^k), \tag{12}$$

for $i, j = 1, \ldots, r$.

Proof. First, we have

$$\begin{array}{rl} & (d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} \\ = & (d_x^{(i)})^T(-A^T\ d_y^{(j)} + c\ d_\tau^{(j)} + \eta^{(j)} r_D^k) + (d_x^{(j)})^T(-A^T\ d_y^{(i)} + c\ d_\tau^{(i)} + \eta^{(i)} r_D^k) \\ & +d_\tau^{(i)}(b^T\ d_y^{(j)} - c^T\ d_x^{(j)} - \eta^{(j)} r_G^k) + d_\tau^{(j)}(b^T\ d_y^{(i)} - c^T\ d_x^{(i)} - \eta^{(i)} r_G^k) \\ = & -\eta^{(i)}\left((d_y^{(j)})^T r_P^k - (d_x^{(j)})^T r_D^k + d_\tau^{(j)}\ r_G^k\right) - \eta^{(j)}\left((d_y^{(i)})^T r_P^k - (d_x^{(i)})^T r_D^k + d_\tau^{(i)}\ r_G^k\right) \\ = & -\eta^{(i)}\left\{\eta^{(j)}((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)})\right\} \\ & -\eta^{(j)}\left\{\eta^{(i)}((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(i)} + (s^k)^T d_x^{(i)} + \tau^k d_\kappa^{(i)} + \kappa^k d_\tau^{(i)})\right\}. \end{array}$$

Now, we prove the lemma by induction on i, j. Note that

$$\eta^{(1)}((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(1)} + (s^k)^T d_x^{(1)} + \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)}) = (\eta^{(1)} + \gamma - 1)((x^k)^T s^k + \tau^k \kappa^k)$$

implies that (12) is true when $i = j = 1$. Assume (11) and (12) hold for all i, j with $1 \le i, j \le p - 1$. We show that they hold for $i = p$ and $j = 1, \ldots, p$. In fact, we only need to prove that for $j = 1, \ldots, p$,

$$\eta^{(j)}((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)}) = f^{(j)}((x^k)^T s^k + \tau^k \kappa^k),$$

which is true because

$$
\begin{aligned}
& \eta^{(j)}((x^k)^T s^k + \tau^k \kappa^k) + ((x^k)^T d_s^{(j)} + (s^k)^T d_x^{(j)} + \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)}) \\
= \;& \eta^{(j)}((x^k)^T s^k + \tau^k \kappa^k) - \sum_{t=1}^{j-1} \left((d_x^{(t)})^T d_s^{(j-t)} + d_\tau^{(t)} \; d_\kappa^{(j-t)} \right) \\
= \;& \eta^{(j)}((x^k)^T s^k + \tau^k \kappa^k) + \left(\sum_{t=1}^{j-1} \eta^{(t)} f^{(j-t)} \right) ((x^k)^T s^k + \tau^k \kappa^k) \\
= \;& f^{(j)}((x^k)^T s^k + \tau^k \kappa^k).
\end{aligned}
$$

Similarly, we can show that (11) and (12) hold for $i = 1, \ldots, p$ and $j = p$, which completes the proof.

□

Let

$$
\begin{aligned}
\triangle x &= x^{k+1} - x^k = \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_x^{(j)} - f^{(j)} x^k) \right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)} \right), \\
\triangle y &= y^{k+1} - y^k = \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_y^{(j)} - f^{(j)} y^k) \right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)} \right), \\
\triangle s &= s^{k+1} - s^k = \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_s^{(j)} - f^{(j)} s^k) \right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)} \right), \\
\triangle \tau &= \tau^{k+1} - \tau^k = \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_\tau^{(j)} - f^{(j)} \tau^k) \right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)} \right), \\
\triangle \kappa &= \kappa^{k+1} - \kappa^k = \left(\textstyle\sum_{j=1}^r (\alpha^k)^j (d_\kappa^{(j)} - f^{(j)} \kappa^k) \right) / \left(1 + \textstyle\sum_{j=1}^r (\alpha^k)^j f^{(j)} \right).
\end{aligned}
$$

Then, we have

$$
\begin{array}{rrrrrcl}
 & A \triangle x & -b \triangle \tau & & & = & \frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} r_P^k, \\
-A^T \triangle y & & +c \triangle \tau & -\triangle s & & = & -\frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} r_D^k, \\
b^T \triangle y & -c^T \triangle x & & & -\triangle \kappa & = & \frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} r_G^k,
\end{array}
\tag{13}
$$

and

$$
\begin{aligned}
& X^k \triangle s + S^k \triangle x \\
= \;& \frac{1}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} \sum_{j=1}^r (\alpha^k)^j (X^k d_s^{(j)} + S^k d_x^{(j)} - f^{(j)} X^k s^k - f^{(j)} S^k x^k) \\
= \;& \frac{1}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k (\gamma \mu^k e - X^k s^k) + \sum_{j=2}^r (\alpha^k)^j (-\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}) - 2(\sum_{j=1}^r (\alpha^k)^j f^{(j)}) X^k s^k \right\} \\
= \;& \frac{1}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k \gamma \mu^k e - (\alpha^k + 2 \sum_{j=1}^r (\alpha^k)^j f^{(j)}) X^k s^k - \sum_{j=2}^r (\alpha^k)^j (\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}) \right\}
\end{aligned}
\tag{14}
$$

$$
\tau^k \triangle \kappa + \kappa^k \triangle \tau = \frac{1}{1 + \sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k \gamma \mu^k - (\alpha^k + 2 \sum_{j=1}^r (\alpha^k)^j f^{(j)}) \tau^k \kappa^k - \sum_{j=2}^r (\alpha^k)^j (\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}) \right\}.
$$

Using relation (14), we further have

$$
\begin{aligned}
& (x^k)^T \triangle s + (s^k)^T \triangle x + \tau^k \triangle \kappa + \kappa^k \triangle \tau \\
= & \frac{1}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k \gamma \mu^k (n+1) - \alpha^k ((x^k)^T s^k + \tau^k \kappa^k) - 2\sum_{j=1}^r (\alpha^k)^j f^{(j)} ((x^k)^T s^k + \tau^k \kappa^k) \right. \\
& \left. - \sum_{j=2}^r (\alpha^k)^j \sum_{t=1}^{j-1} ((d_x^{(t)})^T d_s^{(j-t)} + d_\tau^{(t)} d_\kappa^{(j-t)}) \right\} \\
= & \frac{1}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k \gamma - \alpha^k - 2\sum_{j=1}^r (\alpha^k)^j f^{(j)} + \sum_{j=2}^r (\alpha^k)^j \sum_{t=1}^{j-1} \eta^{(t)} f^{(j-t)} \right\} ((x^k)^T s^k + \tau^k \kappa^k) \quad (15) \\
= & \frac{1}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \alpha^k (f^{(1)} - \eta^{(1)}) - 2\sum_{j=1}^r (\alpha^k)^j f^{(j)} + \sum_{j=2}^r (\alpha^k)^j (f^{(j)} - \eta^{(j)}) \right\} ((x^k)^T s^k + \tau^k \kappa^k) \\
= & \frac{1}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ -2\sum_{j=1}^r (\alpha^k)^j f^{(j)} + \sum_{j=1}^r (\alpha^k)^j (f^{(j)} - \eta^{(j)}) \right\} ((x^k)^T s^k + \tau^k \kappa^k) \\
= & \frac{1}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ -\sum_{j=1}^r (\alpha^k)^j (f^{(j)} + \eta^{(j)}) \right\} ((x^k)^T s^k + \tau^k \kappa^k)
\end{aligned}
$$

which leads to the next lemma and proposition.

Lemma 2 .

$$\triangle x^T \triangle s + \triangle \tau \triangle \kappa = 0.$$

Proof. From equations (13) and (15), we derive

$$
\begin{aligned}
& \triangle x^T \triangle s + \triangle \tau \triangle \kappa \\
= & -\frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \left\{ \frac{\sum_{j=1}^r (\alpha^k)^j (\eta^{(j)} + f^{(j)})}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} ((x^k)^T z^k + \tau^k \kappa^k) \right. \\
& \left. + (x^k)^T \triangle s + (s^k)^T \triangle x + \tau^k \triangle \kappa + \kappa^k \triangle \tau \right\} \\
= & 0.
\end{aligned}
$$

□

Proposition 3 . *The generic algorithm generates $\{y^k,\ x^k, \tau^k, s^k, \kappa^k\}$ and $\{\mu^k, \theta^k\}$ satisfying*

$$\mu^0 = [(x^0)^T s^0 + \tau^0 \kappa^0] \ / \ (n+1),$$

$$\mu^{k+1} = \frac{1-\sum_{j=1}^r (\alpha^k)^j \eta^{(j)}}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \mu^k,$$

and

$$\theta^0 = 1, \quad \theta^{k+1} = \frac{1-\sum_{j=1}^r (\alpha^k)^j \eta^{(j)}}{1+\sum_{j=1}^r (\alpha^k)^j f^{(j)}} \theta^k, \tag{16}$$

with

$$r_P^k = \theta^k \ r_P^0, \quad r_D^k = \theta^k \ r_D^0, \quad r_G^k = \theta^k \ r_G^0. \tag{17}$$

$\square$

Proposition 3 states that the high-order algorithm possesses the property

$$\frac{\mu^k}{\theta^k} = \mu^0, \quad k \geq 1$$

as does the HSD algorithm of Ye et al. [26]. As we discussed early, every limit point of iterates is a strictly complementary solution for (HLF).

3 Special Case: Algorithm 1

We now analyze a special case where

$$\eta^{(1)} = 1 - \gamma, \quad \text{and} \quad \eta^{(j)} = 0 \quad \forall \quad j \geq 2.$$

Note that $f^{(j)} = 0, \ \forall \ j \geq 1$, in this case and the algorithm becomes:

Algorithm 1

1. Solve for the first order direction from

$$\begin{array}{rrrrrcl} & A \ d_x^{(1)} & -b \ d_\tau^{(1)} & & & = & (1-\gamma) r_P^k, \\ -A^T \ d_y^{(1)} & & +c \ d_\tau^{(1)} & -d_s^{(1)} & & = & -(1-\gamma) r_D^k, \\ b^T \ d_y^{(1)} & -c^T \ d_x^{(1)} & & & -d_\kappa^{(1)} & = & (1-\gamma) r_G^k. \end{array} \tag{18}$$

and

$$\begin{array}{rcl} X^k d_s^{(1)} + S^k d_x^{(1)} & = & \gamma \mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} & = & \gamma \mu^k - \tau^k \kappa^k. \end{array} \tag{19}$$

2. For $j = 2, 3, \ldots, r$, solve for the jth order direction from

$$\begin{array}{rrrrrcl} & A \ d_x^{(j)} & -b \ d_\tau^{(j)} & & & = & 0, \\ -A^T \ d_y^{(j)} & & +c \ d_\tau^{(j)} & -d_s^{(j)} & & = & 0, \\ b^T \ d_y^{(j)} & -c^T \ d_x^{(j)} & & & -d_\kappa^{(j)} & = & 0, \end{array} \tag{20}$$

and

$$\begin{array}{rcl} X^k d_s^{(j)} + S^k d_x^{(j)} & = & -\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} & = & -\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{array} \tag{21}$$

3. Choose a step-size α^k such that the new iterate

$$\begin{array}{rcl} x^{k+1} & = & x^k + \sum_{j=1}^r (\alpha^k)^j d_x^{(j)}, \\ y^{k+1} & = & y^k + \sum_{j=1}^r (\alpha^k)^j d_y^{(j)}, \\ s^{k+1} & = & s^k + \sum_{j=1}^r (\alpha^k)^j d_s^{(j)}, \\ \tau^{k+1} & = & \tau^k + \sum_{j=1}^r (\alpha^k)^j d_\tau^{(j)}, \\ \kappa^{k+1} & = & \kappa^k + \sum_{j=1}^r (\alpha^k)^j d_\kappa^{(j)} \end{array}$$

lies in $\mathcal{N}_\infty^\sim(\beta)$.

□

Note that Algorithm 1, with $r = 1$, is exactly the HSD algorithm described in Ye et al. [26] and Xu et al. [25]. In the rest of this section and the next section, the notation,

$$x = \begin{pmatrix} x \\ \tau \end{pmatrix} \in R^{n+1}, \quad s = \begin{pmatrix} s \\ \kappa \end{pmatrix} \in R^{n+1}, \quad d_x^{(j)} = \begin{pmatrix} d_x^{(j)} \\ d_\tau^{(j)} \end{pmatrix} \in R^{n+1}, \quad \text{and} \quad d_s^{(j)} = \begin{pmatrix} d_s^{(j)} \\ d_\kappa^{(j)} \end{pmatrix} \in R^{n+1},$$

will be introduced. Note that for Algorithm 1, Lemma 1 becomes

$$(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} = 0 \quad \text{and} \quad (d_x^{(j)})^T d_s^{(j)} = 0, \quad \forall \quad i, j = 1, \ldots, r. \tag{22}$$

Again, let

$$\begin{array}{rcccl} \triangle x & = & x^{k+1} - x^k & = & \sum_{j=1}^r (\alpha^k)^j d_x^{(j)}, \\ \triangle y & = & y^{k+1} - y^k & = & \sum_{j=1}^r (\alpha^k)^j d_y^{(j)}, \\ \triangle s & = & s^{k+1} - s^k & = & \sum_{j=1}^r (\alpha^k)^j d_s^{(j)}, \end{array}$$

where $(\triangle x, \triangle s, \triangle y)$ satisfy

$$X^k \triangle s + S^k \triangle x \quad = \quad \alpha^k \gamma \mu^k e - \alpha^k X^k s^k - \sum_{j=2}^r (\alpha^k)^j (\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}). \tag{23}$$

Thus, relation (15), Lemma 2 and Proposition 3 become

$$(x^k)^T \triangle s + (s^k)^T \triangle x = \alpha^k (\gamma - 1)(x^k)^T s^k,$$

$$\triangle x^T \triangle s = 0,$$

and

$$\mu^{k+1} = [1 - \alpha^k (1 - \gamma)] \mu^k, \quad \theta^{k+1} = [1 - \alpha^k (1 - \gamma)] \theta^k. \tag{24}$$

By relation (24), γ needs to be less than 1 to guarantee that μ^{k+1} is always less than μ^k. The selection of α^k plays a very important role in our complexity analysis. Let us define

$$x(\alpha) = x^k + \sum_{j=1}^r \alpha^j d_x^{(j)}, \quad s(\alpha) = s^k + \sum_{j=1}^r \alpha^j d_s^{(j)}.$$

and

$$g(\alpha) = \min_{i=1,\ldots,(n+1)} \{x_i(\alpha)s_i(\alpha)\} - \beta\frac{x(\alpha)^T s(\alpha)}{n+1} = \min_{i=1,\ldots,(n+1)} \{x_i(\alpha)s_i(\alpha)\} - \beta[1-\alpha(1-\gamma)]\mu^k. \qquad (25)$$

We can choose α^k as

$$\alpha^k = \max_{\alpha\in(0,\frac{1}{1-\gamma})} \{\alpha : \; g(\zeta) \geq 0, \; \forall\, \zeta \leq \alpha\}. \qquad (26)$$

This choice of α^k minimizes $x(\alpha)^T s(\alpha)$ while it guarantees that the new iterate still lies in $\mathcal{N}_\infty^{\sim}(\beta)$.

In general, problem (26) involves locating roots for each of $n+1$ polynomials with degree $2r$, which with specified error is in the complexity class NC and can be solved efficiently in theory (see Pan [18]). In practice, we need to locate only an approximate step-size. (Even for the case $r=1$, one will never obtain the exact α^k since it is generally irrational.) In the next section, we will establish a lower bound for α^k:

$$\alpha^k \geq \frac{const}{(n+1)^{\frac{r+1}{2r}}},$$

where

$$const = \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}.$$

Thus, we need only to compute an approximate step-size, $\bar{\alpha}$, such that

$$(x(\bar{\alpha}), s(\bar{\alpha})) \in \mathcal{N}_\infty^{\sim}(\beta),$$

and

$$\alpha^k - \bar{\alpha} \leq \frac{.001const}{n+1} \leq 0.001\alpha^k,$$

that is, $\bar{\alpha}$ will be at least a fraction, .999, of the exact step-size α^k, and it approaches above α^k as $n \to \infty$.

We may compute such an approximate step-size using the bisection technique. We know that the step-size must be in

$$\left[\frac{const}{(n+1)^{\frac{r+1}{2r}}}, \frac{1}{1-\gamma}\right].$$

Thus, we take α as the middle-point of this candidate interval. Compute α^j, $j=1,\ldots,r$; this will use r arithmetic operations. Then, compute $x(\alpha)$, $s(\alpha)$, and $g(\alpha)$, which are done basically by multiplications of two $n\times r$ matrices with the vector α^j. This arithmetic operation cost is of order $O(nr)$. If $x(\alpha)>0$, $s(\alpha)>0$ and $g(\alpha)\geq 0$, then set the new candidate interval as

$$\left[\alpha, \frac{1}{1-\gamma}\right];$$

otherwise, set it as

$$\left[\frac{const}{n^{\frac{r+1}{2r}}}, \alpha\right].$$

Continue this bisection process until the length of the candidate interval is less than $.001const/(n+1)$. Obviously, the total number of operations of this process is of order $nr(\log n)$. Even when $r=n$, the cost, $n^2(\log n)$, is well below n^3.

4 Complexity Analysis of Algorithm 1

In this section, we first develop several technical results which are needed to establish our main complexity result. Let $D = (S^k)^{1/2}(X^k)^{-1/2}$. Then, relation (22) implies that, for $j \geq 1$,

$$\|D^{-1}d_s^{(j)}\|^2 + \|Dd_x^{(j)}\|^2 = \|D^{-1}d_s^{(j)} + Dd_x^{(j)}\|^2.$$

Define $q^{(j)} = D^{-1}d_s^{(j)} + Dd_x^{(j)}$, $j \geq 1$, then

$$\max(\|Dd_x^{(j)}\|, \|D^{-1}d_s^{(j)}\|) \leq \|q^{(j)}\|. \tag{27}$$

Moreover, relations (19) and (21) show that

$$q^{(j)} = \begin{cases} (X^kS^k)^{-\frac{1}{2}}(\gamma\mu^k e - X^k s^k), & \text{if } j = 1, \\ (X^kS^k)^{-\frac{1}{2}}(-\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}) & \text{if } j \geq 2. \end{cases} \tag{28}$$

And it is easy to verify the following lemma (e.g., see Mizuno et al. [15]).

Lemma 4 .

$$\|q^{(1)}\|^2 \leq \delta(x^k)^T s^k \quad \textit{where} \quad \delta = \gamma^2/\beta - 2\gamma + 1.$$

□

Relations (27) and (28) are now used to prove the following lemma.

Lemma 5 . *For $j = 1, 2, ..., r$,*

$$\|q^{(j)}\| \leq \frac{p(j)}{(\beta\mu^k)^{\frac{j-1}{2}}}\|q^{(1)}\|^j, \tag{29}$$

where the sequence $p(j)$ is defined recursively as follows:

$$\begin{aligned} p(1) &= 1, \\ p(j) &= \textstyle\sum_{t=1}^{j-1} p(t)p(j-t), \quad j = 2, ..., r. \end{aligned}$$

Proof. The proof is by induction on j. Note that (29) holds for $j = 1$. Assume (29) holds for all t with $1 \leq t < j$. We will show that it holds for j.

$$\begin{aligned} \|q^{(j)}\| &= \|(X^kS^k)^{-\frac{1}{2}}(-\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\| \\ &\leq \|(X^kS^k)^{-\frac{1}{2}}\|\|(-\textstyle\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)})\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}}\textstyle\sum_{t=1}^{j-1}\|Dd_x^{(t)}D^{-1}d_s^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}}\textstyle\sum_{t=1}^{j-1}\|Dd_x^{(t)}\|\|D^{-1}d_s^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}}\textstyle\sum_{t=1}^{j-1}\|q^{(t)}\|\|q^{(j-t)}\| \\ &\leq \frac{1}{(\beta\mu^k)^{\frac{1}{2}}}\textstyle\sum_{t=1}^{j-1}(\frac{p(t)}{(\beta\mu^k)^{\frac{t-1}{2}}}\|q^{(1)}\|^t)(\frac{p(j-t)}{(\beta\mu^k)^{\frac{j-t-1}{2}}}\|q^{(1)}\|^{j-t}) \\ &= \frac{1}{(\beta\mu^k)^{\frac{j-1}{2}}}\{\textstyle\sum_{t=1}^{j-1} p(t)p(j-t)\}\|q^{(1)}\|^j \\ &= \frac{p(j)}{(\beta\mu^k)^{\frac{j-1}{2}}}\|q^{(1)}\|^j, \end{aligned}$$

which proves the lemma.

$\square$

Next, we establish a lower bound for α^k.

Lemma 6 . *Let α^k be defined in (26), then*

$$\alpha^k \geq \frac{\xi^k}{(n+1)^{\frac{r+1}{2r}}},$$

where

$$\xi^k = \min_{j=r+1,\ldots,2r} \left(\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \right).$$

Proof. From the definition of α^k and $g(\alpha)$, α^k is the largest number in $(0, \frac{1}{1-\gamma})$ such that for $\alpha \in [0, \alpha^k]$,

$$x_i(\alpha)s_i(\alpha) - \beta[x(\alpha)^T s(\alpha)]/(n+1) \geq 0, \quad \forall \quad i = 1, 2, \ldots, n+1.$$

Using relation (27), Lemmas 4 and 5, and the fact that $x_i^k s_i^k - \beta\mu^k \geq 0$, we derive that, $\forall\ i = 1, \ldots, n+1$,

$$\begin{aligned}
& x_i(\alpha)s_i(\alpha) - \beta[x(\alpha)^T s(\alpha)]/(n+1) \\
= \ & (1-\alpha)(x_i^k s_i^k - \beta\mu^k) + (1-\beta)\alpha\gamma\mu^k + \sum_{j=r+1}^{2r} \left\{ \alpha^j \sum_{t=j-r}^{r} (d_x^{(t)})_i (d_s^{(j-t)})_i \right\} \\
\geq \ & (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \sum_{t=j-r}^{r} \|q^{(t)}\| \|q^{(j-t)}\| \right\} \\
\geq \ & (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \frac{\|q^{(1)}\|^j}{(\beta\mu^k)^{\frac{j-2}{2}}} \sum_{t=j-r}^{r} p(t)p(j-t) \right\} \\
\geq \ & (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ \alpha^j \frac{\|q^{(1)}\|^j}{(\beta\mu^k)^{\frac{j-2}{2}}} \sum_{t=1}^{j-1} p(t)p(j-t) \right\} \\
\geq \ & (1-\beta)\alpha\gamma\mu^k - \sum_{j=r+1}^{2r} \left\{ (n+1)^{\frac{j}{2}} \mu^k \alpha^j \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j) \right\} \\
= \ & \alpha\mu^k \left\{ (1-\beta)\gamma - \sum_{j=r+1}^{2r} \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1} \right\} \\
= \ & \alpha\mu^k h(\alpha),
\end{aligned}$$

where

$$h(\alpha) = (1-\beta)\gamma - \sum_{j=r+1}^{2r} \frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1}.$$

As a result, for $\alpha > 0$, it is clear that $g(\alpha) \geq 0$ if $h(\alpha) \geq 0$.

By substitution, we see that $h(\alpha) \geq 0$ for $\alpha \in [0, \xi^k/(n+1)^{\frac{r+1}{2r}}]$, since for $j = r+1, ..., 2r$, each

$$\frac{\delta^{\frac{j}{2}}}{\beta^{\frac{j-2}{2}}} p(j)(n+1)^{\frac{j}{2}} \alpha^{j-1} \leq (n+1)^{\frac{r+1-j}{2r}} \frac{(1-\beta)\gamma}{r} \leq \frac{(1-\beta)\gamma}{r}.$$

Hence

$$\alpha^k \geq \xi^k/(n+1)^{\frac{r+1}{2r}}.$$

□

Finally, we need to check whether or not ξ^k can be bounded from below. Recall that $\delta = \gamma^2/\beta - 2\gamma + 1$. Assume that $\gamma \leq 1/2$. Then, we have

$$\delta \leq \gamma^2/\beta + (1-2\gamma)/\beta = (1-\gamma)^2/\beta.$$

Also recall that the solution for $p(j)$ in Lemma 5 is given by

$$p(j) = \frac{1}{j}\begin{pmatrix} 2(j-1) \\ j-1 \end{pmatrix} \leq \frac{1}{j}2^{2j-2}.$$

Thus, for $j = r+1, \ldots, 2r$, we have

$$\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \geq \sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma\beta^{\frac{j}{2}}}{rp(j)(1-\gamma)^j}} \geq \frac{\beta}{1-\gamma}\sqrt[j-1]{\frac{(1-\beta)\gamma}{rp(j)(1-\gamma)}} \geq \frac{\beta}{1-\gamma}\sqrt[j-1]{\frac{(1-\beta)\gamma}{\frac{r}{j}2^{2j-2}}}$$

$$\geq \frac{\beta}{4(1-\gamma)}\sqrt[j-1]{\frac{j(1-\beta)\gamma}{r}} \geq \frac{\beta}{4(1-\gamma)}\sqrt[j-1]{(1-\beta)\gamma} \geq \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}.$$

Therefore,

$$\xi^k \geq \frac{\beta}{4(1-\gamma)}\sqrt[r]{(1-\beta)\gamma}. \tag{30}$$

We now present our main complexity result.

Theorem 7 . *Given any (infeasible) initial point in $\mathcal{N}_\infty^{\sim}(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 1, with any constant $0 < \gamma < 1$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+1}{2r}}L)$ iterations where $r \in [1, n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

Proof. We easily see when $1 \leq r \leq n$, the total arithmetic operations in each iteration is $O(n^3)$, since the linear equations in all corrections have the same left-hand matrix. Furthermore, Inequality (30) implies that

$$\mu^{k+1} = \left(1 - \alpha^k(1-\gamma)\right)\mu^k \leq \left(1 - \frac{\xi^k(1-\gamma)}{n^{\frac{r+1}{2r}}}\right)\mu^k \leq \left(1 - \frac{\beta\sqrt[r]{(1-\beta)\gamma}}{4(n+1)^{\frac{r+1}{2r}}}\right)\mu^k,$$

which, together with Proposition 3, prove the theorem.

□

As we can see that if $r = n$ and n increases, the iteration complexity of Algorithm 1 tends to $O(\sqrt{n}L)$ asymptotically. Furthermore, a popular choice for γ in practice is not a constant but $\gamma = O(1/n)$ (e.g., [10]). Interestingly, the asymptotical iteration complexity of Algorithm 1 for such a choice of γ is still $O(\sqrt{n}L)$. More precisely, we have

Corollary 8 . *Given any (infeasible) initial point in $\mathcal{N}_\infty^\sim(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 1, with $\gamma = 1/(n+1)$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+3}{2r}} L)$ iterations where $r \in [1,n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

5 Algorithm 2 and Its Complexity

Another special case is to set

$$\eta^{(j)} = 0, \quad \forall \quad j \geq 1.$$

Therefore $f^{(1)} = \gamma - 1, \quad f^{(j)} = 0, \ \forall \ j \geq 2$, and the algorithm becomes:

1. Solve for the first order direction from

$$\begin{array}{rrrrrcl} & A\ d_x^{(1)} & -b\ d_\tau^{(1)} & & & = & 0 \\ -A^T\ d_y^{(1)} & & +c\ d_\tau^{(1)} & -d_s^{(1)} & & = & 0, \\ b^T\ d_y^{(1)} & -c^T\ d_x^{(1)} & & & -d_\kappa^{(1)} & = & 0. \end{array} \tag{31}$$

   and

$$\begin{array}{rcl} X^k d_s^{(1)} + S^k d_x^{(1)} & = & \gamma\mu^k e - X^k s^k, \\ \tau^k d_\kappa^{(1)} + \kappa^k d_\tau^{(1)} & = & \gamma\mu^k - \tau^k \kappa^k, \end{array} \tag{32}$$

2. For $j = 2, 3, \ldots, l$, solve for the jth order direction from

$$\begin{array}{rrrrrcl} & A\ d_x^{(j)} & -b\ d_\tau^{(j)} & & & = & 0, \\ -A^T\ d_y^{(j)} & & +c\ d_\tau^{(j)} & -d_s^{(j)} & & = & 0, \\ b^T\ d_y^{(j)} & -c^T\ d_x^{(j)} & & & -d_\kappa^{(j)} & = & 0, \end{array} \tag{33}$$

   and

$$\begin{array}{rcl} X^k d_s^{(j)} + S^k d_x^{(j)} & = & -\sum_{t=1}^{j-1} D_x^{(t)} d_s^{(j-t)}, \\ \tau^k d_\kappa^{(j)} + \kappa^k d_\tau^{(j)} & = & -\sum_{t=1}^{j-1} d_\tau^{(t)} d_\kappa^{(j-t)}. \end{array} \tag{34}$$

3. Choose a step-size α^k such that the new iterate,

$$\begin{array}{rcl} x^{k+1} & = & \left(x^k + \sum_{j=1}^r (\alpha^k)^j d_x^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ y^{k+1} & = & \left(y^k + \sum_{j=1}^r (\alpha^k)^j d_y^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ s^{k+1} & = & \left(s^k + \sum_{j=1}^r (\alpha^k)^j d_s^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ \tau^{k+1} & = & \left(\tau^k + \sum_{j=1}^r (\alpha^k)^j d_\tau^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right), \\ \kappa^{k+1} & = & \left(\kappa^k + \sum_{j=1}^r (\alpha^k)^j d_\kappa^{(j)}\right) / \left(1 + \alpha^k(\gamma - 1)\right) \end{array}$$

   lies in $\mathcal{N}_\infty^\sim(\beta)$.

$\square$

As discussed in Xu et al. [24], an interesting feature of Algorithm 2 is that it does not need to compute the infeasibility residuals at all. In this case, we again have

$$(d_x^{(i)})^T d_s^{(j)} + (d_x^{(j)})^T d_s^{(i)} + d_\tau^{(i)} d_\kappa^{(j)} + d_\tau^{(j)} d_\kappa^{(i)} = 0 \quad \text{and} \quad (d_x^{(j)})^T d_s^{(j)} + d_\tau^{(j)} d_\kappa^{(j)} = 0,$$

for all $i, j = 1, \ldots, r$. Relation (15), Lemma 2 and Proposition 3 become

$$(x^k)^T \triangle s + (s^k)^T \triangle x + \tau^k \triangle \kappa + \kappa^k \triangle \tau = \frac{-\alpha^k(\gamma-1)}{1+\alpha^k(\gamma-1)}((x^k)^T s^k + \kappa^k \tau^k),$$

$$\triangle x^T \triangle s + \triangle\tau\triangle\kappa = 0,$$

$$\mu^{k+1} = \frac{1}{1+\alpha^k(\gamma-1)}\mu^k, \quad \theta^{k+1} = \frac{1}{1+\alpha^k(\gamma-1)}\theta^k. \tag{35}$$

From relation (35), that γ is greater than 1 implies that μ^{k+1} is less than μ^k. Similarly, we choose the step-size α^k as

$$\alpha^k = \max_{\alpha>0}\{\alpha : g(\zeta) \geq 0, \forall\, \zeta \leq \alpha\}. \tag{36}$$

where g is defined as in (25).

One can verify that all lemmas in the last two sections hold for Algorithm 2. The only difference is that $\delta \leq \frac{\gamma^2}{\beta}$, since $\gamma > 1$. Therefore, for $j = r+1, \ldots, 2r$,

$$\sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\gamma}{rp(j)\delta^{\frac{j}{2}}}} \geq \sqrt[j-1]{\frac{(1-\beta)\beta^{\frac{j-2}{2}}\beta^{\frac{j}{2}}\gamma}{rp(j)\gamma^j}} \geq \frac{\beta}{4\gamma}\sqrt[j-1]{\frac{(1-\beta)}{rp(j)}} \geq \frac{\beta}{4\gamma}\sqrt[r]{1-\beta},$$

which implies that $\xi^k \geq \frac{\beta}{4\gamma}\sqrt[r]{1-\beta}$. Thus, we have

$$\mu^{k+1} = \frac{1}{1+\alpha^k(\gamma-1)}\mu^k \leq \left(1 - \frac{\beta(\gamma-1)\sqrt[r]{1-\beta}}{4\gamma(n+1)^{\frac{r+1}{2r}} + \beta(\gamma-1)\sqrt[r]{1-\beta}}\right)\mu^k.$$

Therefore, the complexity result for Algorithm 2 can be presented as follows.

Theorem 9 . *Given any (infeasible) initial point in $\mathcal{N}_\infty^{\sim}(\beta)$ for any constant $\beta \in (0,1)$, Algorithm 2, with any $\gamma > 1$, generates a strictly complementary feasible solution for (HLF) in $O(n^{\frac{r+1}{2r}}L)$ iterations where $r \in [1, n]$, and each iteration uses $O(n^3)$ arithmetic operations.*

6 Conclusion

A number of implications and points can be drawn from our main result:

- The high-order power-series method, where iterative points move along a high-order polynomial curve, has been used in practice and partially analyzed in theory. Our result indicates that the use of this method also significantly improves the worst-case iteration complexity. Our result provides a further theoretical base for using this approach.

- From our theoretical result, the order r of power-series has a diminishing role in improving the worst-case complexity result. Thus, we probably expect only the first few order steps really make a difference in algorithm performance. This seems what is observed in practice.

- Our result also provides a justification for using the wider neighborhood $\mathcal{N}_\infty^-$ or $\mathcal{N}_\infty^\sim$, coupled with a high-order method. The theoretical complexity based on wider neighborhoods is not much worse than that based on smaller neighborhoods. We hope this is a significant step to bridge the gap between theory and practice.

- Our result also indicates how insensitive of the value of γ, the centering weight, is in high-order power-series methods. Virtually, γ can be set to any positive number if iterative points move along a high-order polynomial curve. This implies that the method has a sufficient self-centering function even γ is close to 0. Note that, when $\gamma = 0$, the algorithm becomes the pure Newton method.

Acknowledgement. The first author wishes to thank Dr. Xiaojie Xu for his comments and suggestions on this topic.

References

[1] K. M. Anstreicher and R. A. Bosch, "Long steps in an $O(n^3L)$ algorithm for linear programming," *Mathematical Programming* 54 (1992) 251-267.

[2] D. A. Bayer and J. C. Lagarias, "The nonlinear geometry of linear programming: I. Affine and projective scaling trajectories, II. Legendre transform coordinates and central trajectories," *Transactions of the American Mathematical Society* 314 (1989) 499-581.

[3] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno, "High-order predictor-corrector interior point methods with application to quadratic objectives," *SIAM Journal on Optimization* 3 (1993) 696-725.

[4] A. J. Goldman and A. W. Tucker, "Polyhedral convex cones," in: H. W. Kuhn and A. W. Tucker eds., *Linear Inequalities and Related Systems* (Princeton University Press, Princeton, NJ, 1956) pp. 19-40.

[5] C. C. Gonzaga, "An algorithm for solving linear programming programs in $O(n^3L)$ operations," in: N. Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods* (Springer-Verlag, New York, 1988) 1-28.

[6] O. Güler and Y. Ye, "Convergence behavior of interior-point algorithms," *Mathematical Programming* 60 (1993) 215-228.

[7] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatoria* 4 (1984) 373-395.

[8] M. Kojima, S. Mizuno, and A. Yoshise, "A primal-dual interior point algorithm for linear programming," in: N. Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods* (Springer-Verlag, New York, 1989) 29-47.

[9] M. Kojima, S. Mizuno, and A. Yoshise, "A polynomial-time algorithm for a class of linear complementarity problems," *Mathematical Programming* 44 (1989) 1-26.

[10] I. J., Lustig, R. E. Marsten, and D. F. Shanno, "Computational experience with a primal-dual interior point method for linear programming" *Linear Algebra and Its Applications* 152 (1991) 191-222.

[11] N. Megiddo, "Pathways to the optimal set in linear programming," in: N. Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods* (Springer-Verlag, New York, 1988) 131-158.

[12] S. Mehrotra "On the implementation of a (primal-dual) interior point method," *SIAM Journal on Optimization* 2 (1992) 575-601.

[13] R. C. Monteiro and I. Adler, "Interior path following primal-dual algorithms. Part I: Linear programming," *Mathematical Programming* 44 (1989) 27-42.

[14] R. C. Monteiro and 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," *Mathematics of Operations Research* 15 (1990) 191-214.

[15] S. Mizuno, M. J. Todd, and Y. Ye, "On adaptive-step primal-dual interior-point algorithms for linear programming," *Mathematics of Operations Research* 18 (1993) 964-981.

[16] S. Mizuno, M. J. Todd, and Y. Ye, "A surface of analytic centers and infeasible-interior-point algorithms for linear programming," Technical Report, School of Operations Research and Industrial Engineering, Cornell University (Ithaca, New York, 1992), to appear in *Mathematics of Operations Research.*

[17] Yu. Nesterov, "Long steps strategies in interior-point potential reduction methods," Central Economical and Mathematical Institute, Russian Academy of Science (Moscow, Russia, 1993).

[18] V. Pan, "Complexity of computations with matrices and polynomials," *SIAM Review* 34 (1992) 225-262.

[19] F. A. Potra, "On a predictor-corrector method for solving linear programs from infeasible starting points," Technical Report 34, Department of Mathematics, University of Iowa (Iowa City, IA, 1992).

[20] C. Roos and J.-Ph. Vial, "Long steps with the logarithmic penalty barrier function in linear programming," in: J. Gabszeywicz, J. F. Richard, and L. Wolsey, eds., *Economic Decision Making: Games, Economics and Optimization* (Elsevier Science Publisher B. V., Amsterdam, 1989) 433-441.

[21] J. Renegar, "A polynomial-time algorithm based on Newton's method for linear programming," *Mathematical Programming* 40 (1988) 59-94.

[22] G. Sonnevend, "An analytical center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming," in: *Lecture Notes in Control and Information Sciences* 84 (Springer, New York, 1985) 866-876.

[23] R. A. Tapia, Y. Zhang, M. Saltzman, and A. Weiser, "The predictor-corrector interior-point method as a composite Newton method," TR 90-17, Department of Mathematical Sciences, Rice University (Houston, TX, 1990).

[24] X. Xu and Y. Ye, "A generalized homogeneous and self-dual linear programming algorithm," Working Paper, College of Business Administration, The University of Iowa (Iowa City, IA, 1993).

[25] X. Xu, P. F. Hung, and Y. Ye, "A simplified homogeneous and self-dual linear programming algorithm and its implementation," Working Paper, College of Business Administration, The University of Iowa (Iowa City, IA, 1993).

[26] Y. Ye, M. J. Todd, and S. Mizuno, "An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm," *Mathematics of Operations Research* 19 (1994) 53-67.

[27] L. Zhang and Y. Zhang "On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms," Technical Report 93-12, Department of Mathematics and Statistics, University of Maryland at Baltimore County (Baltimore, MD, 1993).