

On the worst case complexity of potential reduction algorithms for linear programming

Dimitris Bertsimas^{a,*}, Xiaodong Luo^b

^a *Sloan School of Management and Operations Research Center, MIT, E53-359, Cambridge, MA 02139, USA*

^b *Operations Research Center, MIT, Cambridge, MA 02139, USA*

Received 13 July 1995; revised manuscript received 16 July 1996

Abstract

There are several classes of interior point algorithms that solve linear programming problems in $O(\sqrt{n}L)$ iterations. Among them, several potential reduction algorithms combine both theoretical ($O(\sqrt{n}L)$ iterations) and practical efficiency as they allow the flexibility of line searches in the potential function, and thus can lead to practical implementations. It is a significant open question whether interior point algorithms can lead to better complexity bounds. In the present paper we give some negative answers to this question for the class of potential reduction algorithms. We show that, even if we allow line searches in the potential function, and even for problems that have network structure, the bound $O(\sqrt{n}L)$ is tight for several potential reduction algorithms, i.e., there is a class of examples with network structure, in which the algorithms need at least $\Omega(\sqrt{n}L)$ iterations to find an optimal solution. © 1997 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Interior point methods; Lower bounds on complexity; Potential reduction algorithms

1. Introduction

In the last decade there has been a lot of excitement about interior point algorithms for linear programming. In his seminal work, Karmarkar [7] presented an $O(nL)$ iterations algorithm with $O(n^3)$ work per iteration, thus resulting in an $O(n^4L)$ algorithm for the linear programming problem:

* Corresponding author. Email: dbertsim@aris.mit.edu. The research of the author was partially supported by a Presidential Young Investigator Award DDM-9158118 with matching funds from Draper Laboratory.

$$\begin{aligned}
 &\text{minimize} && c'x && (1) \\
 &\text{subject to} && Ax = b, \\
 &&& x \geq 0,
 \end{aligned}$$

and its dual

$$\begin{aligned}
 &\text{maximize} && b'y \\
 &\text{subject to} && A'y + s = c, \\
 &&& s \geq 0,
 \end{aligned}$$

where A is an $m \times n$ matrix and L is the input size of the linear program. Moreover, he used rank one updates to obtain an $O(n^{3.5}L)$ algorithm for linear programming. Since then, there have been several path following algorithms for linear programming that need $O(\sqrt{n}L)$ iterations (Renegar [12], Gonzaga [4]) with time complexity $O(n^3L)$. The drawback of the latter approaches is that, although they achieve the best known worst case bound, they are restricted to make small steps by following the central trajectory. As a result, these path following methods are not attractive for a practical implementation.

A major step in the direction of practical algorithms, still achieving the best known worst case bounds, was taken by Ye [18], in which he presented a potential reduction primal–dual algorithm that solves linear programming in $O(\sqrt{n}L)$ iterations and $O(n^3L)$ time. Freund [3] simplified the algorithm by Ye and showed that certain choices for the parameters in the algorithm are in a sense best possible. Gonzaga and Todd [5] proposed another potential reduction algorithm, which also solves linear programming in $O(\sqrt{n}L)$ iterations. The major advantage of this class of algorithms is that they are conceptually elegant and natural, achieve the best known worst case bounds, and yet they lead to practical implementations through the use of line searches. In the theory and practice of linear programming this is a rare combination. The simplex method and affine scaling algorithm, for example, are practical algorithms, but are not known to be polynomial algorithms.

Since a variety of interior point algorithms solve linear programming problems in $O(\sqrt{n}L)$ iterations, one can naturally ask:

1. Can the bound $O(\sqrt{n}L)$ be improved for any interior point algorithm, i.e., is the bound inherent in the algorithm or is it because of the weakness of our proof methods?
2. Do line searches improve the theoretical complexity of interior point algorithms?
3. Can the bound be improved for linear programs of network structure?
4. More ambitiously, is the bound $O(\sqrt{n}L)$ inherent in the linear programming problem? We remark that this question is rather deep as it is related to the existence of a strongly polynomial algorithm for linear programming.

Related to the first two questions, it is well known that short step path following algorithms need $\Theta(\sqrt{n}L)$ iterations for all linear programs. Anstreicher [1] (see also McDiarmid [9]) showed that in the worst-case, there is a step among all steps of

Karmarkar's algorithm such that the reduction of the potential function can not be greater than a constant. Anstreicher [2] sharpened this observation and showed that with exact line searches, there is a class of linear programs, in which Karmarkar's algorithm can only obtain a decrease of the potential function that is no more than a constant for every iteration; he further showed that $\Theta(\log nL)$ iterations are actually needed to solve this example. Kaliski and Ye [6] showed that Karmarkar's algorithm requires at least $\Omega(n)$ iterations to converge for solving a linear program with only one constraint. Powell [11] also showed a lower bound of $\Omega(n)$ for problems that are natural discretizations of a semi-infinite programming problem. Todd [14] (see also Todd and Ye [16]) showed a lower bound of $\Omega(n^{\frac{1}{2}})$ on the number of steps needed by certain interior point primal–dual methods to reduce the duality gap by a constant at the start of the algorithm.

In the present paper we give negative answers to the first three questions for the potential reduction algorithms of Ye [18], Freund [3], and Gonzaga and Todd [5]. Potential function algorithms are characterized by a parameter q (see next section). We give tight bounds for the number of iterations for $q = n + O(\sqrt{n})$, even if we use line searches and even if the linear program has network structure.

The paper is structured as follows. In Section 2 we summarize the potential reduction algorithm and the known complexity results. In Section 3 we exhibit an example that has network structure and requires $\Theta(\sqrt{nL})$ iterations even if we allow line searches. The final section contains some concluding remarks.

We briefly describe our notation below. All vectors are column vectors and the superscript $'$ denotes transpose. Unless otherwise specified, $\|\cdot\|$ denotes the usual Euclidean norm, $\log(\cdot)$ denotes the natural logarithmic function. For any $x \in \mathbb{R}^n$, we denote by x_i the i th coordinate of x , and when describing an algorithm, we use the notation x^k to denote the value of variable x evaluated at the k th iteration of the algorithm.

2. The potential reduction algorithm

For every linear programming problem, Todd and Ye [15] and Tanabe [13] introduced the primal–dual logarithmic barrier function:

$$G(x, s) = q \log x' s - \sum_{j=1}^n \log x_j - \sum_{j=1}^n \log s_j, \quad (2)$$

where the first term is a measure of the duality gap $x' s$ and the two other terms are the barrier terms. The goal of the algorithm proposed by Ye [18] is to decrease the duality gap below a tolerance ε by decreasing the potential function $G(x, s)$ at every iteration. We first give the following definition.

Definition 1. A primal and dual solution pair (x, s) is called ε -optimal if the duality gap satisfies $x' s \leq \varepsilon$.

Let L be the input size of the linear program (1). Note that if an ε -optimal solution is available with $\varepsilon = 2^{-2L}$, then an exact solution can be found in polynomial time through rounding. For this class of potential functions the following result holds:

Theorem 1 (Ye [18]). *An algorithm that reduces $G(x, s)$ by an amount greater or equal to $\delta > 0$ at each iteration finds an ε -optimal solution (\bar{x}, \bar{s}) in*

$$O\left(\frac{q-n}{\delta} \log \frac{1}{\varepsilon}\right) \text{ iterations,}$$

as long as the initial feasible primal–dual values (x^0, s^0) satisfy

$$G(x^0, s^0) = O\left(\frac{q-n}{\delta} \log \frac{1}{\varepsilon}\right).$$

We note that the previous theorem focuses in finding an ε -optimal solution. If we want an exact solution we need to replace $\log(1/\varepsilon)$ by L . Let (A, b, c) be the data for the linear program (1); let $x^0 > 0, s^0 > 0, y^0$ be the initial primal and dual solutions respectively; let $\varepsilon > 0$ be the optimality tolerance. Let X denote a diagonal matrix with (x_1, x_2, \dots, x_n) in the diagonal. Let e_n be an n -dimensional column vector of all ones. Then Ye’s potential function algorithm, which uses the parameters (γ, α, q) ($0 < \gamma < 1, \alpha \geq 0, q \geq n$) can be described as follows:

Algorithm \mathcal{A} $(A, b, c, x^0, s^0, y^0, \varepsilon)$.

$k = 0$;

while $(x^k)'s^k > \varepsilon$ **do**

$$X^k = \text{diag}(x^k);$$

$$\bar{A}^k = (AX^k)'(AX^kX^kA')^{-1}AX^k;$$

$$u^k = (I - \bar{A}^k) \left(\frac{q}{(x^k)'s^k} X^k s^k - e_n \right);$$

if $\|u^k\| \geq \gamma$

then (*primal step*)

$$x^{k+1} = x^k - \alpha \frac{X^k u^k}{\|u^k\|};$$

$$s^{k+1} = s^k;$$

else (*dual step*)

$$x^{k+1} = x^k;$$

$$s^{k+1} = \frac{(x^k)'s^k}{q} (X^k)^{-1} (u^k + e_n);$$

end if

$k = k + 1$;

end while

If $\|u^k\| \geq \gamma$ we say that the algorithm performs a *primal step*, while if $\|u^k\| < \gamma$, we say that the algorithm performs a *dual step*.

The algorithm proposed by Freund [3] is somewhat different in the dual step: It finds a scalar $\Delta^k \in (0, (x^k)'s^k]$ such that

$$\left\| (I - \bar{A}^k) \left(\frac{q}{\Delta^k} X^k s^k - e_n \right) \right\| = \gamma$$

and then updates the dual slacks as follows:

$$\begin{aligned} \bar{u}^k &= (I - \bar{A}^k) \left(\frac{q}{\Delta^k} X^k s^k - e_n \right), \\ s^{k+1} &= \frac{(x^k)'s^k}{q} (X^k)^{-1} (\bar{u}^k + e_n). \end{aligned}$$

The algorithm proposed by Gonzaga and Todd's [5] uses exactly the same primal step, while the dual step is more symmetric than both Ye's and Freund's algorithms as it uses projected gradients of the potential function both in the primal and in the dual space. To be precise the dual step is:

$$\begin{aligned} S^k &= \text{diag}(s^k); \\ \bar{A}^k &= (AS^k)'(AS^k S^k A')^{-1} AS^k; \\ v^k &= (I - \bar{A}^k) \left(\frac{q}{(x^k)'s^k} X^k s^k - e_n \right); \end{aligned}$$

If $\|v^k\| \geq \gamma$, then

$$x^{k+1} = x^k; \quad s^{k+1} = x^k - \beta \frac{S^k v^k}{\|v^k\|}.$$

The reason why potential reduction algorithms are attractive from a practical point of view is that we can improve the performance of all these algorithms by using line searches in the potential function. By varying α we can maximize the decrease of the potential function at each step of the algorithm.

For Algorithm \mathcal{A} the following theorem holds:

Theorem 2 (Freund [3], Ye [18]).

1. If Algorithm \mathcal{A} executes a primal step at step k and the step size is chosen to be α , then

$$G(x^{k+1}, s^{k+1}) - G(x^k, s^k) \leq -\alpha\gamma + \frac{\alpha^2}{2(1-\alpha)}.$$

If the algorithm executes a dual step at step k then

$$G(x^{k+1}, s^{k+1}) - G(x^k, s^k) \leq -\frac{q-n}{q}(q-n-\gamma\sqrt{n}) + \frac{2\gamma^2}{(1-\gamma)(1-3\gamma)}.$$

2. If $q = n + \sqrt{n}$, $\gamma = 0.22$, $\alpha = \frac{1}{6}$, then Algorithm \mathcal{A} reduces $G(x, s)$ by a constant amount $\delta = 0.02$ at each iteration, and thus finds an ε -optimal solution in $O(\sqrt{n} \log(1/\varepsilon))$ iterations, provided that $G(x^0, s^0) = O(\sqrt{n} \log(1/\varepsilon))$.

From Theorems 1 and 2 it follows that Algorithm \mathcal{A} finds an ε -optimal solution in $O(\sqrt{n} \log(1/\varepsilon))$ iterations. In the next section we show that even if we use exact line searches this bound is tight.

3. Bounds on the number of primal steps of Algorithm \mathcal{A} with line searches

In this section, we show that Algorithm \mathcal{A} needs at least $\Omega((\sqrt{n}/\nu) \log(1/\varepsilon))$ primal steps to find an ε -optimal solution for $q = n + \nu\sqrt{n}$ and $\nu = o(\sqrt{n})$. The reasons this is a strong lower bound are as follows: (a) Since the work per iteration in a primal step (it involves the solution of a linear system) dominates the work per iteration in a dual step, our lower bound on the number of primal iterations implies a lower bound on the time complexity; (b) The example we construct is a network problem, which implies that the network structure does not help in improving the complexity of potential reduction algorithms; (c) The algorithm uses exact line searches in the potential function.

We analyze the following example:

$$(P) \quad \begin{aligned} & \text{minimize} && \sum_{i=1}^{2n} x_i \\ & \text{subject to} && x_i = x_{i+1}, \quad \text{for } i \neq n, 2n, \\ & && x \geq 0, \end{aligned}$$

and its dual

$$(D) \quad \begin{aligned} & \text{maximize} && 0 \\ & \text{subject to} && y_i + s_i = 1, \quad \text{for } i \in \{1, n+1\}, \\ & && -y_i + s_i = 1, \quad \text{for } i \in \{n, 2n\}, \\ & && y_i - y_{i-1} + s_i = 1, \quad \text{for } i \notin \{1, n, n+1, 2n\}, \\ & && s \geq 0. \end{aligned}$$

Interestingly, problem (P) has network structure. Problem (P) can be written in the standard form (1) with $b = 0$, $c = e_{2n}$ and A the $2(n-1) \times 2n$ matrix

$$A = \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix},$$

where B is the $(n-1) \times n$ matrix $B = (I_{n-1}, 0) - (0, I_{n-1})$.

Notice that for this particular example, the optimal value is 0, all the feasible points of (D) are also dual optimal and the unique primal optimal solution is the zero vector. In this section, we analyze the complexity of Algorithm \mathcal{A} when we allow line searches, and show that there exists a starting point such that Algorithm \mathcal{A} needs $\Omega((\sqrt{n}/\nu) \log(1/\varepsilon))$ primal steps to find an ε -optimal solution for (P) where $q = 2n + \nu\sqrt{n}$ and $\nu = o(\sqrt{n})$.

In order to simplify the notation, we will replace $\alpha/\|u^k\|$ by α in the description of the Algorithm \mathcal{A} , and will suppress the superscript k on α that indicates the current number of iterations.

Obviously the primal iterates have $x_i^k = r^k$, for $i \leq n$ and $x_i^k = t^k$ for $i \geq n + 1$, where r^k and t^k are positive constants. Moreover, since $b = 0$, the duality gap is $c'x^k = n(r^k + t^k)$.

Let $q = 2n + \nu\sqrt{n}$, $\nu > 0$ (note that (P) is a problem with $2n$ variables). We also introduce the notation $\bar{\nu} = \nu/\sqrt{n}$. We first show the following preliminary properties.

Proposition 1. 1. \bar{A}^k is independent of k and

$$\bar{A}^k = \begin{pmatrix} I_n - \frac{1}{n}e_n e_n' & 0 \\ 0 & I_n - \frac{1}{n}e_n e_n' \end{pmatrix}. \tag{3}$$

2. At every iteration, $u_i^k = u_{i+1}^k, \forall i \neq n, 2n$ with

$$u_i^k = \begin{cases} \frac{(2 + \bar{\nu})r^k}{r^k + t^k} - 1, & i \leq n, \\ \frac{(2 + \bar{\nu})t^k}{r^k + t^k} - 1, & i \geq n + 1. \end{cases} \tag{4}$$

Proof.

$$\begin{aligned} \bar{A}^k &= (AX^k)'(AX^k X^k A')^{-1} AX^k \\ &= \begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix}' \left[\begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix} \begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix}' \right]^{-1} \begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix} \\ &= \begin{pmatrix} r^k B' & 0 \\ 0 & t^k B' \end{pmatrix} \begin{pmatrix} (r^k)^2 BB' & 0 \\ 0 & (t^k)^2 BB' \end{pmatrix}^{-1} \begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix} \\ &= \begin{pmatrix} r^k B' & 0 \\ 0 & t^k B' \end{pmatrix} \begin{pmatrix} (r^k)^{-2} (BB')^{-1} & 0 \\ 0 & (t^k)^{-2} (BB')^{-1} \end{pmatrix} \begin{pmatrix} r^k B & 0 \\ 0 & t^k B \end{pmatrix} \\ &= \begin{pmatrix} U^k & 0 \\ 0 & U^k \end{pmatrix}, \end{aligned}$$

where $U^k = B'(BB')^{-1}B$. Since $I_n - U^k = (1/n)e_n e_n'$, (3) follows. Observe that \bar{A}^k is independent of k . Since $b = 0$, the duality gap is $c'x^k$ and therefore, by (3),

$$u^k = (I - \bar{A}^k) \begin{pmatrix} q \\ (x^k)'_c X^k c - e_{2n} \end{pmatrix} = \frac{1}{n} \begin{pmatrix} e_n e_n' & 0 \\ 0 & e_n e_n' \end{pmatrix} \begin{pmatrix} q \\ (x^k)'_c X^k c - e_{2n} \end{pmatrix}.$$

For $i \leq n$,

$$u_i^k = \frac{1}{n} \cdot \frac{q \sum_{i=1}^n x_i^k}{(x^k)'c} - 1 = \frac{1}{n} \cdot \frac{(2n + \nu\sqrt{n}) \cdot n \cdot r^k}{n \cdot (r^k + t^k)} - 1 = \frac{(2 + \bar{\nu})r^k}{r^k + t^k} - 1.$$

Similarly, for $i \geq n + 1$,

$$u_i^k = \frac{(2 + \bar{\nu})t^k}{r^k + t^k} - 1. \quad \square$$

We now make an important observation, on which we will base our analysis.

Proposition 2. *If at every primal step, an exact line search is used, the step size will only depend on the ratio r^k/t^k , i.e., there exists a function $g : \mathbb{R} \mapsto \mathbb{R}$, such that $\alpha = g(r^k/t^k)$. Furthermore,*

$$g(r^k/t^k) = g(t^k/r^k). \tag{5}$$

Proof. The potential function can be written as

$$\begin{aligned} G(x^{k+1}, s^{k+1}) &= q \log(x^{k+1})'s^{k+1} - \sum_{j=1}^{2n} \log x_j^{k+1} - \sum_{j=1}^{2n} \log s_j^{k+1} \\ &= q \log(n(r^{k+1} + t^{k+1})) - n \log r^{k+1} - n \log t^{k+1} + M_1 \\ &= q \log(r^{k+1} + t^{k+1}) - n \log r^{k+1} - n \log t^{k+1} + M_2 \\ &= q \log(r^k + t^k - \alpha(u_1^k r^k + u_{n+1}^k t^k)) \\ &\quad - n \log(1 - \alpha u_1^k) r^k - n \log(1 - \alpha u_{n+1}^k) t^k + M_2 \\ &= q \log \left(1 + \frac{r^k}{t^k} - \alpha \left(u_1^k \frac{r^k}{t^k} + u_{n+1}^k \right) \right) \\ &\quad - n \log(1 - \alpha u_1^k) - n \log(1 - \alpha u_{n+1}^k) + M_3, \end{aligned} \tag{6}$$

where M_1, M_2 and M_3 are constants independent of α . From (4), u_i^k depends on the ratio r^k/t^k . From (6), we observe that in order to minimize $G(x^{k+1}, s^{k+1})$, the choice of the step size α will depend on the ratio r^k/t^k , but not on the actual magnitude of r^k and t^k , i.e., we can always scale r^k and t^k by the same amount without affecting the minimizers of (6).

The potential function can be written as a function of α as follows:

$$G(\alpha)/n = (2 + \bar{\nu}) \log(1 - \alpha M_4) - \log(1 - \alpha M_5) - \log(1 - \alpha M_6) + M_7,$$

with M_4, M_5, M_6, M_7 constants independent of α . Then

$$G'(\alpha)/n = -(2 + \bar{\nu}) \frac{M_4}{1 - \alpha M_4} + \frac{M_5}{1 - \alpha M_5} + \frac{M_6}{1 - \alpha M_6},$$

and

$$G''(\alpha)/n = -(2 + \bar{\nu}) \left(\frac{M_4}{1 - \alpha M_4} \right)^2 + \left(\frac{M_5}{1 - \alpha M_5} \right)^2 + \left(\frac{M_6}{1 - \alpha M_6} \right)^2.$$

An optimizer of $G(\alpha)$ satisfies $G'(\alpha^*) = 0$. Therefore,

$$\begin{aligned} G''(\alpha^*)/n &= -\frac{1}{2+\bar{\nu}} \left(\frac{M_5}{1-\alpha^*M_5} + \frac{M_6}{1-\alpha^*M_6} \right)^2 + \left(\frac{M_5}{1-\alpha^*M_5} \right)^2 + \left(\frac{M_6}{1-\alpha^*M_6} \right)^2 \\ &= \frac{\bar{\nu}}{2+\bar{\nu}} \left(\left(\frac{M_5}{1-\alpha^*M_5} \right)^2 + \left(\frac{M_6}{1-\alpha^*M_6} \right)^2 \right) + \frac{1}{2+\bar{\nu}} \left(\frac{M_5}{1-\alpha^*M_5} - \frac{M_6}{1-\alpha^*M_6} \right)^2 \\ &> 0. \end{aligned}$$

Since all solutions of $G'(\alpha^*) = 0$ satisfy $G''(\alpha^*) > 0$, there exists at most one minimizer $\alpha^* > 0$, which minimizes $G(\alpha)$. Since we compute a solution α^* explicitly in Proposition 3, there exists a unique minimizer. Therefore, there exists a function $g : \mathbb{R} \mapsto \mathbb{R}$, such that $\alpha^* = g(r^k/t^k)$ minimizes $G(\alpha)$ in (6).

Notice that the potential function $G(x^{k+1}, s^{k+1})$ can be written as $f(r^k, t^k, \alpha)$ for some real function $f(\cdot, \cdot, \cdot)$. Notice, however, that $f(r^k, t^k, \alpha) = f(t^k, r^k, \alpha)$. Moreover, as we have shown before, $f(r^k, t^k, \alpha) = f_1(r^k/t^k, \alpha)$ for some function f_1 and $f(t^k, r^k, \alpha) = f_1(t^k/r^k, \alpha)$. So, $f_1(r^k/t^k, \alpha) = f_1(t^k/r^k, \alpha)$. Therefore, $g(r^k/t^k) = g(t^k/r^k)$, establishing (5). \square

Our strategy to bound the number of primal steps is based on the previous proposition as follows: We select a particular starting point (r^0, t^0) with $r^0/t^0 = 1 + \omega^*$. We will choose ω^* , such that the minimizer

$$\alpha^* = \arg \min_{\alpha > 0} G(x^0 - \alpha X^0 u^0, s^0),$$

satisfies further that $r^0/t^0 = t^1/r^1$. Because of the previous lemma the new minimizer $\alpha^1 = g(t^1/r^1) = g(r^0/t^0) = \alpha^*$. Because of the symmetry of the particular example, we will show that $r^1/t^1 = t^2/r^2$, which means that the new minimizer α^2 will also be the same as α^* , and so on. Therefore, by selecting carefully the starting point (r^0, t^0) , we can force the algorithm to pick the same minimizer in the line search of the potential function at every primal step.

Let $x_1^0 = r^0 = 1 + \omega$, $x_{n+1}^0 = t^0 = 1$, $s^0 = e$ and $y^0 = 0$ be the initial interior solutions of (P) and (D).

We consider the following system of equations in (α, ω)

$$r^0/t^0 = t^1/r^1, \tag{7}$$

$$\alpha = \arg \min_{\alpha > 0} G(x^0 - \alpha X^0 u^0, s^0). \tag{8}$$

We will first show that the nonlinear system of equations (7), (8) has a solution for (α, ω) , which we compute explicitly in Proposition 3. We then show that we can bound the number of primal steps for the Algorithm A to find an ε -optimal solution.

Proposition 3. *The solution*

$$\omega^* = \bar{\nu}, \quad \alpha^* = \frac{2 + \bar{\nu}}{(1 + \bar{\nu})^2},$$

with $\bar{\nu} = \nu/\sqrt{n}$ satisfies the system (7), (8).

Proof. Since $r^1 = r^0(1 - \alpha u_1^0)$ and $t^1 = t^0(1 - \alpha u_{n+1}^0)$, where u_1^0 and u_{n+1}^0 are given in Eq. (4), we obtain that Eq. (7) is equivalent to

$$\alpha = \frac{(2 + \omega)^2}{(1 + \bar{\nu})[(1 + \omega)^2 + 1] + \bar{\nu}(1 + \omega)}. \tag{9}$$

From the proof of Proposition 2 the function $G(\alpha) = G(x^0 - \alpha X^0 u^0, s^0)$ has at most one minimizer that satisfies $G'(\alpha) = 0$.

Dropping the superscript 0 for ease of exposition we obtain that

$$G'(\alpha)/n = -(2 + \bar{\nu}) \frac{\Delta r + \Delta t}{r + t - \alpha(\Delta r + \Delta t)} + \frac{\Delta r}{r - \alpha \Delta r} + \frac{\Delta t}{t - \alpha \Delta t}, \tag{10}$$

where we define $\Delta r = u_1 r$ and $\Delta t = u_{n+1} t$.

From (4) we obtain

$$\frac{\Delta r}{r} = u_1 = \frac{(2 + \bar{\nu})r}{r + t} - 1 = \frac{(2 + \bar{\nu})\frac{r}{t}}{1 + \frac{r}{t}} - 1 = \frac{(2 + \bar{\nu})(1 + \omega)}{2 + \omega} - 1. \tag{11}$$

Similarly,

$$\frac{\Delta t}{t} = u_{n+1} = \frac{(2 + \bar{\nu})t}{r + t} - 1 = \frac{2 + \bar{\nu}}{1 + \frac{r}{t}} - 1 = \frac{2 + \bar{\nu}}{2 + \omega} - 1. \tag{12}$$

Since, we have defined $r/t = 1 + \omega$, we substitute Eqs. (11) and (12) into Eq. (10) and we obtain that Eq. (8) is equivalent to

$$\begin{aligned} & -(2 + \bar{\nu}) \frac{\left[\frac{(2 + \bar{\nu})(1 + \omega)}{2 + \omega} - 1 \right] (1 + \omega) + \frac{2 + \bar{\nu}}{2 + \omega} - 1}{2 + \omega - \alpha \left[\left(\frac{(2 + \bar{\nu})(1 + \omega)}{2 + \omega} - 1 \right) (1 + \omega) + \frac{2 + \bar{\nu}}{2 + \omega} - 1 \right]} \\ & + \frac{\frac{(2 + \bar{\nu})(1 + \omega)}{2 + \omega} - 1}{1 - \alpha \left[\frac{(2 + \bar{\nu})(1 + \omega)}{2 + \omega} - 1 \right]} + \frac{\frac{2 + \bar{\nu}}{2 + \omega} - 1}{1 - \alpha \left[\frac{2 + \bar{\nu}}{2 + \omega} - 1 \right]} = 0. \end{aligned} \tag{13}$$

By direct substitution it is easy to check that the solution

$$\omega^* = \bar{\nu} \quad \text{and} \quad \alpha^* = \frac{2 + \bar{\nu}}{(1 + \bar{\nu})^2}$$

satisfies the system of equations (9) and (13), which is equivalent to the original system (7), (8). \square

We next prove our main result.

Theorem 3. *Algorithm A implemented with exact line searches in the potential function needs at least*

$$\frac{\sqrt{n}}{\nu} \log \frac{\nu\sqrt{n} + 2n}{\varepsilon} = \Omega\left(\frac{\sqrt{n}}{\nu} \log \frac{1}{\varepsilon}\right)$$

primal steps to find an ε -optimal solution to the linear program (P), when started from the solution x^0, s^0, y^0 .

Proof. We first show by induction on k that for all primal steps, $\alpha^k = \alpha^*$ and $r^k/t^k = t^{k+1}/r^{k+1}$. For $k = 0$ the induction hypothesis corresponds exactly to the condition that the system (7), (8) has a solution (Proposition 3).

Assume that the induction hypothesis holds for $k - 1$. In Proposition 2 we showed that $\alpha^k = g(t^k/r^k)$. By the induction hypothesis $r^{k-1}/t^{k-1} = t^k/r^k$. Therefore, $g(t^k/r^k) = g(r^{k-1}/t^{k-1})$. From Proposition 2, $\alpha^{k-1} = g(r^{k-1}/t^{k-1})$. Therefore, $\alpha^k = \alpha^{k-1} = \alpha^*$ from the induction hypothesis.

Moreover, $r^{k+1} = r^k(1 - \alpha^k u_1^k)$, and $t^{k+1} = t^k(1 - \alpha^k u_{n+1}^k)$. From (4) we obtain

$$\begin{aligned} r^{k+1} &= r^k \left(1 - \alpha^k \left(\frac{(2 + \bar{\nu})r^k}{r^k + t^k} - 1 \right) \right), \\ t^{k+1} &= t^k \left(1 - \alpha^k \left(\frac{(2 + \bar{\nu})t^k}{r^k + t^k} - 1 \right) \right). \end{aligned}$$

Since $r^{k-1}/t^{k-1} = t^k/r^k$ and we have shown that $\alpha^k = \alpha^* = \alpha^{k-1}$ we obtain that

$$\begin{aligned} r^{k+1} &= r^k \left(1 - \alpha^{k-1} \left(\frac{(2 + \bar{\nu})t^{k-1}}{r^{k-1} + t^{k-1}} - 1 \right) \right), \\ t^{k+1} &= t^k \left(1 - \alpha^{k-1} \left(\frac{(2 + \bar{\nu})r^{k-1}}{r^{k-1} + t^{k-1}} - 1 \right) \right). \end{aligned}$$

Therefore,

$$r^{k+1} = r^k \frac{t^k}{t^{k-1}} \quad \text{and} \quad t^{k+1} = t^k \frac{r^k}{r^{k-1}},$$

and hence

$$\frac{r^{k+1}}{t^{k+1}} = \frac{r^{k-1}}{t^{k-1}} = \frac{t^k}{r^k},$$

where the last equation follows from the induction hypothesis. Hence, the induction is complete.

Therefore, $r^k/t^k = 1 + \omega^*$, if k is even and $t^k/r^k = 1 + \omega^*$, if k is odd. We next examine the reduction in the duality gap at the k th primal step. Assuming that k is even (the case k odd is the same), we obtain

$$r^{k+1} = r^k \left(1 - \alpha^k \left(\frac{(2 + \bar{\nu})r^k}{r^k + t^k} - 1 \right) \right) = r^k \left(1 - \alpha^* \left(\frac{(2 + \bar{\nu})(1 + \omega^*)}{2 + \omega^*} - 1 \right) \right) \\ = r^k \frac{1}{(1 + \bar{\nu})^2}.$$

Similarly,

$$t^{k+1} = t^k \left(1 - \alpha^* \frac{\bar{\nu} - \omega^*}{2 + \omega^*} \right) = t^k.$$

Note that the dual steps do not influence the duality gap and therefore, we need only to bound the reduction in the duality gap after a primal step. Then

$$\frac{(x^{k+1})'s^{k+1}}{(x^k)'s^k} = \frac{r^{k+1} + t^{k+1}}{r^k + t^k} = \frac{1}{1 + \bar{\nu}}.$$

Since the initial duality gap is $n(r^0 + t^0) = 2n + \nu\sqrt{n}$, in order for the duality gap to be smaller than ϵ , we need at least K primal steps such that

$$(2n + \nu\sqrt{n}) \left(\frac{1}{1 + \nu/\sqrt{n}} \right)^K < \epsilon,$$

leading to

$$K > \frac{\sqrt{n}}{\nu} \log \frac{\nu\sqrt{n} + 2n}{\epsilon}. \quad \square$$

Notice that for $\nu = \Theta(1/\sqrt{n})$, which means that $q = 2n + \Theta(1)$ we obtain that Algorithm \mathcal{A} needs $\Omega(n \log(1/\epsilon))$ primal steps to find an ϵ -optimal solution to the linear program (P) . For $\nu = \Theta(1)$, i.e., $q = 2n + \Theta(\sqrt{n})$ we obtain the bound $\Omega(\sqrt{n} \log(1/\epsilon))$ primal iterations. Finally, as ν increases the bound becomes weaker. For example, for $\nu = \Theta(\log n)$, i.e., $q = 2n + \Theta(\log n \sqrt{n})$ the bound becomes $\Omega((\sqrt{n}/\log n) \log(1/\epsilon))$ primal steps.

We remark that the derived lower bounds can be extended to the algorithms in [3] and [5]. However, our example may not establish the worst case lower bound for the potential reduction algorithms proposed in [8,10,17,19].

4. Concluding remarks

The example we provided shows that exact line searches for a class of potential reduction algorithms do not help in decreasing the number of iterations even for linear programs with network structure. We have left open, however, the case of $q = n + \nu\sqrt{n}$, with $\nu = \Omega(\sqrt{n})$.

All known proofs of polynomial number of iterations for interior point algorithms have some relation with the logarithmic barrier function. Algorithm \mathcal{A} uses the logarithmic barrier function to design the algorithm directly, while many other algorithms use logarithmic barrier functions to bound the number of iterations. It would be interesting to prove a general lower bound for all potential reduction algorithms.

Acknowledgements

We would like to thank a reviewer of the paper for suggesting a substantial simplification in our proof of Proposition 3. We would also like to thank Kurt Anstreicher, Mike Todd and Yinyue Ye for their comments on an earlier version of the paper. The first author would also like to thank his colleague Robert Freund for stimulating his interest in interior point algorithms over the years.

References

- [1] K.M. Anstreicher, The worst-case step in Karmarkar's algorithm, *Mathematics of Operations Research* 14 (1989) 295–302.
- [2] K.M. Anstreicher, On the performance of Karmarkar's algorithm over a sequence of iterations, *SIAM Journal on Optimization* 1 (1991) 22–29.
- [3] R. Freund, Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function, *Mathematical Programming* 51 (1991) 203–222.
- [4] C.C. Gonzaga, An algorithm for solving linear programming problems in $O(n^3L)$ operations, in: N. Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods* (Springer, New York, 1988) 1–28.
- [5] C.C. Gonzaga and M.J. Todd, An $O(\sqrt{n}L)$ -iteration large-step primal–dual affine algorithm for linear programming, *SIAM Journal on Optimization* 2 (1992) 349–359.
- [6] J. Kaliski and Y. Ye, Convergence behavior of Karmarkar's projective algorithm for solving a simple linear program, *Operations Research Letters* 10 (1991) 389–393.
- [7] N. Karmarkar (1984), A new polynomial-time algorithm for linear programming, *Combinatorica* 4 (1991) 373–395.
- [8] M. Kojima, S. Mizuno and A. Yoshise, An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems, *Mathematical Programming* 50 (1991) 331–342.
- [9] C. McDiarmid, On the improvement per iteration in Karmarkar's algorithm for linear programming, *Mathematical Programming*, 46 (1990) 299–320.
- [10] Y. Nesterov, Long-step strategies in interior-point potential reduction methods, Department SES COMIN, Geneva University, 1993.
- [11] M.J.D. Powell, On the number of iterations of Karmarkar's algorithm for linear programming, *Mathematical Programming* 62 (1993) 153–197.
- [12] J. Renegar, A polynomial-time algorithm, based on Newton's method, for linear programming, *Mathematical Programming* 40 (1988) 59–93.
- [13] K. Tanabe, Centered Newton method for mathematical programming, in: M. Iri and K. Yajima, eds., *System Modeling and Optimization: Proceedings of the 13th IFIP Conference*, Tokyo, Japan, Aug.–Sept. 1987, Lecture Notes in Control and Information Sciences 113 (Springer, Berlin, 1988) 197–206.
- [14] M.J. Todd, A lower bound on the number of iterations of primal–dual interior-point methods for linear programming, in: G.A. Watson and D.F. Griffiths, eds., *Numerical Analysis 1993*, Pitman Research Notes in Mathematics 303 (Longman Press, Burnt Hill, UK, 1994) 237–259.
- [15] M.J. Todd and Y. Ye, A centered projective algorithm for linear programming, *Mathematics of Operations Research* 15 (1990) 508–529.
- [16] M.J. Todd and Y. Ye, A lower bound on the number of iterations of long-step and polynomial interior-point programming algorithms, to appear in *Annals of Operations Research* (1995).
- [17] Y. Ye, A class of projective transformations for linear programming, *SIAM Journal on Computing* 19 (1990) 457–466.
- [18] Y. Ye, An $O(n^3L)$ potential reduction algorithm for linear programming, *Mathematical Programming* 50 (1991) 239–258.
- [19] Y. Ye, K. Kortanek, J. Kaliski and S. Huang, Near boundary behavior of the primal–dual potential reduction algorithm for linear programming, *Mathematical Programming* 58 (1993) 243–255.