SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK 14853

TECHNICAL REPORT NO. 1032.

September 1992

Some Variants of Todd's Low-Complexity Algorithm

by
Ai-Ping Liao[1]

[1]Center for Applied Mathematics, Cornell University, Ithaca, NY 14853.

Some Variants of Todd's Low-Complexity Algorithm

Ai-Ping Liao[1]

Abstract

Todd [13] describes an interior-point algorithm for linear programming that is almost as simple as the affine-scaling method and yet achieves the currently best complexity of $O(\sqrt{n}t)$ iterations to attain precision t based on the primal-only potential function $\psi(x)$. In this paper we propose some variants of Todd's algorithm by considering two local models: (a). $\psi-$model: trying to reduce the potential function $\psi(x)$ as much as possible; and (b). $c-$model: trying to reduce the objective function $c^T x$ as much as possible, with polynomiality retained. Numerical results are presented comparing these algorithms.

Key words. Linear Programming, interior point algorithm, potential function.

[1]Center for Applied Mathematics, Cornell University, Ithaca, NY 14853.

1 Introduction

Stemming from the seminal paper of Karmarkar [6], many polynomial-time interior-point algorithms have been developed and many contributions have been made towards both the theoretical and practical aspects of interior-point algorithms. Interior-point algorithms can be roughly classified into the following categories: projective-scaling algorithms, affine-scaling algorithms, path-following algorithms, and potential-reduction algorithms. A very comprehensive bibliography for interior-point methods can be found in Kranich [8].

Among the algorithms mentioned above, the affine-scaling algorithm, originally proposed by Dikin [3] and rediscovered by Barnes [2] and Vanderbei-Meketon-Freedman [14], has the simplest form and works well in practice although its polynomial status remains unknown. The search direction used in the affine-scaling algorithm, called the affine-scaling direction, together with the centering direction, the direction towards the analytic center (Sonnevend [11]), forms the basic search direction for most interior-point algorithms. Indeed, the polynomiality of most interior-point algorithms is obtained by balancing the affine-scaling direction and the centering direction properly. One nice way to handle the balance of these directions is the potential-reduction method. Potential-reduction methods were first introduced by Karmarkar [6] and further studied by Gonzaga [5], Ye [15], Freund [4], Kojima-Mizuno-Yoshise [7] and Anstreicher [1]. A potential-reduction algorithm is motivated by seeking a constant reduction in a properly chosen potential function, and the balance between the affine-scaling direction and the centering direction is automatically adjusted and polynomiality can be obtained.

Recently Todd [13] proposed a low complexity algorithm whose search direction is a very simple combination of the affine-scaling direction and the centering direction. The algorithm is based on a primal-only potential function $\psi(x)$ and achieves the currently best complexity of $O(\sqrt{n}t)$ iterations to attain precision t. The potential function $\psi(x)$ in Todd [13] is given implicitly and one only knows that its gradient lies somewhere in a half-line. We note that Mizuno and Nagasawa [9] propose a strictly monotone variant of Todd's algorithm.

In this paper we will further study this potential function and propose some variants of Todd's algorithm. In Section 2 we consider the $\psi-$model: trying to reduce the potential function $\psi(x)$ as much as possible based on the currently available information. In Section 3 we consider another model, called the $c-$model: trying to reduce the objective function $c^T x$ as much as possible, with polynomiality retained. We also consider the $c-$ model for the primal potential function:

$$\phi_q^P(x,s) = q\ln(x^T s) - \sum_{j=1}^{n} \ln(x_j). \tag{1}$$

Numerical results are presented in Section 4 comparing these algorithms.

2 The ψ–model

We consider the linear programming problem in standard form:

$$\begin{array}{ll} \min c^T x & \\ Ax = b & \qquad (P) \\ x \geq 0, & \end{array}$$

where A is $m \times n$. Let $F(P) := \{x \in R^n : Ax = b, x \geq 0\}$ and $F_+(P) := \{x \in F(P) : x > 0\}$. We assume that $F_+(P)$ is nonempty, and that the set of optimal solutions of (P) is nonempty and bounded. Let $v(P)$ denote the optimal value of (P). The dual of (P) is given by

$$\begin{array}{ll} \max b^T y & \\ A^T y + s = c & \qquad (D) \\ s \geq 0. & \end{array}$$

Let $F(D) := \{s \in R^n : A^T y + s = c \text{ for some } y \text{ and } s \geq 0\}$ and $F_+(D) := \{s \in F(D) : s > 0\}$. We note that the second assumption above implies that $F_+(D) \neq \emptyset$ which can be proved by the alternative theorem. For any $q \geq 0$ we define the primal-dual potential function to be

$$\phi_q(x, s) := q \ln(x^T s) - \sum_j \ln x_j - \sum_j \ln s_j - n \ln n \tag{2}$$

for $x \in F_+(P)$ and $s \in F_+(D)$. We also use $\phi(x, s)$ to denote $\phi_{\bar{q}}(x, s)$ where $\bar{q} := n + \sqrt{n}$.

All the algorithms we consider are in the affine-scaling framework, i.e., we first find an initial point $x^0 \in F_+(P)$, and at iteration k we take the affine transformation $x \rightarrow \bar{x} := X_k^{-1} x$ under which (P) is transformed to

$$\begin{array}{ll} \min \bar{c}^T \bar{x} & \\ \bar{A}\bar{x} = b & \qquad (\bar{P}) \\ \bar{x} \geq 0, & \end{array}$$

where $\bar{A} := AX_k$ and $\bar{c} := X_k c$. Also note that $x^k \rightarrow e$ – the all-ones vector. Find a search direction in the scaled space, say $\bar{d}$, and a step size $\lambda > 0$. Set $\bar{x}_+ := e + \lambda \bar{d}$ and transform it back to the original space; we thus have $x^{k+1} = X_k \bar{x}_+ = X_k(e + \lambda \bar{d}) = x^k + \lambda X_k \bar{d}$.

In the scaled problem $(\bar{P})$, the affine-scaling direction is given by

$$-\bar{c}_p := -P_{\bar{A}} \bar{c}, \tag{3}$$

where $P_{\bar{A}}$ is the projection into the null space of $\bar{A}$; the centering direction is

$$e_p := P_{\bar{A}} e. \tag{4}$$

If $\bar{c}_p = 0$, then it is easy to prove that all feasible points of $(\bar{P})$ are optimal, and hence x^k is optimal in (P). We thus assume in the following that $\bar{c}_p \neq 0$. For each β, we define

$$\bar{d}_\beta := -\beta\bar{c}_p + e_p. \tag{5}$$

In particular we define $\bar{d}_\alpha := -\alpha\bar{c}_p + e_p$ where $\alpha := \bar{c}_p^T e/\bar{c}_p^T\bar{c}_p$, i.e.,

$$\bar{d}_\alpha = \operatorname{argmin}\{\|\bar{d}\| : \bar{d} = \bar{d}_\beta \text{ for some } \beta\}.$$

The search direction of Todd's basic algorithm [13] is chosen as follows: if $\|\bar{d}_\alpha\| \geq 0.3$, set $\bar{d} = \bar{d}_\alpha/\|\bar{d}_\alpha\|$; otherwise, set $\bar{d} = -\bar{c}_p/\|\bar{c}_p\|$. The step size used in Todd [13] is 0.2. By working with the primal-only potential function

$$\psi(x) := \min\{\phi(x,s) : s \in F_+(D)\} = \phi(x, s(x)), \tag{6}$$

Todd showed that his basic algorithm is an $O(\sqrt{n}t)$ iteration algorithm. About the potential function $\psi(x)$ we have the following property which is due to Todd [13]:

Proposition 2.1 *If $\hat{x} \in F_+(P)$ then $\inf\{\phi(\hat{x},s) : s \in F_+(D)\}$ is attained by a unique $\hat{s} \in F_+(D)$. Write $\hat{s} = s(\hat{x})$. If $\tilde{x} \in F_+(P)$ with $c^T\tilde{x} \leq c^T\hat{x}$, then $c^T\hat{x} - \hat{x}^T\hat{s} \leq c^T\tilde{x} - \tilde{x}^T\tilde{s}$, where $\hat{s} = s(\hat{x})$, $\tilde{s} = s(\tilde{x})$.*

Note that $\phi(\Lambda^{-1}x, \Lambda s) = \phi(x,s)$ for any positive definite diagonal matrix Λ. We can therefore always scale so that our current iterate x^k is e. We therefore focus on the quantities in the scaled space and omit the overbars in our notation. For any $q \geq 0$, we denote

$$\begin{aligned} d_\delta &:= -P_A(\nabla_x\phi_q(e, s(e))) = -P_A(\frac{q}{e^Ts(e)}s(e) - e) \\ &= -\frac{q}{e^Ts(e)}c_p + e_p \\ &=: -\delta c_p + e_p. \end{aligned} \tag{7}$$

The following is a standard result that can be found for instance in Ye [15].

Lemma 2.2 *if $Ad = 0$, $\|d\| = 1$, and $0 < \lambda < 1$, then*

$$\phi_q(e + \lambda d, s) \leq \phi_q(e,s) + \lambda\nabla_x\phi_q(e,s)^Td + \frac{\lambda^2}{2(1-\lambda)}. \tag{8}$$

Thus if we can find a direction d with $Ad = 0$ such that $d_\delta^T d$ $(= -\nabla_x\phi_q(e,s)^Td)$ is greater than or equal to some positive constant, we can reduce function ϕ by a constant factor by choosing an appropriate step size λ, and ψ can be reduced by at least as much. The larger of $d_\delta^T d$, the more ψ can be reduced. Todd [13] shows that the search direction d of his basic algorithm satisfies $d_\delta^T d \geq 0.25$.

Our first variant of Todd's algorithm is motivated by trying to find a search direction d so that $d_\delta^T d$ is as large as possible, which in turn ensures a more promising reduction for the potential function $\psi(x)$ or a larger step size. We first investigate the value of $\|d_\delta\|$.

Proposition 2.3 *For any $q \geq n+\sqrt{n}$, $\|d_\delta\| \geq \eta$, where $\eta = \frac{1}{2} - \frac{1}{16n}$.*

Proof. Assume the contrary, so that $\|d_\delta\| < \eta$ for some $q \geq n+\sqrt{n}$. Let $s^+ = \frac{e^Ts}{q}(e-d_\delta)$ with $s := s(e)$. We show in the following that
(a) $s^+ \in F_+(D)$,
(b) $\|s^+ - \frac{e^Ts^+}{n}e\| < 0.5\frac{e^Ts^+}{n}$,
(c) $\frac{e^Ts^+}{n} \leq (1-\frac{1-\eta}{\sqrt{n}+1})\frac{e^Ts}{n}$.
The proof of (a) can be found in Todd [13] or Ye [15]. Suppose that (b) does not hold. Then using the same argument of Ye [15] we have

$$\begin{aligned}
\|d_\delta\|^2 &= (\frac{q}{e^Ts})^2\|s^+ - \frac{e^Ts}{n}e\|^2 + \|\frac{qe^Ts^+}{ne^Ts}e - e\|^2 \\
&\geq (\frac{q}{e^Ts})^2 0.5^2 + (\frac{qe^Ts^+}{ne^Ts} - 1)^2 n \\
&\geq \frac{0.5^2 n}{n+0.5^2} = \frac{n}{4n+1} \geq \eta^2
\end{aligned}$$

which is a contradiction. (c) is true since

$$\begin{aligned}
\frac{e^Ts^+}{n} &= \frac{e^Ts}{nq}(e^T(e-d_\delta)) \leq \frac{e^Ts}{nq}(n + \|d_\delta\|_1) \\
&\leq \frac{e^Ts}{nq}(n+\sqrt{n}\|d_\delta\|) \\
&\leq (1-\frac{1-\eta}{\sqrt{n}+1})\frac{e^Ts}{n}.
\end{aligned}$$

Let $q = n+\mu\sqrt{n}$ for some $\mu \geq 1$. Using (b) and applying Lemma 1 of Ye [15] we have

$$n\ln(e^Ts^+) - \sum_{j=1}^n s_j^+ - (n\ln(e^Ts) - \sum_{j=1}^n s_j) \leq 0.25. \tag{9}$$

Using (c) we have

$$\begin{aligned}
\mu\sqrt{n}\ln(e^Ts^+) - \mu\sqrt{n}\ln(e^Ts) &= \mu\sqrt{n}\ln(\frac{e^Ts^+/n}{e^Ts/n}) \\
&\leq \mu\sqrt{n}\ln(1-\frac{1-\eta}{\sqrt{n}+1}) \\
&\leq -\frac{1-\eta}{2}.
\end{aligned} \tag{10}$$

Adding (9) and (10) we have

$$\phi_q(e,s^+) - \phi_q(e,s) \leq -\frac{1-\eta}{2} + 0.25 = -\frac{1}{32n} < 0,$$

which contradicts the choice of $s(e)$. Therefore $\|d_\delta\| \geq \eta$. $\square$

Now we are ready to state our algorithm. As mentioned before we need only to state the iterate in the scaled space.

Algorithm 1(in the scaled space):

Take $\bar{\eta} \leq \eta$, say $\bar{\eta} = 0.499$ if $n \geq 100$.

If $c_p^T e_p \leq 0$, let $\omega = \text{argmin}\{\beta \geq 0 : \|-\beta c_p + e_p\| \geq \bar{\eta}\}$; otherwise let $\omega = \text{argmin}\{\beta \geq \alpha : \|-\beta c_p + e_p\| \geq \bar{\eta}\}$. Set $d_\omega = -\omega c_p + e_p$ and $d = d_\omega/\|d_\omega\|$. Take a step size λ_0 (specified below), and set $x_+ = e + \lambda_0 d$.

We note that Proposition 2.3 ensures the existence of such ω. This ω can be obtained by, at most, solving a quadratic equation of one variable.

Lemma 2.4 *Suppose the search direction d is generated by Algorithm 1. Then $d^T d_\delta \geq \bar{\eta}$.*

Proof. If $c_p^T e_p \leq 0$, since $\delta := \frac{q}{e^T s(e)} > 0$ and $\|d_\delta\| \geq \bar{\eta}$, we thus have

$$
\begin{aligned}
d_\delta^T d_\omega &= (-\delta c_p + e_p)^T(-\omega c_p + e_p) \\
&= \delta\omega\|c_p\|^2 - (\delta+\omega)c_p^T e_p + \|e_p\|^2 \\
&\geq \omega^2\|c_p\|^2 - (\delta+\omega)c_p^T e_p + \|e_p\|^2 \text{ (since } \delta \geq \omega) \\
&\geq \omega^2\|c_p\|^2 - (\omega+\omega)c_p^T e_p + \|e_p\|^2 \\
&= \|d_\omega\|^2.
\end{aligned}
$$

Therefore $d_\delta^T d = \frac{d_\delta^T d_\omega}{\|d_\omega\|} \geq \|d_\omega\| \geq \bar{\eta}$.

Suppose $c_p^T e_p > 0$. If $\omega = \alpha$, i.e. $\|-\alpha c_p + e_p\| \geq \bar{\eta}$, then

$$d_\delta^T d_\omega = d_\delta^T d_\alpha = d_\alpha^T d_\alpha = \|d_\alpha\|^2.$$

Hence $d_\delta^T d = \frac{d_\delta^T d_\omega}{\|d_\omega\|} = \|d_\alpha\| \geq \bar{\eta}$. If $\omega \neq \alpha$, i.e., $\|d_\alpha\| < \bar{\eta}$, then $\delta \geq \omega$ $(\geq \alpha)$ (otherwise $\alpha = \frac{q}{e^T s(e)}$ for some $q \geq n + \sqrt{n}$ contradicting Proposition 2.3) and

$$
\begin{aligned}
d_\delta^T d_\omega &= (-\delta c_p + e_p)^T(-\omega c_p + e_p) \\
&= (-(\delta-\alpha)c_p - \alpha c_p + e_p)^T(-(\omega-\alpha)c_p - \alpha c_p + e_p) \\
&= (-(\delta-\alpha)c_p + d_\alpha)^T(-(\omega-\alpha)c_p + d_\alpha) \\
&= (\delta-\alpha)(\omega-\alpha)\|c_p\|^2 + \|d_\alpha\|^2 \\
&\geq (\omega-\alpha)^2\|c_p\|^2 + \|d_\alpha\|^2 = \|d_\omega\|^2.
\end{aligned}
$$

Hence $d_\delta^T d = \frac{d_\delta^T d_\omega}{\|d_\omega\|} \geq \|d_\omega\| \geq \bar{\eta}$. $\square$

Note that we can take $\bar{\eta} = 0.4$ and have $d_\delta^T d \geq 0.4$ which is better than that of Todd [13] (in which $d_\delta^T d \geq 0.25$). If we take $\bar{\eta} = 0.4$ then the step size can be taken, for example, $\lambda_0 = 0.4$. Using Lemma 2.2 we have

$$
\begin{aligned}
\phi_q(e + \lambda_0 d, s(e)) &\leq \phi_q(e, s(e)) - \frac{(1-2\bar{\eta})\bar{\eta}^2}{2(1-\bar{\eta})} \\
&\leq \phi_q(e, s(e)) - 0.02.
\end{aligned}
$$

Or, if we seek larger decrease of the potential function we can take $\lambda_0 = 0.2$ and $\phi_q(e + \lambda_0 d, s(e)) - \phi_q(e, s(e)) \leq 0.05$. We thus have

Theorem 2.5 *Algorithm 1 reduces the potential function ϕ (or ψ) by at least a fixed positive constant at each iteration. If $\phi(x^0, s^0) = O(\sqrt{n}t)$ for some $s^0 \in F_+(D)$, then after $O(\sqrt{n}t)$ iterations we have x^k with $c^T x^k - v(P) \leq 2^{-t}$.*

One advantage of Algorithm 1 is that the parameter ω can be used to induce a lower bound as shown in the following procedure.

LB 1. Suppose the initial lower bound is z_0. z_0 can be $-\infty$. If $\omega = 0$, $z_{k+1} := z_k$. Otherwise, let $z_+ := c^T e - (n + \sqrt{n})/\omega$ and $z_{k+1} := \max\{z_k, z_+\}$.

This procedure is similar to that in Todd's first variant [13] and we also have

Lemma 2.6 *Assume that $c^T e - z_k \geq e^T s(e)$, so that z_k is a valid lower bound on $v(P)$. Then if z_{k+1} is updated by LB 1, $c^T e - z_{k+1} \geq e^T s(e)$ also, so that z_{k+1} is a valid lower bound too.*

Therefore a line search on the half line $\{e + \lambda d : \lambda \geq 0\}$ seeking to minimize

$$\phi_q^P(x, z_{k+1}) := \begin{cases} -\sum_j \ln x_j & \text{if } z_{k+1} = -\infty, \\ q \ln(c^T x - z_{k+1}) - \sum_j \ln x_j & \text{if } z_{k+1} > -\infty \end{cases} \tag{11}$$

can be employed and the $O(\sqrt{n}t)$ complexity preserved as shown in Todd [13]. Todd [13] shows that a finite lower bound must be generated in $O(\sqrt{n}t)$ iterations. This argument is also available for our procedure. However, in the following we show how to generate $z_1 > -\infty$ if we know an upper bound of $\phi(x^0, s^0)$, say $\phi(x^0, s^0) \leq \kappa$.

If $\omega \neq 0$ then the LB 1 gives $z_1 > -\infty$. Otherwise, we have $d_\omega = e_p$. Since $e^T s(e) = (x^0)^T s(x^0)$, $(q - n)\ln((x^0)^T s(x^0)) \leq \kappa$. Therefore

$$\frac{q}{e^T s(e)} = \frac{q}{(x^0)^T s(x^0)} \geq q 2^{-\kappa}.$$

We thus can take $\omega = q2^{-\kappa}$ which does not alter our result. Hence $z_1 = c^T e - (n + \sqrt{n})/q2^{-\kappa} = c^T e - 2^{\kappa}$.

We mentioned above that Algorithm 1 can generate lower bounds; it can be, on the other hand, improved by incorporating with a given lower bound. Suppose $\{z_k\}$ is a sequence of lower bounds. During each iteration of Algorithm 1, if $c^T e - z_{k+1} \geq e^T s(e)$ and $\frac{q}{c^T e - z_{k+1}} > \omega$ we take $\frac{q}{c^T e - z_{k+1}}$ as the modified ω. By doing so the potential function can be reduced by a larger amount or a larger step size can be employed. This modification preserves the complexity of $O(\sqrt{n}t)$ iterations.

The following procedure, which can be found in Todd [13], generates better lower bounds than those of LB 1.

LB 2. If there is some $\beta > 0$ with $c_p + (e - e_p)/\beta \geq 0$, let $\bar{\beta}$ be the maximum one. Let $z_+ := c^T e - c_p^T e - \|e - e_p\|^2/\bar{\beta}$ and set $z_{k+1} := \max\{z_k, z_+\}$.

With a lower bound we can improve our algorithm in two ways: first, the lower bound may be used to improve our knowledge of the value of d_δ, thus a better search direction can be obtained; second, it allows us to perform a line search seeking to minimize ϕ_q^P of (11) over a half-line hence a larger step size can be employed. By combining these two strategies with the two lower bound generators, LB 1 and LB 2, we have the following variants of Algorithm 1.

Algorithm 1a. Algorithm 1 with a line search using the lower bounds generated by LB 1.

Algorithm 1b. Algorithm 1 with the modified direction using LB 1 and line search with the lower bounds generated by LB 1.

Algorithm 1c. Algorithm 1 with the modified direction using LB 2.

Algorithm 1d. Algorithm 1 with the modified direction using LB 2 and line search with the lower bounds generated by LB 1.

Algorithm 1e. Algorithm 1 with the modified direction using LB 2 and line search with the lower bounds generated by LB 2.

We note that Algorithms 1a–1d have the complexity of $O(\sqrt{n}t)$ iterations, while Algorithm 1e needs $O(nt)$ iterations.

3 The c–model

As we know the "pure" affine-scaling algorithm works well in practice although its polynomial status has not been verified. We thus consider the model that tries to be as close to the affine-scaling algorithm as possible with polynomiality retained.

Let's first modify Algorithm 1. We take $\bar{\eta} = 0.4$. For maintaining a polynomial complexity, we require that the search direction $d = d_\beta/\|d_\beta\|$ for some β and satisfies $d_\delta^T d \geq \tau$ where $\tau < 0.4$ is a small positive constant. Under these constraints we want to find such d with $\frac{-c_p^T d}{\|c_p\|}$ as large as possible. Since d_δ is unknown and the only information about d_δ is that $\delta \in I_\delta := [a_\delta, \infty)$, where

$$a_\delta = \begin{cases} 0 & \text{if } -c_p^T e_p < 0 \text{ and } \|d_\alpha\| \geq 0.4 \\ \omega & \text{otherwise,} \end{cases} \tag{12}$$

we thus come to the problem

$$\begin{array}{l} \max \frac{-c_p^T d}{\|c_p\|} \\ d = d_\beta/\|d_\beta\| \text{ for some } \beta, \text{ or } d = -c_p/\|c_p\| \\ \hat{d}^T d \geq \tau \text{ for all } \hat{d} \in \{d_\beta : \beta \in I_\delta\} \end{array} \qquad \text{(C)}$$

Lemma 3.1 *The solution of problem (C) is given by*

$$d = \begin{cases} -c_p/\|c_p\| & \text{if } -c_p^T d_{a_\delta}/\|c_p\| \geq \tau \\ d_{\beta_0}/\|d_{\beta_0}\| & \text{otherwise} \end{cases}$$

where

$$\beta_0 = \frac{-b+\sqrt{b^2-ac}}{a}$$

with

$$\begin{aligned} a &= (a_\delta\|c_p\|^2 - c_p^T e_p)^2 - \tau^2\|c_p\|^2, \\ b &= (a_\delta\|c_p\|^2 - c_p^T e_p)(-a_\delta c_p^T e_p + \|e_p\|^2) + \tau^2 c_p^T e_p, \\ c &= (-a_\delta c_p^T e_p + \|e_p\|^2)^2 - \tau^2\|e_p\|^2. \end{aligned}$$

Proof. We need only to consider the case when $-c_p^T d_{a_\delta}/\|c_p\| < \tau$. In this case, problem (C) is equivalent to finding the largest β satisfying the constraints of (C). Since $-c_p/\|c_p\| = \lim_{\beta\to\infty} d_\beta/\|d_\beta\|$, $-c_p^T d_{a_\delta}/\|c_p\| < \tau$ implies that there exists some $\beta_u > 0$ such that $d_{\beta_u}^T d_{a_\delta}/\|d_{\beta_u}\| < \tau$. On the other hand,

$$d_\omega^T d_{a_\delta}/\|d_\omega\| = \|d_\omega\| > \tau. \tag{13}$$

Thus, by the continuity, there must exist some $\beta_0 > \omega \geq 0$, such that

$$d_{\beta_0}^T d_{a_\delta}/\|d_{\beta_0}\| = \tau. \tag{14}$$

Solving the above equation we thus get the solution stated in the lemma. The optimality of this solution is verified by noting that $d_\beta^T d_{a_\delta}/\|d_\beta\|$ is a strictly decreasing function for $\beta \in (\omega, \infty)$. □

Since $\beta_0 > \omega$, the search direction $d = d_{\beta_0}/\|d_{\beta_0}\|$ satisfies $-c_p^T d > 0$. Therefore the algorithm with the search direction given in the above lemma is strictly monotone in the objective.

The solution given by Theorem 3.1 is complicated. For getting a simpler form of the search direction we relax the model (C) with the following feasibility problem:

$$\begin{aligned} &\text{find } d \text{ such that:} \\ &\tfrac{-c_p^T d}{\|c_p\|} > 0, \\ &d = d_\beta/\|d_\beta\| \text{ for some } \beta, \\ &\hat{d}^T d \geq \tau \text{ for all } \hat{d} \in \{d_\beta : \beta \in I_\delta\}. \end{aligned} \tag{FC}$$

A simple formula of some solutions of (FC) is given in the following lemma.

Lemma 3.2 *If* $0 < \tau \leq 0.2$, *then for any* ζ *such that* $\zeta_* < \zeta \leq \zeta^*$ *with*

$$(\zeta_*, \zeta^*) := \begin{cases} (\frac{c_p^T e_p}{\|c_p\|\|e_p\|}, 1) & \text{if } -c_p^T e_p < 0 \text{ and } \|d_\alpha\| > 0.4 \\ (0, \frac{\sqrt{0.4^2-\tau^2}}{\tau}) & \text{otherwise,} \end{cases}$$

$d = \tilde{d}/\|\tilde{d}\|$, *where*

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \zeta\frac{-c_p}{\|c_p\|},$$

is a solution to (FC).

Proof. It is obvious by the definition of a_δ that $-c_p^T d > 0$. We shall show in the following that

$$\hat{d}^T d \geq \tau \text{ for all } \hat{d} \in \{d_\beta : \beta \in I_\delta\}.$$

We first consider the case when

$$-c_p^T e_p < 0, \; \|d_\alpha\| > 0.4. \tag{15}$$

In this case, $a_\delta = 0$. Let $\tilde{d}_\zeta := d_{a_\delta} - \zeta\frac{\|d_{a_\delta}\|}{\|c_p\|}c_p$, then $d = \tilde{d}/\|\tilde{d}\| = \tilde{d}_\zeta/\|\tilde{d}_\zeta\|$. By some calculation we have

$$\begin{aligned} \tilde{d}_\zeta^T d_{a_\delta} &= \|d_{a_\delta}\|^2 - \zeta\frac{\|d_{a_\delta}\|}{\|c_p\|}c_p^T d_{a_\delta}, & (16) \\ \|\tilde{d}_\zeta\|^2 &= (1+\zeta^2)\|d_{a_\delta}\|^2 - 2\zeta\frac{\|d_{a_\delta}\|}{\|c_p\|}c_p^T d_{a_\delta}. & (17) \end{aligned}$$

Subtracting half of (17) from (16) we thus have

$$\frac{\tilde{d}_\zeta^T d_{a_\delta}}{\|\tilde{d}_\zeta\|} = \frac{\|\tilde{d}_\zeta\|}{2} + \frac{1-\zeta^2}{2}\frac{\|d_{a_\delta}\|^2}{\|\tilde{d}_\zeta\|}. \tag{18}$$

Since $\|d_\alpha\| > 0.4$, $\|\tilde{d}_\zeta\| > 0.4$ (note that $\tilde{d}_\zeta$ can be written in the form of (5) with $\beta = \zeta\frac{\|d_{a_\delta}\|}{\|c_p\|}$). For any $0 < \zeta \leq 1$, we have by (18),

$$d^T d_{a_\delta} = \frac{\tilde{d}_\zeta^T d_{a_\delta}}{\|\tilde{d}_\zeta\|} \geq \frac{\|\tilde{d}_\zeta\|}{2} \geq 0.2 \geq \tau.$$

Now we suppose that (15) does not hold. In this case, $a_\delta = \omega$, thus $-c_p^T d_{a_\delta} > 0$. We denote by θ the angle between $\tilde{d}$ and d_{a_δ}. Obviously, $\cos\theta > 1/\sqrt{1+\zeta^2}$ and hence

$$\begin{aligned} d^T d_{a_\delta} &= \tilde{d}^T d_{a_\delta}/\|\tilde{d}\| = \|d_{a_\delta}\|\cos\theta \\ &\geq \|d_{a_\delta}\|/\sqrt{1+\zeta^2} \\ &\geq 0.4/\sqrt{1+(\zeta^*)^2} = \tau. \end{aligned}$$

Noting that $d^T d_\beta \geq d^T d_{a_\delta}$ for any $d_\beta \in \{d_\beta : \beta \geq a_\delta\}$ since $-c_p^T d > 0$, the proof is thus complete. □

Assume $q = n + \sqrt{n}$. Then the potential function $\psi(x)$ can be decreased by a constant factor by taking a step in the direction defined in Lemma 3.1 or Lemma 3.2, which leads to an algorithm of complexity of $O(\sqrt{n}t)$. In particular, if we take $\tau = 0.2$, then we can take $\zeta_* = 1$ for all cases. We take $\zeta = \zeta_* = 1$ and have

Algorithm 2(in the scaled space):

Take $d = \tilde{d}/\|\tilde{d}\|$, where

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \frac{-c_p}{\|c_p\|}.$$

Set $x_+ = e + 0.25d$.

It is easy to show that Algorithm 2 reduces the potential function $\psi(x)$ by at least 0.008. We also note that all the search directions defined above lead to a strict decrease in the objective function.

Let's consider some modifications of Algorithm 2. As we mentioned in section 2, if we have a lower bound then this lower bound can be used to improve our knowledge of the value of d_δ. We now use the lower bound generator, LB 2, to modify Algorithm 2 and have

Algorithm 2a(in the scaled space):
If $q/(c^T e - z_{k+1}) > a_\delta$, where $\{z_k\}$ is generated by LB 2, then we take $q/(c^T e - z_{k+1})$ as our new a_δ and take $d = \tilde{d}/\|\tilde{d}\|$, where

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \frac{-c_p}{\|c_p\|}.$$

Set $x_+ = e + 0.25d$.

We can also employ a line search to get a large step size and have

Algorithm 2b(in the scaled space):
Set $q = n + \sqrt{n}$ or $q = 2n$. If $q/(c^T e - z_{k+1}) > a_\delta$, where $\{z_k\}$ is generated by LB 2, then we take $q/(c^T e - z_{k+1})$ as our new a_δ and take $d = \tilde{d}/\|\tilde{d}\|$, where

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \frac{-c_p}{\|c_p\|}.$$

Search on the half-line $\{e + \lambda d : \lambda \geq 0\}$ seeking to minimize

$$\phi_q^P(x, z_{k+1}) = q \ln(c^T x - z_{k+1}) - \sum_j \ln x_j.$$

Let λ_* be the output of this search and set $x_+ = e + \lambda_* d$.

If we choose $\tau = 0.001$ in Lemma 3.2, then we have the following solution

$$d = \tilde{d}/\|\tilde{d}\|, \tag{19}$$

where

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \zeta\frac{-c_p}{\|c_p\|}$$

and

$$\zeta := \begin{cases} 1 & \text{if } -c_p^T e_p < 0,\ \|d_\alpha\| > 0.4 \\ 100 & \text{otherwise.} \end{cases} \tag{20}$$

Use (19) as the basic search direction, incorporating with LB 2 for improving the search direction and a line search, we have

Algorithm 3(in the scaled space):
Set $q = n + \sqrt{n}$ or $q = 2n$. If $q/(c^T e - z_{k+1}) > a_\delta$, where $\{z_k\}$ is generated by LB 2, then we take $q/(c^T e - z_{k+1})$ as our new a_δ and take $d = \tilde{d}/\|\tilde{d}\|$, where

$$\tilde{d} = \frac{d_{a_\delta}}{\|d_{a_\delta}\|} + \zeta\frac{-c_p}{\|c_p\|}$$

and ζ is given by (20). Search on the half-line $\{e + \lambda d : \lambda \geq 0\}$ seeking to minimize

$$\phi_q^P(x, z_{k+1}) = q \ln(c^T x - z_{k+1}) - \sum_j \ln x_j.$$

Let λ_* be the output of this search and set $x_+ = e + \lambda_* d$.

Algorithm 2a has the complexity of $O(\sqrt{n}t)$ iterations, while Algorithm 2b and Algorithm 3 have the complexity of $O(nt)$ iterations since the line searches in Algorithm 2b and Algorithm 3 cannot guarantee a constant decrease for the potential function $\psi(x)$.

We note that these models apply also to the algorithms of Freund [4], Gonzaga [5] and Ye [15] where the direction $d_\delta := -P_A(\nabla_x \phi_q(e, s))$ is explicitly given by letting $\delta = q/(e^T s)$ in (7). We thus can get the corresponding c-models by replacing the interval I_δ in Lemma 3.1 and Lemma 3.2 with the singleton $\{\delta := q/(e^T s)\}$. These variants maintain the same complexity as the original algorithms.

4 Computational results

The test problems were generated in the same way as in Todd [13], i.e., for a given m and n, we generated each entry of A, y and s as an independent standard Gaussian random variable, then set $b = Ae$ and $c = A^T y + |s|$, where $|s| := (|s_j|)$; the initial solution was $x^0 = e$. We also used the same termination criterion as Todd [13]:

$$(c^T x - z_k)/\max\{1, |c^T x|\} < 10^{-4} \tag{21}$$

where $x = x^k + \lambda_{\max} d^k$ with $\lambda_{\max} = \max\{\lambda : x^k + \lambda d^k \geq 0\}$.

We tested the following algorithms: Algorithm 1 with $\bar{\eta} = 0.4$ and the step size $\lambda_0 = 0.4$; Algorithm 1a; Algorithm 1b; Algorithm 1c; Algorithm 1d; Algorithm 1e; Algorithm 2 with $q = n + \sqrt{n}$; Algorithm 2a with $q = n + \sqrt{n}$; Algorithm 2b with $q = n + \sqrt{n}$ and $q = 2n$; Algorithm 3 with $q = n + \sqrt{n}$ and $q = 2n$. We used the termination criterion (21) and z_k was generated either by LB 2, if the procedure LB 2 was involved, or, otherwise, by LB 1. For one 50×100 problem, we tested all these algorithms and the results are given in Table 1. All runs were performed using PRO-MATLAB [10] Version 3.5h on a Sun SPARCstation 2.

Among these algorithms, Algorithm 2b and Algorithm 3 (both with $q = n + \sqrt{n}$ or $q = 2n$) are the best. This is because Algorithm 2b and Algorithm 3 employ a line search and a better lower bound generator (LB 2) which improve the search direction and achieve a larger step size.

We solved ten random 50×100 problems using Algorithm 3. With $q = n + \sqrt{n}$, the average number of iterations was 12.5, with $\lambda_k/\lambda_{\max}$ typically 0.88; with $q = 2n$, the average number of iterations was 10.5 and the typical $\lambda_k/\lambda_{\max}$ was 0.97.

We also solved some larger problems and the results are reported in Table 2 which gives the average number of iterations required for five random problems and Algorithm 3 (with $q = 2n$). The data in the last 3 rows are taken from Todd [12], and we quote them for a comparison. (Data in parentheses are the corresponding typical $\lambda_k/\lambda_{\max}$.)

Table 1: Computational results on a 50×100 problem.

Algorithm	Number of iterations	Typical $\lambda_k/\lambda_{\max}$
Algorithm 1	167	0.08
Algorithm 1a	95	0.14
Algorithm 1b	95	0.14
Algorithm 1c	131	0.08
Algorithm 1d	104	0.10
Algorithm 1e	13	0.88
Algorithm 2	220	0.05
Algorithm 2a	208	0.05
Algorithm 2b, $q = n + \sqrt{n}$	12	0.88
Algorithm 2b, $q = 2n$	11	0.98
Algorithm 3, $q = n + \sqrt{n}$	12	0.88
Algorithm 3, $q = 2n$	11	0.98

Table 2: Computational results of Algorithm 3.

	100×200	200×400	300×600	400×800
Algorithm 3, $q = 2n$	11.2(0.98)	11.8(0.98)	13.6(0.98)	14.0(0.98)
Affine-scaling	11.8(0.95)	12.8(0.95)	13.8(0.95)	14.4(0.95)
Todd's Variant 2	12.2(0.96 $\sim$ 0.99)	13.6(0.96 $\sim$ 0.99)	13.8(0.96 $\sim$ 0.99)	14.4(0.96 $\sim$ 0.99)
Todd's Variant 3	13.6(0.88 $\sim$ 0.99)	16.0(0.88 $\sim$ 0.99)	18.2(0.88 $\sim$ 0.99)	19.2(0.88 $\sim$ 0.99)

5 Conclusion

The potential function $\psi(x)$ is the only primal-only function we know that can ensure a bound of $O(\sqrt{n}t)$ iterations. Usually, given a potential function, the search direction is defined as the projection of the negative of the gradient of this potential function. However, here we do not know this quantity for the function $\psi(x)$; the only information about $d_\delta := -P_A\psi$ is that it lies somewhere in a half-line, i.e. $d_\delta = -\delta c_p + e_p$ and $\delta \in I_\delta := \{\beta : \beta \geq a_\delta\}$. Trying to find a β_0 such that $d_{\beta_0}/\|d_{\beta_0}\|$ is as close to $d_\delta/\|d_\delta\|$ as possible leads to our first model: the ψ–model. On the other hand, it is well known that the "pure" affine-scaling algorithm works well in practice, though its polynomial status is still unknown. We thus try to find a search direction that is as close to the affine-scaling direction as possible with polynomiality retained. This is the motivation of our second model: the c–model.

By choosing different parameters in these models, we have different algorithms. Especially, when there is a lower bound, we can use it to improve our knowledge of the value of δ, thus a better search direction can be obtained. We can also employ a line search to get a larger step size. These two strategies associated with the lower bound improve the efficiency of our algorithms dramatically. As a matter of fact, more efficient algorithms can be obtained if there is a better lower bound generator.

Acknowledgment: I would like to thank Professor Michael J. Todd for his constant help and encouragement. I am also very grateful to him for his studying the early version of this paper and for his helpful suggestions and corrections.

References

[1] K. M. Anstreicher. A combined phase I – phase II scaled potential algorithm for linear programming. *Mathematical Programming*, 52:429–439, 1991.

[2] E. R. Barnes. A variation on Karmarkar's algorithm for solving linear programming problems. *Mathematical Programming*, 36:174–182, 1986.

[3] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Doklady Akademii Nauk SSSR*, 174:747–748, 1967. Translated in : *Soviet Mathematics Doklady*, 8:674–675, 1967.

[4] R. M. Freund. Polynomial–time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical Programming*, 51:203–222, 1991.

[5] C. C. Gonzaga. Large steps path–following methods for linear programming, Part II : Potential reduction method. *SIAM Journal on Optimization*, 1:280–292, 1991.

[6] N. K. Karmarkar. A new polynomial–time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[7] M. Kojima, S. Mizuno, and A. Yoshise. An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50:331–342, 1991.

[8] E. Kranich. Interior point methods for mathematical programming : A bibliography. Discussion Paper 171, Institute of Economy and Operations Research, FernUniversität Hagen, P.O. Box 940, D–5800 Hagen 1, West–Germany, May 1991. The (actual) bibliography can be accessed electronically by sending e–mail to '*netlib@research.att.com*' with message '*send index from bib*'.

[9] S. Mizuno and A. Nagasawa. Strict monotonicity in todd's low-complexity algorithm for linear programming. Technical Report, Dept. Prediction and Control, Tokyo Institute of Technology, Oh–Okayama, Meguro–ku, Tokyo 152, Japan, 1991, to appear in *Operations Research Letters.*

[10] C. B. Moler, J. Little, S. Bangert, and S. Kleiman. *Pro-Matlab User's Guide.* MathWorks, Sherborn, MA, 1987.

[11] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prekopa, J. Szelezsan, and B. Strazicky, editors, *System Modelling and Optimization : Proceedings of the 12th IFIP–Conference held in Budapest, Hungary, September 1985*, volume 84

of *Lecture Notes in Control and Information Sciences*, pages 866–876. Springer Verlag, Berlin, West–Germany, 1986.

[12] M. J. Todd. Playing with interior points. *COAL Newsletter*, 19:17–25, August 1991.

[13] M. J. Todd. A low complexity interior point algorithm for linear programming. *SIAM Journal on Optimization*, 2:198–209, 1992.

[14] R. J. Vanderbei, M. S. Meketon, and B. A. Freedman. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1(4):395–407, 1986.

[15] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.