Optimization Methods & Software, 2014
Vol. 29, No. 3, 497–502, http://dx.doi.org/10.1080/10556788.2013.814653

An interesting characteristic of phase-1 of dual–primal algorithm for linear programming

Haohao Li*

Department of Mathematics, State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, People's Republic of China

(*Received 16 April 2013; final version received 10 June 2013*)

Most algorithms for solving linear program require a phase-1 procedure to find a feasible solution. Recently, a dual–primal algorithm for linear optimization has been proposed by Li [*Dual–primal algorithm for linear optimization*, Optimiz. Methods Softw. 28 (2013), pp. 327–338]. In the process of implementing the dual–primal algorithms, we found an interesting phenomenon that the phase-1 algorithm developed in [Li (2013)] always terminates in one iteration. This fact does not come by chance. A rigorous proof is given in this paper.

Keywords: linear programming; dual–primal algorithm; Farkas's Lemma

MSC 2010: 65C05; 90C05

1. Introduction

Most algorithms for linear programming problems, say, simplex and interior point methods, require a phase-1 algorithm to obtain a feasible solution as input, e.g., see [2,3,5–10]. Recently, an interesting dual–primal algorithm for linear optimization and a new phase-1 have been proposed [4]. This new phase-1 algorithm exploits the advantage of dual–primal algorithm and only one single artificial variable is introduced. When applying the dual–primal algorithm to some linear program arisen from scheduling problems, we note that all test problems obtain the feasible solution by this new phase-1 procedure with only one iteration. This fact does not come by chance. A rigorous proof is given in this paper.

In the next section, for self-contained the dual–primal algorithm and the new phase-1 algorithm are presented. Then in Section 3, it is proved that this new phase-1 algorithm is always terminated in one iteration. Some conclusion remarks are given in Section 4.

2. Dual–primal algorithm for linear programming and corresponding phase-1 algorithm

Consider the linear programming problem in the standard form

$$\begin{aligned} \min \quad & c^{\mathrm{T}}x \\ \text{s.t.} \quad & Ax = b, \quad x \geq 0, \end{aligned} \tag{1}$$

*Email: hhlzju@126.com

and its associated dual problem

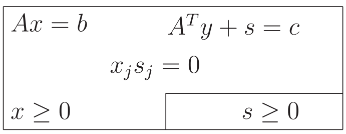
$$\begin{aligned} \max \quad & b^{\mathrm{T}}y \\ \text{s.t.} \quad & A^{\mathrm{T}}y + s = c, \quad s \geq 0, \end{aligned} \tag{2}$$

where $c, x, s \in R^n, b, y \in R^m, A \in R^{m\times n}$ with $m < n, \operatorname{rank}(A) = m$.

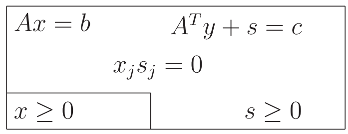
If $x \in R^n$ is primal feasible, and $y \in R^m$ is dual feasible and $x^{\mathrm{T}}s = 0$ then x and y are primal and dual optimal respectively. This well know result shows that the set of optimal solutions can be characterized as the solutions of the Karush–Kuhn–Tucker (KKT) system as follows:

$$1.\ Ax = b; \quad 2.\ A^{\mathrm{T}}y + s = c; \quad 3.\ x^{\mathrm{T}}s = 0; \quad 4.\ x \geq 0; \quad 5.\ s \geq 0.$$

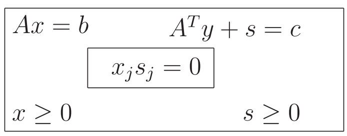
An algorithm for solving linear optimization is called a *single-term KKT algorithm*, if it maintains the validity of any four of above five conditions in the solving process and tries to achieve the remainder one condition, which acts as a termination condition. Primal simplex method, dual simplex method, primal–dual method and interior point method are all single-term KKT algorithms. Patterns 1, 2, 3 and 5 below illustrate the requirements of the above mentioned algorithms, respectively [4].



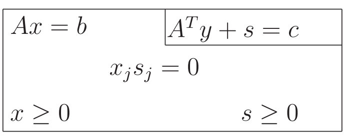
Pattern 1. simplex method



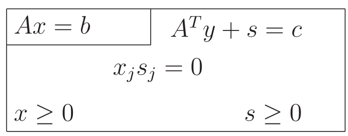
Pattern 2. dual simplex method



Pattern 3. interior point methods



Pattern 4. dual-primal algorithm



Pattern 5. primal-dual algorithm

The dual–primal algorithm for linear optimization associated with patterns 4, where preserving all KKT conditions 1, 3, 4 and 5, but the equality 2 is attained at termination, has been newly developed by Li [4].

Let $\bar{x}$ be a primal feasible solution to the problem (1). Let M and N be the two index sets, respectively:

$$\begin{aligned} M &= \{i|\bar{x}_i > 0,\ i \in \{1,\dots,n\}\}, \\ N &= \{i|\bar{x}_i = 0,\ i \in \{1,\dots,n\}\}. \end{aligned} \tag{3}$$

The coefficient matrix A and the identity coefficient matrix of s can be partitioned correspondingly as $[A_M, A_N]$ and $[I_M, I_N]$, respectively. Vectors x, s and c are partitioned conformably. Consider least-squares problem

$$\min_{z\geq 0} \|\hat{A}z - c\|^2, \tag{4}$$

where $\hat{A} = [A^{\mathrm{T}}, -A^{\mathrm{T}}, I_N] \in R^{n\times(2m+|N|)}$, $z^{\mathrm{T}} = (u^{\mathrm{T}}, v^{\mathrm{T}}, s_N^{\mathrm{T}})^{\mathrm{T}} \in R^{2m+|N|}$.

Dual–primal algorithm (main procedure) [4]

Let $\bar{x}$ be a primal feasible solution.

1. Determine N by (3).
2. Compute the residual $r = [A^{\mathrm{T}}, I_N]\left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right] - c$ at the solution to the problem (4).
3. Stop if the residual $r = 0$: the current solution $\bar{x}$ and

$$\bar{s} = \begin{cases} 0 & \text{for } j \in M; \\ \bar{s}_j & \text{for } j \in N; \end{cases}$$

   are primal and dual optimal, respectively.
4. Stop if $r \geq 0$: the problem (1) is unbounded below.
5. Determine step length α by $\alpha = \min\{-\bar{x}_j/r_j | r_j < 0, j \in M\}$.
6. Update $\bar{x}$ by $\bar{x} := \bar{x} + \alpha r$, and go to Step 1.

The dual–primal algorithm requires a primal feasible solution as its input. A phase-1 approach which only introduces a single artificial variable is developed in [4]. Without loss of generality we assume that $b \geq 0$ in the program (1). Introduce an artificial variable x_{n+1} and construct the following auxiliary program:

$$\begin{aligned} \min \quad & x_{n+1} \\ \text{s.t.} \quad & Ax + x_{n+1}b = b, \\ & x \geq 0, \quad x_{n+1} \geq 0. \end{aligned} \tag{5}$$

Program (5) has *an* obviously feasible solution $x^{(0)} = 0 \in R^n$, $x_{n+1}^{(0)} = 1$.

Phase-1 algorithm [4] Solve (5) by dual–primal algorithm. Let $(x^*, x_{n+1}^*) \geq 0$ be the optimal solution to (5). If $x_{n+1}^* > 0$, then the program (1) is infeasible. If $x_{n+1}^* = 0$, then x^* is a feasible solution to program (1).

Main result

Like least-squares primal–dual algorithm proposed in [1], the dual–primal algorithm [4] is expected to have an advantage on combinatorial problems, such as set partitioning problems, assignment problems and scheduling problems etc, over the simplex method, because these problems are highly degenerate.

In this section, we prove an interesting property – that the Phase-1 procure of dual–primal algorithm always terminates in one iteration. First, we introduce two propositions to be used later in the proofs of the main theorem.

PROPOSITION 1 *If the residual r at the solution to* (4) *satisfies* $r \neq 0$, *then it is a feasible direction of the problem* (1).

Proof Proposition 1 is Lemma 4.3 in [4]. ∎

PROPOSITION 2 *If the dual–primal algorithm terminates at Step* 4, *i.e.* $r \geq 0$ *and* $r \neq 0$, *then the problem* (1) *is unbounded below.*

Proof Proposition 2 is Lemma 4.4 in [4]. ■

Now we prove the following main result.

THEOREM *The Phase-1 procedure terminates in one iteration at either*

(i) *with a feasible solution to the problem* (1) *reached; or*
(ii) *detecting the infeasibility of problem* (1).

Proof Rewrite the problem (5) as

$$\begin{aligned} \min \quad & \tilde{c}^{\mathrm{T}}\tilde{x} \\ \text{s.t.} \quad & \tilde{A}\tilde{x} = b, \\ & \tilde{x} \geq 0, \end{aligned} \tag{6}$$

where $\tilde{c} = (0, \ldots, 0, 1)^{\mathrm{T}} \in R^{n+1}, \tilde{x} = (x_1, \ldots, x_n, x_{n+1})^{\mathrm{T}} \in R^{n+1}$, $\tilde{A} = [A, b]_{m\times(n+1)}$. The auxiliary problem (6) has an obviously feasible solution $\tilde{x}^{(0)} = (0, \ldots, 0, 1)^{\mathrm{T}} \in R^{n+1}$, and therefore it exists a nonnegative optimal solution, since the objective function $\tilde{c}^{\mathrm{T}}\tilde{x} = x_{n+1} \geq 0$. Let the residual of the associated nonnegative least-squares problem

$$\min_{z\geq 0} \left\| [\tilde{A}^{\mathrm{T}}, I_N] \begin{bmatrix} y \\ z \end{bmatrix} - \tilde{c} \right\|^2 \tag{7}$$

be

$$r^{(0)} = [\tilde{A}^{\mathrm{T}}, I_N] \begin{bmatrix} y \\ z \end{bmatrix} - \tilde{c} = \begin{pmatrix} r_1^{(0)} \\ \cdots \\ r_n^{(0)} \\ r_{n+1}^{(0)} \end{pmatrix}, \tag{8}$$

where

$$I_N = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{(n+1)\times n}$$

is the first n columns of the identity matrix I_{n+1}.

By Proposition 1 we know that $\tilde{x}^{(0)} + \alpha r^{(0)} \geq 0$, for some $\alpha > 0$, i.e.,

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} r_1^{(0)} \\ \vdots \\ r_n^{(0)} \\ r_{n+1}^{(0)} \end{pmatrix} \geq 0$$

and hence $r_1^{(0)} \geq 0, \ldots, r_n^{(0)} \geq 0$ and

$$1 + \alpha r_{n+1}^{(0)} \geq 0. \tag{9}$$

Now consider $r_{n+1}^{(0)}$, we distinguish two cases:

① $r_{n+1}^{(0)} < 0$.

In case ①, $r_{n+1}^{(0)}$ is the unique nonnegative component of the residual. Thus, the inequality (9) implies that the next step length is given by

$$\alpha = -\frac{1}{r_{n+1}^{(0)}}.$$

Thus, after one iteration we obtain the next solution

$$x_j^{(1)} = x_j^{(0)} + \frac{r_j^{(0)}}{(-r_{n+1}^{(0)})} \geq 0, \quad j = 1, \ldots, n,$$
$$x_{n+1}^{(1)} = x_{n+1}^{(0)} + \frac{r_{n+1}^{(0)}}{(-r_{n+1}^{(0)})} = 1 - 1 = 0.$$

Note that $x_{n+1}^{(1)} = 0$, thus, $(x_1^{(1)}, \ldots, x_{n+1}^{(1)})$ is an optimal solution of the problem (5), and hence $(x_1^{(1)}, \ldots, x_n^{(1)})$ is a feasible solution of the original linear programming problem (1).

② $r_{n+1}^{(0)} \geq 0$.

In case ②, the problem (1) is infeasible. In fact, in this case the condition $r_1^{(0)} \geq 0, \ldots, r_n^{(0)} \geq 0$ and $r_{n+1}^{(0)} \geq 0$ imply that $r_1^{(0)} = 0, \ldots, r_n^{(0)} = 0, r_{n+1}^{(0)} = 0$, since otherwise by Proposition 2, the auxiliary program (6) will be unbounded if there exits $j \in \{1, \ldots, n+1\}$ such that $r_j^{(0)} > 0$. Thus by (8) we have

$$[\tilde{A}^{\mathrm{T}}, I_N] \begin{pmatrix} y^{(0)} \\ z^{(0)} \end{pmatrix} - \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{10}$$

where $y^{(0)} \in R^m, z^{(0)} \in R^n$. From (10) it follows that

$$\begin{pmatrix} A^{\mathrm{T}} \\ b^{\mathrm{T}} \end{pmatrix} y^{(0)} + I_N z^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{11}$$

where $0 \in R^n$. Note that $z^0 \geq 0$, by (10) we have

$$\begin{aligned} A^{\mathrm{T}} y^0 &\leq 0 \\ b^{\mathrm{T}} y^0 &= 1 \end{aligned} \tag{12}$$

which implies that the system

$$\begin{aligned} A^{\mathrm{T}} y &\leq 0 \\ b^{\mathrm{T}} y &> 0 \end{aligned} \tag{13}$$

has a solution. By Farkas's Lemma we know that the system

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{14}$$

has no solution. So the problem (1) is infeasible. ■

3. Final remark

We prove that one can find a feasible solution or detect the infeasibility to the problem (1) in only one iteration, by applying the phase-1 procedure of dual–primal algorithm developed in [4]. It may be interesting to note that if we set $y = u - v, u, v \geq 0$, then the KKT conditions for linear programming problem (1) can be written as

$$\begin{aligned} Ax &= b, \\ A^{\mathrm{T}}u - A^{\mathrm{T}}v + s &= c, \\ c^{\mathrm{T}}x - b^{\mathrm{T}}u + b^{\mathrm{T}}v &= 0, \\ x, s, u, v &\geq 0. \end{aligned} \tag{15}$$

Thus, finding an optimal solution to problem (1) is equivalent to the problem of finding a solution to a larger linear system (15). Now for any given $c \in R^n$, say, $c = 0$, consider linear program

$$\begin{aligned} \min \quad & 0^{\mathrm{T}}x \\ \text{s.t.} \quad & \text{constrains (15)} \end{aligned} \tag{16}$$

Clearly, finding an optimal solution to problem (1) is equivalent to find a feasible solution to (16). Apply the phase-1 to problem (16), it terminates in one iteration. Thus, to solve problem (1) is equivalent to solve a least-squares problem with nonnegative constraints, which is solvable in polynomial-time.

Acknowledgement

The author is grateful to two anonymous referees for their constructive comments and suggestions, which have improved the presentation of this paper.

References

[1] E. Barnes, V. Chen, B. Gopalakrishnan, and E.L. Johnson, *A least-squares primal–dual algorithm for solving linear programming problems*, Oper. Res. Lett. 30 (2002), pp. 289–294.
[2] K. Fukuda and T. Terlaky, *Criss-cross methods: A fresh view on pivot algorithms*, Math. Program. 79 (1997), pp. 369–395.
[3] T. Illés and T. Terlaky, *Pivot versus interior points methods: Pros and cons*, Eur. J. Oper. Res. 140 (2002), pp. 170–190.
[4] W. Li, *Dual–primal algorithm for linear optimization*, Optim. Methods Softw. 28 (2013), pp. 327–338.
[5] W. Li, P. Guerrero-García, and A. Santos-Palomo, *A basis-deficiency-allowing phase-I algorithm using the most-obtuse-angle column rule*, Comput. Math. Appl. 51 (2006), pp. 903–914.
[6] W. Li, P.Q. Pan, and G.T. Chen, *A combined projected gradient algorithm for linear programming*, Optimiz. Methods Softw. 21 (2006), pp. 541–550.
[7] P.Q. Pan, *A basis-deficiency-allowing variation of the simplex method*, Comput. Math. Appl. 36 (1998), pp. 33–53.
[8] C. Roos, T. Terlaky, and J.P. Vial (eds.), *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley, New York, 1997.
[9] T. Terlaky and S. Zhang, *Pivot rule for linear programming: A survey on recent theoretical developments*, Ann. Oper. Res. 46 (1993), pp. 203–233.
[10] Y. Ye (ed.), *Interior Point Algorithms*, John Wiley, New York, 1997.