Classical Simplex Methods for Linear Programming and Their Developments *

Yan Zizong † Fei Pusheng‡ Wang Xiaoli §

Abstract

This paper presents a new primal dual simplex method and investigates the duality formation implying in classical simplex methods. We reviews classical simplex methods for linear programming problems and give a detail discussion for the relation between modern and classical algorithms. The two modified versions are present. The advantages of the new algorithms are simplicity of implementation, low computational overhead and surprisingly good computational performance. they always proved to be more efficient than classical simplex methods on our test problems.

Keywords. Linear programming, Duality gap, Simplex method, Pivot rule

AMS subject classifications. 90C05, 90C49, 65K05

1 Introduction

The classical simplex algorithm is the most popularly used for linear programming. It performs sufficiently well in practice, particularly on linear problems of small or medium size. Recently, Chen et al.[1] and Paparrizos et al. [3] developed a primal-dual simplex algorithm for the general linear programming problem. In this paper we will present a novel primal-dual simplex algorithm that use the current equivalent facet technique.

This paper is organized as follows. In Section 2 we recall some well-known facts about the pivot algorithms and give some definitions such as the hard upper bound,

*Foundation item: Supported by the National Natural Science Foundation of China (70371032) and and the Specialized Research Fund for the Doctoral Program of Higher Education (20020486035), P.R.C.

†School of Mathematics and Statistics, Wuhan university, China, and School of Information and Mathematics, Yangtze university, Jingzhou, Hubei, China.

‡School of Mathematics and Statistics, Wuhan university,China.

§School of Mathematics and Statistics, Wuhan university,China.

[0]Biography: Yan Zizong :(1964-),male,Doctor candidate, research direction: optimization theory and method. E-mail:zzyan@jznu.net

etc. Also, we present a new primal dual algorithm. In Section 3 we describe the relation between the primal dual algorithm and classical simplex methods and investigate the duality properties implying in classical simplex methods. We give a new simple proof for finiteness of pivot methods. In Section 4 we present two modified versions of the primal dual algorithm so that we find a better search direction. In Section 5 we give the computational results that demonstrate the superiority of our algorithms over classical methods on a class of randomly generated problems. Finally, we give our conclusion and discuss possible extension of our algorithms.

2 Algorithm description

2.1 Natation and definitions

Let us introduce some notation and state some results. Let $c, x \in R^n, b \in R^m$ and A be an $m \times n$ matrix. The primal LP problem in standard format is given by

$$\min\ c^T x \quad s.t. \quad Ax = b,\ x \geq 0 \tag{2.1}$$

The associated dual problem reads

$$\max\ b^T w \quad s.t. \quad A^T w + r = c,\ r \geq 0 \tag{2.2}$$

The sets of feasible solutions of (2.1) and (2.2) are denoted by P and D respectively. Problem (2.1) is called feasible if the set P is nonempty; if P is empty then (2.1) is infeasible; if there is a sequence of feasible solutions for which the objective value goes to minus infinity the (2.1) is said to be unbounded; analogous statements hold for (2.2). We assume throughout that A has full row rank. The first theorem is the main result in the theory of LP.

Theorem 2.1. *For problems* (2.1) *and* (2.2) *one of the following alternatives hold:*
(i) Both two problems are feasible and there exist $x^* \in P$ *and* $(w, r) \in D$ *such that* $c^T x^* = b^T y^*$*;*
(ii) (2.1) *is infeasible and* (2.2) *is unbounded;*
(iii) (2.2) *is infeasible and* (2.1) *is unbounded;*
(iv) Both are infeasible.

An alternative way of writing the optimality condition in the Theorem (2.1) is by using the complementary slack condition

$$x^T r = 0 \tag{2.3}$$

The complementary slackness condition represents a strong combinatorial character of linear programming problems that is featured by pivot algorithms. The set of optimal solutions can be characterized as the set of solutions of the system

$$\begin{aligned} Ax = b, &\quad x \geq 0 \\ A^T w + r = c, &\quad r \geq 0 \\ r^T x = 0 & \end{aligned} \tag{2.4}$$

Since all pivot methods generate basic solutions, their intermediate solutions satisfy both constraints and complementary slackness condition(see [4]).

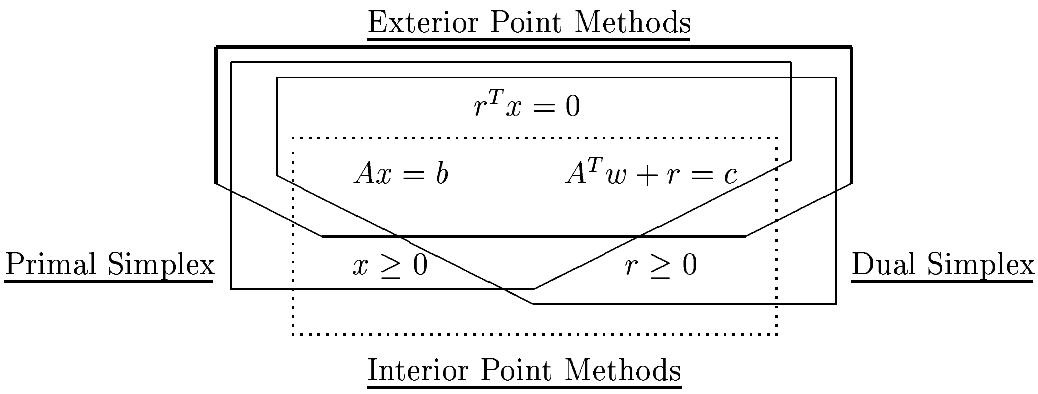

Fig. 1. Algorithms

We refer the increment of the objective value z (replaced by $-x_0$) to as a variable and consider the problem (2.1) in equality form

$$\begin{array}{rl} \max & x_0 \\ s.t. & x_0 + c^T x = 0, \quad Ax = b, \quad x \geq 0 \end{array} \tag{2.5}$$

Set

$$\hat{A} = \begin{pmatrix} 1 & c^T \\ 0 & A \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} x_0 \\ x \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

The problem can be rewritten in the following form

$$\max \ x_0 \quad s.t. \ \hat{A}\hat{x} = \hat{b}, \ x \geq 0 \tag{2.6}$$

It is called the increment model, or the current equivalent facet model in [5], of the problem (2.1).

To distinguish successive pivotal transforms of simplex methods, we shall use the superscript ν as an iteration counter. The initial value of ν will be 0. The vector x^ν and r^ν represent the current basic solution and the current reduced cost vector, respectively. Sometimes the notations x^ν_P, r^ν_P (or x^ν_D, r^ν_D) are also used in the primal (or dual) simplex method.

In general, after ν principal pivots, the system (2.5) will be $\hat{A}^\nu \hat{x}^\nu = \hat{b}^\nu$. Its can also be represented in the tableau form

x^ν_0	x^ν_1	$\cdots$	x^ν_n	rhs
$\hat{a}^\nu_{01}$	$\hat{a}^\nu_{02}$	$\cdots$	$\hat{a}^\nu_{1n}$	$\hat{b}^\nu_0$
$\hat{a}^\nu_{11}$	$\hat{a}^\nu_{12}$	$\cdots$	$\hat{a}^\nu_{2n}$	$\hat{b}^\nu_1$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$\hat{a}^\nu_{m1}$	$\hat{a}^\nu_{m2}$	$\cdots$	$\hat{a}^\nu_{m,n}$	$\hat{b}^\nu_m$

Because of the non-negative constrict of variables, we must assure that the inequalities

$$\hat{b}_i^\nu - \hat{a}_{i0}^\nu \hat{x}_0^\nu \geq 0, \quad i = 0, 1, \cdots, m \tag{2.7}$$

are satisfied in the primal simplex method. The set of solutions of this inequalities is an internal or an empty set. If $\hat{a}_{i0}^\nu > 0$ is satisfied for an index i, the associated component will remain feasible down to the critical value $\delta_i^\nu = \frac{\hat{b}_i^\nu}{\hat{a}_{i0}^\nu}$. For an index i, δ_i^ν is called an upper bound provided $\hat{a}_{i0}^\nu > 0$. In particular, $\delta_i^\nu = +\infty$ is called an upper bound provided $\hat{a}_{i0}^\nu = 0$ and $\hat{b}_i^\nu > 0$, and $\delta_i^\nu = -\infty$ is called an upper bound provided $\hat{a}_{i0}^\nu = 0$ and $\hat{b}_i^\nu < 0$. Analogous we can define a lower bound.

If δ_i^ν is an upper bound with $\hat{a}_{ij}^\nu \geq 0, j = 0, 1, \cdots, n$, δ_i^ν is called a hard upper bound; otherwise, it is called a soft upper bound.

2.2 Basic framework

The basis x^ν is dual feasible only if its complementary slackness dual r^ν in (2.3) is equal to or greater than 0. It means that there exists an index l in the system (2.6) such that the right coefficient $\hat{b}_l^\nu$ is equal to zero and the left each coefficient $\hat{a}_{lj}^\nu$ is equal to or greater than zero in the equation

$$\hat{a}_{l0}^\nu x_0 + \hat{a}_{l1}^\nu x_1^\nu + \cdots + \hat{a}_{ln}^\nu x_n^\nu = \hat{b}_l^\nu \tag{2.8}$$

Without loss of generality, we might assume $\hat{a}_{l0}^\nu > 0$ regardless of the right hand constraint $\hat{b}^\nu$. Then the equation (2.8) leads to one dual vector

$$(\frac{\hat{a}_{l1}^\nu}{\hat{a}_{l0}^\nu}, \frac{\hat{a}_{l2}^\nu}{\hat{a}_{l0}^\nu}, \cdots, \frac{\hat{a}_{ln}^\nu}{\hat{a}_{l0}^\nu})^T \tag{2.9}$$

with the duality gap $\frac{\hat{b}_l^\nu}{\hat{a}_{l0}^\nu}$. This critical value is just a hard upper bound defined in the previous subsection. If $\hat{b}_l^\nu > 0$, we try to pivot the simplex tableau so that the duality gap decreases. The equation (2.8) is also called a dual feasible constraint only if $\hat{a}_{lj}^\nu \geq 0, j = 0, 1, \cdots, n$. In other word, a basis is called dual feasible if there exists a dual feasible constraint in the simplex tableau.

If $\hat{a}_{p0}^\nu > 0$ and $\min\{\hat{a}_{p1}^\nu, \cdots, \hat{a}_{pn}^\nu\} < 0$, the equation

$$\hat{a}_{p0}^\nu x_0^\nu + \hat{a}_{p1}^\nu x_1^\nu + \cdots + \hat{a}_{pn}^\nu x_n^\nu = \hat{b}_p^\nu \tag{2.10}$$

decides a primal test row for the current solution x^ν. It was also called the current equivalent facet in [5, 6]. It leads to another dual infeasible vector

$$(\frac{\hat{a}_{p1}^\nu}{\hat{a}_{p0}^\nu}, \frac{\hat{a}_{p2}^\nu}{\hat{a}_{p0}^\nu}, \cdots, \frac{\hat{a}_{pn}^\nu}{\hat{a}_{p0}^\nu})^T \tag{2.11}$$

The corresponding critical value $\frac{\hat{b}_p^\nu}{\hat{a}_{p0}^\nu}$ is a soft upper bound. Choose an index

$$q = \arg\min_j\{-\frac{\hat{a}_{lj}^\nu}{\hat{a}_{pj}^\nu} | \hat{a}_{lj}^\nu \geq 0, \hat{a}_{pj}^\nu < 0\} \tag{2.12}$$

and set $\mu = -\frac{\hat{a}^{\nu}_{lq}}{\hat{a}^{\nu}_{pq}}$. After pivoting on $\hat{a}^{\nu}_{pq}$, the l-th row becomes

$$(\hat{a}^{\nu}_{l0} + \mu\hat{a}^{\nu}_{p0})x^{\nu+1}_0 + (\hat{a}^{\nu}_{l1} + \mu\hat{a}^{\nu}_{p1})x^{\nu+1}_1 + \cdots + (\hat{a}^{\nu}_{ln} + \mu\hat{a}^{\nu}_{pn})x^{\nu+1}_n = \hat{b}^{\nu}_l + \mu\hat{b}^{\nu}_p \tag{2.13}$$

From the formula (2.12) we know that all left coefficients are not negative. Then

$$\delta^{\nu+1}_l = \frac{\hat{b}^{\nu+1}_l}{\hat{a}^{\nu+1}_{l0}} = \frac{\hat{b}^{\nu}_l + \mu\hat{b}^{\nu}_p}{\hat{a}^{\nu}_{l0} + \mu\hat{a}^{\nu}_{p0}} = \frac{\delta^{\nu}_l\hat{a}^{\nu}_{l0} + \mu\delta^{\nu}_p\hat{a}^{\nu}_{p0}}{\hat{a}^{\nu}_{l0} + \mu\hat{a}^{\nu}_{p0}} \tag{2.14}$$

remains still a hard upper bound.

In the above relation $\hat{a}_{pq}$ denotes the pivot element, column q is called the pivot column and row p is called the pivot row. A formal description of the primal dual algorithm in the revised form is given below.

Algorithm 2.2. Primal Dual Algorithm

Step 0. Start with a feasible basis and an index l with $\hat{a}_{lj} \geq 0, j = 0, 1, \cdots, n$. Set $p = 0$ and $\nu = 0$.

Step 1. If the p-th row becomes a dual constraint row, STOP. If $\hat{a}^{\nu}_{i0} \leq 0, i = 0, 1, \cdots, m$, STOP.

Step 2. Pivot on $\hat{a}^{\nu}_{pq}(< 0)$ with an index q according to the rule (2.12).

Step 3. Choose an index p such that

$$p = \arg\min\{\frac{\hat{b}^{\nu}_i}{\hat{a}^{\nu}_{i0}} | \hat{a}^{\nu}_{i0} > 0\} \tag{2.15}$$

Step 4. Update $\hat{b}^{\nu+1} = \hat{b}^{\nu} - \lambda^{\nu}\hat{a}^{\nu}_0$, where

$$\lambda^{\nu} = \frac{\hat{b}^{\nu}_p}{\hat{a}^{\nu}_{p0}} \tag{2.16}$$

denotes the moving step size. Set $\nu \leftarrow \nu + 1$, return to step 1.

There are two interesting properties of this algorithm. On one hand there are two ratio tests in every pivot steps - both the leaving and entering variables are selected by performing a ratio test. They are used in the primal or dual methods independently. On the other hand, primal and dual feasibilities are preserved during the algorithm due to the second and the first ratio tests, respectively - while the complementary slackness condition is lost, but recovered in the last step.

To illustrate the pivot rules fully, we consider their application to a simple example. The solution of this problem by the simplex methods demonstrates the various stages of the methods.

Example 2.3. Consider the linear programming problem

$$\begin{array}{ll} \min & -\frac{3}{4}x_4 + 20x_5 - \frac{1}{2}x_6 + 6x_7 \\ s.t. & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = \frac{1}{100} \\ & x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = \frac{1}{25} \\ & x_3 + x_4 + x_6 = 1 \\ & x_1, x_2, \cdots, x_7 \geq 0 \end{array}$$

We solve it by the algorithm (2.2). The initialized simplex schema is

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
1	0	0	0	$-\frac{3}{4}$	20	$-\frac{1}{2}$	6	0
0	1	0	0	$\frac{1}{4}$	-8	-1	9	$\frac{1}{100}$
0	0	1	0	$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	$\frac{1}{25}$
0	0	0	1	1	0	1	0	1

x_4 is to be dropped from the basic set in the next cycle in view of the rule (2.12). The new canonical system is

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{4}{3}$	0	0	0	1	$-\frac{80}{3}$	$\frac{2}{3}$	-8	$\frac{1}{25}$
$\frac{1}{3}$	1	0	0	0	$-\frac{4}{3}$	$-\frac{7}{6}$	11	0
$\frac{2}{3}$	0	1	0	0	$\frac{4}{3}$	$-\frac{5}{6}$	7	$\frac{1}{50}$
$\frac{4}{3}$	0	0	1	0	$\frac{80}{3}$	$\frac{1}{3}$	8	$\frac{24}{25}$

where the column rhs denotes the new right hand constant vector in view of the step size rule in formulas (2.16). The solution listed in the rhs column is primal feasible, and is also dual feasible according to the dual vector (2.9).

The first major cycle is complete, but since the termination criteria are not satisfied. In accordance with the statement of the algorithm (2.2), we select the nonbasic variable x_6 in place of x_1; it is, in fact, only a choices for this schema according to the pivot rules (2.12) and (2.15). The pivot $\hat{a}^1_{16}$ yields the schema

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{8}{7}$	$\frac{4}{7}$	0	0	1	$-\frac{192}{7}$	0	$-\frac{12}{7}$	$\frac{7}{75}$
$-\frac{2}{7}$	$-\frac{6}{7}$	0	0	0	$\frac{8}{7}$	1	$-\frac{66}{7}$	$\frac{1}{75}$
$\frac{3}{7}$	$-\frac{5}{7}$	1	0	0	$\frac{16}{7}$	0	$-\frac{6}{7}$	0
$\frac{10}{7}$	$\frac{2}{7}$	0	1	0	$\frac{184}{7}$	0	$\frac{78}{7}$	$\frac{67}{75}$

Analogously, the corresponding pivot $\hat{a}^2_{21}$ leads to the resulting schema

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{4}{5}$	0	$\frac{4}{5}$	0	1	$-\frac{128}{5}$	0	$-\frac{12}{5}$	$\frac{27}{50}$
$-\frac{4}{5}$	0	$-\frac{6}{5}$	0	0	$-\frac{8}{5}$	1	$-\frac{42}{5}$	$\frac{23}{50}$
$-\frac{3}{5}$	1	$-\frac{7}{5}$	0	0	$-\frac{16}{5}$	0	$\frac{6}{5}$	$\frac{67}{200}$
$\frac{8}{5}$	0	$\frac{2}{5}$	1	0	$\frac{136}{5}$	0	$\frac{54}{5}$	0

After three iterations, the algorithm terminates finding the optimal solution. The optimal solution is $(\frac{67}{200}, 0, 0, \frac{27}{50}, 0, \frac{23}{50}, 0)^T$, and the optimal value of the objective function is

$$-\frac{127}{200} = -\frac{3}{100} - \frac{7}{150} - \frac{67}{120}$$

3 Relation with classical algorithms

3.1 Classical methods

The primal dual algorithm of the previous section is very similar to the primal dual simplex algorithms presented in [1, 3]. If we omit the pivot rule (2.12) temporarily, we obtain a new primal version of simplex methods. It is also called a current equivalent facet algorithm in [5]. The formula (2.12), like Dantzig's rule, offer only a new choice of pivot rules. Of course, it is initialized with a basic solution that is primal feasible. We list all basic feasible solutions listed in the *rhs* column and their complementary dual vectors according to (2.11) in the following tableau

ν	x^ν	r^ν
0	$(\frac{1}{100}, \frac{1}{25}, 1, 0, 0, 0, 0)$	$(0, 0, 0, -\frac{3}{4}, 20, -\frac{1}{2}, 6)$
1	$(0, \frac{1}{50}, \frac{24}{25}, \frac{1}{25}, 0, 0, 0)$	$(3, 0, 0, 0, -4, -\frac{7}{2}, 33)$
2	$(0, 0, \frac{67}{75}, \frac{7}{75}, 0, \frac{1}{75}, 0)$	$(-\frac{5}{3}, \frac{7}{3}, 0, 0, \frac{16}{3}, 0, -2)$
3	$(\frac{67}{200}, 0, 0, \frac{27}{50}, 0, \frac{23}{50}, 0)$	$(0, \frac{1}{4}, \frac{5}{8}, 0, 17, 0, \frac{27}{4})$

The solutions of this example demonstrate several typical feature of the primal simplex method. Each basis x^ν is primal feasible and dual infeasible for r^ν listed in the tableau except for the last one. In additional, it is implied for the complementary conditions(2.3) to be satisfied in the primal simplex method. We can obtain their all duality gap

$r^T x$	r^0	r^1	r^2	r^3
x^0	0	$\frac{3}{100}$	$\frac{23}{300}$	$\frac{127}{200}$
x^1	$-\frac{3}{100}$	0	$\frac{7}{150}$	$\frac{121}{200}$
x^2	$-\frac{23}{300}$	$-\frac{7}{150}$	0	$\frac{67}{120}$
x^3	$-\frac{127}{200}$	$-\frac{121}{200}$	$-\frac{67}{120}$	0

Then we easily verify

$$\begin{aligned} (x^0)^T r^3 &= (x^0)^T r^1 + (x^1)^T r^3 \\ (x^1)^T r^3 &= (x^1)^T r^2 + (x^2)^T r^3 \end{aligned}$$

and so on.

The step size rule (2.16) maintains the primal feasibility of solutions in the algorithm (2.2), but it is not only one choice. If we replace it by

$$\lambda^\nu = \frac{\hat{b}^\nu}{\hat{a}^\nu_{l0}} \tag{3.1}$$

the primal feasibility is lost -while the current basis is still dual feasible according to (2.9) in the new moving step size (3.1). If we choose it, we get another dual version of simplex methods that is equivalent to classical dual simplex method in [6]. The formula (2.15) offer only a new choice of pivot rules in the dual simplex method.

We list all basic (dual feasible) solutions and their complementary dual vectors according to (2.9) in the following tableau

ν	x^ν	r^ν
0	$(\frac{1}{100}, \frac{1}{25}, 1, 0, 0, 0, 0)$	
1	$(-\frac{6}{25}, -\frac{23}{50}, 0, 1, , 0, 0, 0)$	$(0, 0, \frac{3}{4}, 0, 20, \frac{1}{24}, 6)$
2	$(0, -\frac{67}{250}, 0, \frac{101}{125}, 0, \frac{24}{125}, 0)$	$(\frac{1}{5}, 0, \frac{7}{10}, 0, \frac{92}{5}, 0, \frac{39}{5})$
3	$(\frac{67}{200}, 0, 0, \frac{27}{50}, 0, \frac{23}{50}, 0)$	$(0, \frac{1}{4}, \frac{5}{8}, 0, 17, 0, \frac{27}{4})$

where x^ν is decided by the step size (3.1) in the algorithm (2.2). Each basis x^ν is dual feasible and primal infeasible except for the last one, and the complementary slackness conditions (2.3) are satisfied. The corresponding duality gaps are listed in the following tableau

$r^T x$	r^1	r^2	r^3
x^1	0	$-\frac{6}{125}$	$-\frac{23}{200}$
x^2	$\frac{1}{125}$	0	$-\frac{67}{1000}$
x^3	$\frac{23}{1200}$	$\frac{67}{1000}$	0

Analogous we can also yield

$$\begin{aligned} (x^1)^T r^3 &= (x^1)^T r^1 + (x^1)^T r^3 \\ (x^1)^T r^3 &= (x^1)^T r^2 + (x^2)^T r^3 \end{aligned}$$

and so on.

3.2 Convergence

In this subsection we study the convergence of the algorithm (2.2). We give now a detail statements for the duality gaps and the convergence in both the primal and the dual simplex methods, including in our algorithm (2.2). Firstly, we have the following result:

Theorem 3.1. *In the (primal or dual) simplex method, if the basic solution sequence* $\{x^0, x^1, \cdots, x^K\}$ *and the corresponding reduced cost vector sequence* $\{r^0, r^1, \cdots, r^K\}$ *satisfy the complementary slackness conditions* (2.3)*, then*

(1). $(x^i)^T r^\nu = c^T x^i - c^T x^\nu$ *is satisfied for any* $0 < i, j \leq K$*;*

(2). *The increment of the objective value is* $(x^{\nu+1})^T r^\nu$ *at the* ν*-th pivot step;*

(3). $(x^i)^T r^j = (x^i)^T r^\nu + (x^\nu)^T r^j$ *is satisfied for any* $0 < i, \nu, j \leq K$*;*

(4). $(x^i)^T r^\nu = -(x^\nu)^T r^i$ *is satisfied for any* $0 < i, \nu \leq K$*;*

(5). *The total increment of the objective value is* $(x^K)^T r^0$ *after the* K*-th pivot steps.*

Proof: According to the pivot rules in the (primal or dual) simplex method, for any r^ν there are m numbers $w_1^\nu, w_2^\nu, \cdots, w_m^\nu$ satisfying

$$r^\nu = c - w_1^\nu a_1 - w_2^\nu a_2 - \cdots - w_m^\nu a_m, \quad \nu = 1, 2, \cdots, K$$

where a_j denotes j-th row of the matrix A. Set $w^\nu = (w_1^\nu, w_2^\nu, \cdots, w_m^\nu)^T$, then

$$r^\nu = c - A^T w^\nu$$

is satisfied for each ν. Left multiplied two sides of this formula by both the transposes of x^ν and x^i, respectively, it can yield

$$0 = (x^\nu)^T r^\nu = (x^\nu)^T c - (x^\nu)^T A^T w^\nu = (x^\nu)^T c - (Ax^\nu)^T w^\nu = c^T x^\nu - b^T w^\nu$$

$$(x^i)^T r^\nu = (x^i)^T c - (x^i)^T A^T w^\nu = c^T x^i - b^T w^\nu = c^T x^i - c^T x^\nu$$

Then (1) is established. After replacing i by $\nu+1$, we get the formula $(x^{\nu+1})^T r^\nu = c^T x^{\nu+1} - c^T x^\nu$. It means that (2) is correct. Furthermore, we have

$$(x^i)^T r^j = c^T x^i - c^T x^j$$

$$(x^\nu)^T r^j = c^T x^\nu - c^T x^j$$

They imply (3) is correct. Then (3) establishes (4) if we replace j by i.

At last we know that (5) is correct from (3) and (4).□

From the theorem (3.1) and its proof, we easily yield the basic theorem (2.1).

In the primal simplex method, the basis is primal feasible and dual infeasible except for the last step. At each pivot step, we always change the nonbasic variable with a negative complementary dual variable in (2.11). This results in the moving step size $x^{\nu+1} r^\nu = x_p^{\nu+1} r_p^\nu < 0$ so that the objection value is monotonically decreasing at each basis exchange in the primal simplex method. Similar cases also appear in the dual simplex algorithm. Finiteness of classical methods can be easily shown in view of the theorem (3.1) under the assumption of nondegeneracy, which leads to finiteness of the algorithm (2.2). Next we give another explain for algorithm (2.2) and prove its finiteness again.

For the example (2.3), let x_P and r_D denotes a basis in the primal simplex method and a reduced cost vector in the dual simplex method, respectively. In fact, r_D^ν is also a dual vector given according to (2.9). We get one group of duality gaps

$$(x_P^1)^T r_D^1 = \frac{18}{25}, \quad (x_P^2)^T r_D^2 = \frac{469}{750}, \quad (x_P^3)^T r_D^3 = 0$$

They are just equal to the hard upper bound δ_l^ν. On the other hand, We obtain another group of duality gaps

$$(x_P^0)^T r_D^1 = \frac{3}{4}, \quad (x_P^1)^T r_D^2 = \frac{84}{125}, \quad (x_P^2)^T r_D^3 = \frac{67}{120}$$

Those dual gaps are just equal to the hard upper bound δ_l^ν before updating the right hand constant $\hat{b}^\nu$ at each pivot step. We easily verify that the step sizes satisfy the following equations

$$\lambda_1 = (x_P^0)^T r_D^1 - (x_P^1)^T r_D^1 = \frac{3}{100}$$

$$\lambda_2 = (x_P^1)^T r_D^2 - (x_P^2)^T r_D^2 = \frac{7}{150}$$

$$\lambda_3 = (x_P^2)^T r_D^3 - (x_P^3)^T r_D^3 = \frac{67}{120}$$

Meanwhile, the objective value of the dual problem (2.2) increases

$$(x_P^1)^T r_D^2 - (x_P^1)^T r_D^1 = \frac{6}{125}$$

$$(x_P^2)^T r_D^3 - (x_P^2)^T r_D^2 = \frac{67}{1000}$$

They result in that the total duality gap decreases

$$\frac{3}{4} = (\frac{3}{100} + \frac{7}{150} + \frac{67}{120}) + (\frac{5}{125} + \frac{67}{1000})$$

We might refer r_D^ν (or $r_D^{\nu+1}$) to as the complementary dual vector of x_P^ν regardless of the constraint (2.3). In each iteration of our algorithm (2.2) a pair of basic solutions that is primal and dual feasible is reconstructed. In [6] it is shown that if at least one of the two basic solutions is nondegenerate, the duality gap decreases strictly from iteration to iteration. Consequently, stalling can occur when the two basic solutions of the pair are simultaneously degenerate.

Different with classical methods, the algorithm (2.2) maintains not only the primal feasibility but also the dual feasibility in view of (2.9), but the complementary slackness condition is lost before the last step. The following two results reveal that the hard upper bound is monotonically decreasing in the algorithm (2.2).

Corollary 3.2. *On the assumption that δ_l^ν is a hard upper bound, if the moving step size is greater than zero, then the hard upper bound decreases at the ν-th pivot step.*
Proof: It is obvious.□

Corollary 3.3. *On the assumption that a hard upper bound δ_l^ν is equal to or greater than a soft upper bound δ_p^ν, if we always choose the pivot principal element $\hat{a}_{pq}^\nu$ according to the rule* (2.12), *the hard upper bound is preserved and decrease strictly if $\hat{a}_{lq} > 0$.*
Proof: On this assumption of this theorem, if $\hat{a}_{lq} > 0$, from (2.14) we have

$$\delta_l^{\nu+1} = \frac{\delta_l^\nu \hat{a}_{l0}^\nu + \mu \delta_p^\nu \hat{a}_{p0}^\nu}{\hat{a}_{l0}^\nu + \mu \hat{a}_{p0}^\nu} < \frac{\delta_l^\nu \hat{a}_{l0}^\nu + \mu \delta_l^\nu \hat{a}_{p0}^\nu}{\hat{a}_{l0}^\nu + \mu \hat{a}_{p0}^\nu} = \delta_l^\nu$$

This complete our proof. □

Of course, we can not compel the hard upper bound to change if $\hat{a}_{lq} = 0$. A dual constraint (2.8) is dual degeneracy if some dual variables of the non-basic variables appear zeros. In additional, a basis is primal degeneracy if at least one of its basic variables appear zeros.

Finiteness of the algorithm, under the assumption that either the primal or the dual feasible solution is nondegenerate, is easily shown.

Theorem 3.4. *Under the assumption that the primal or dual feasible solution is nondegenerate, the algorithm* (2.2) *convergent to the optimal solution of* (2.1) *in the finite steps.*
Proof: From the corollary (3.2) and (3.3), a sequence of hard upper bound δ_l^ν is strict monotonically decreasing as ν. Then the finiteness of the extreme points results in the algorithm (2.2) terminates in the finite steps.□

4 Enhancements

We find it convenient to drive other new versions of our algorithms using slight modification of the algorithm (2.2).

The algorithm (2.2) needs the strong initialized conditions-both primal and dual feasible. Those conditions can achieve so hard that they might lead to applying it difficult. The acquisition of the initialized primal feasible solution requires the use of the two phase method or big-M method. If there is not a dual constraint row, we can add an additional dual constraint row (2.8) ($l = m+1$) to the simplex tableau. This process is very similar to the additional artificial constraint

$$x_1 + x_2 + \cdots + x_n \leq M$$

added to the tableau in the dual simplex method, where M is enough big positive number. We obtain a modification of the algorithm (2.2).

Algorithm 4.1. Modified Algorithm (I)

Step 0. Start with an initialized feasible basis. Set $p = 0$ and $\nu = 0$. Add an additional dual constraint row

$$\hat{a}^0_{m+1,1}x_1 + \hat{a}^0_{m+1,2}x_2 + \cdots + \hat{a}^0_{m+1,n}x_n \leq M \tag{4.1}$$

to the tableau, where each $\hat{a}^0_{m+1,j}$ is not negative at $j = 1, 2, \cdots, n$. Set $l = m + 1$.

Step 1 $\sim$ Step 5. The same as in the primal dual algorithm.

It is clear that the modified algorithm determines a unique pivot path if no degeneracy occurs. Actually, in case of degeneracy it has more flexibility than other rules. Different with classical dual methods, we do not care about how big M is, and not introduce a artificial variable in the modified algorithm (I). In additional, a dual constraint row can be replaced by a almost dual facet only if we choose an index l such that

$$l = \arg\min_i \{ \sum_{j\in J(i)^-} sgn(\hat{a}^\nu_{ij}) | \hat{a}^\nu_{i0} \geq 0, i \neq p\} \tag{4.2}$$

or

$$l = \arg\min_i \{ \sum_{j\in J(i)^-} sgn(\hat{a}^\nu_{ij}) - \sum_{j\in J(i)^+} sgn(\hat{a}^\nu_{ij}) | \hat{a}^\nu_{i0} \geq 0, i \neq p\} \tag{4.3}$$

where $J(i)^- = \{j | \hat{a}^\nu_{ij} < 0\}$ and $J(i)^+ = \{j | \hat{a}^\nu_{ij} > 0\}$ denote two index sets and sgn denotes the signum function.

Algorithm 4.2. Modified Algorithm (II)

Step 0a. Start with an initialized feasible basis. Set $p = 0$ and $\nu = 0$.

Step 0b. Choose an index l according to (4.3).

Step 2a. Choose an index q according to (2.12).

Step 2b. If there is not any index l or q, pivot on $\hat{a}_{p0}$ and go to Step 0b.

Step 1, Step 3 $\sim$ Step 5. The same as in the primal dual algorithm.

We can choose coefficients in the additional dual constraint row (4.1) such that

$$\hat{a}^{\nu}_{m+1,j} = \begin{cases} M_1 & j \in J(l)^- \\ \hat{a}_{lj} & otherwise \end{cases} \tag{4.4}$$

where l is defined as (4.2) or (4.3), and M_1 is a suitable positive number.

It seems easily that the next step is to extend our primal dual algorithm to general exterior point algorithm. Different from the algorithm (2.2), the new algorithm might start with any basic solution. We illustrate it only by applying it to the following linear problem. A detail discussion will be study in the future paper.

Example 4.3. Consider the linear programming problem

$$\begin{array}{ll} \min & -7x_4 + 10x_5 - 6x_6 \\ s.t. & x_1 + 4x_4 + x_5 - 2x_6 + x_7 = -9 \\ & x_2 - 2x_4 + x_5 + x_6 - 4x_7 = 4 \\ & x_3 - x_4 + 2x_5 + x_6 + 2x_7 = 1 \\ & x_1, x_2, \cdots, x_7 \geq 0 \end{array}$$

It is easily seen that the initialized basic solution is not feasible because the first component of x is negative. For this reason we apply the exterior point algorithm. The first schema is given as the following form

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
1	0	0	0	-7	10	-6	0	0
0	1	0	0	4	1	-2	1	-9
0	0	1	0	-2	1	1	-4	4
0	0	0	1	-1	2	1	2	6

Note that the primal test row appears in the first row ($p = 0$), and the almost dual constraint facet appears in fourth row ($l = 3$) because of only one negative coefficient -1. In view of the rule (2.12), we can make x_6 basis in place of x_0, to yield the following schema

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{1}{6}$	0	0	0	$\frac{7}{6}$	$-\frac{5}{3}$	1	0	4
$-\frac{1}{3}$	1	0	0	$\frac{19}{3}$	$-\frac{7}{3}$	0	1	-1
$\frac{1}{6}$	0	1	0	$-\frac{19}{6}$	$\frac{8}{3}$	0	-4	0
$\frac{1}{6}$	0	0	1	$-\frac{13}{6}$	$\frac{11}{3}$	0	2	2

From the rule (2.15) we know that the third row ($p = 2$) becomes the primal test row and the fourth row ($l = 3$) is still almost dual constraint row. We get another schema

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{1}{6}$	0	0	0	$\frac{7}{6}$	$-\frac{5}{3}$	1	0	$\frac{16}{3}$
$-\frac{7}{24}$	1	$\frac{1}{4}$	0	$\frac{133}{24}$	$-\frac{5}{3}$	0	0	$\frac{4}{3}$
$-\frac{1}{24}$	0	$-\frac{1}{4}$	0	$\frac{19}{24}$	$-\frac{2}{3}$	0	1	$\frac{1}{3}$
$\frac{1}{4}$	0	$\frac{1}{2}$	1	$-\frac{15}{4}$	5	0	0	0

From the rule (2.15) we find that the fourth row ($p = 3$) becomes the primal test row. But we can not find a almost dual constraint row because the set $\{i | \hat{a}_{ij} > 0, j \neq p\}$ is empty. The pivot $\hat{a}_{34} (< 0)$, in fact it is only a choice, leads to the following schema

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	rhs
$-\frac{4}{45}$	0	$\frac{7}{45}$	$\frac{14}{45}$	0	$-\frac{1}{9}$	1	0	$\frac{48}{7}$
$\frac{7}{90}$	1	$\frac{89}{90}$	$\frac{133}{90}$	0	$\frac{103}{18}$	0	0	0
$\frac{1}{90}$	0	$-\frac{13}{90}$	$\frac{19}{90}$	0	$\frac{7}{18}$	0	1	$\frac{1}{7}$
$-\frac{1}{15}$	0	$-\frac{2}{15}$	$-\frac{4}{15}$	1	$-\frac{4}{3}$	0	0	$\frac{8}{7}$

After three iterations, the primal test row is dual constraint face such that the exterior point algorithm terminates finding the optimal solution $(0, 0, 0, \frac{8}{7}, 0, \frac{48}{7}, \frac{1}{7})^T$ with the objective value $\frac{344}{7}$.

5 Numerical test

In this section we present computational results comparing the primal dual simplex algorithm and the classical primal simplex algorithm that use the most negative reduced cost rule. We have chosen to use the following linear programming problems in inequality form

$$\min\ c^T x \quad s.t. \quad Ax \leq 0,\ x_1 + x_2 + \cdots + x_n \leq 1,\ x \geq 0 \tag{5.1}$$

where the number of the constraint inequalities is $m + 1$ in total. These are chosen for two reasons. Firstly, we have an obvious initial solution $(0, 0, \cdots, 0)$, since the basis is always feasible. Secondly, it is possible to generate cycling in the classical primal simplex algorithm because of the serious degeneracy of (5.1). We test these problems so that we realize the advantage of the algorithm (2.2) in Section 3.

For the degenerate problems (5.1), it is possible that there are several current equivalent faces in our algorithm from [5, 6] on the premise that the primal feasibility is maintained. We choose one of them in which the left hand side contains the most nonnegative (or positive) coefficients.

All test results reported here were obtained on an intel (R) Pentium(R) 4 CPU 2.00GHz personal computer, RAM 256 Mb and with the windows XP operating system. Our implementation was done under the MATLAB environment. MATLAB offers the appropriate environment for programming quick this kind of algorithms.

This is how we generated our test problems. For each (m, n), we constructed ten random instances, making 70 instances in total. The left hand side coefficient matrix is dense. We find that more complex instances (e.g., with more constraints and variables) cause the classical primal simplex method to run into numerical difficulties and crash.

The results with the algorithm (2.2) are displayed at the left hand in Table 1. The pair (m, n) denotes the numbers of both constraints and variables. After we set a limit of one million, the following two columns show the both minimum and maximum number of successful iterations. The column 'Ave' shows the average number of successful iterations. The right hand in Table 1 shows the analogous

Table 1: Computational results by the modern and classical algorithms

algorithm (2.2)				*primal simplex method*			
(m, n)	Min	Max	Ave	Min	Max	Ave	$Cycling$
(30, 30)	8	62	22.8	51	225	133.2	1
(40, 30)	14	70	43.9	84	690	200.4	1
(40, 40)	15	77	37.6	131	1118	494.4	1
(50, 40)	18	133	65.7	108	1087	414.75	6
(50, 50)	19	83	41.1	750	3090	1762.3	3
(60, 50)	26	211	99.1	290	2240	1447.4	4
(60, 60)	25	253	93.3	568	10265	3218.3	6

figures for the classical primal simplex method, but there is an extra column showing the number that cycling happens in ten random instances. For example, in the first row there is only one instance with numerical problem due to cycling.

Preliminary numerical examples indicate that the new algorithm is potentially effective for the problems (5.1) than the classical primal simplex method. It is worth investigating whether it is suit for large scale linear problems.

6 Conclusions

In this paper we have presented several computationally attractive approach to solving linear programming problems. In the previous section we have addressed the most issue of an efficient implementation of our algorithm. In all test problems, the algorithm (2.2) finds an optimal solution in the average number of iterations that is less than $m + n$. In our opinion this is a good practical performance that gives much promise to the approach.

Thus far the duality gaps we have been considering have been used in our algorithms. The increment model of (2.1) offer us the more dual information about the current solution. This turns out to be a useful class of gaps, and not only because a very satisfactory theory can be developed for it. As we have been seen, the duality gap property is a reasonable and verifiable hypothesis in a wide range of applications, and one that is satisfied in particular when both the primal and dual feasibility are present.

The experimental implementation in randomly generated LPs proved that our algorithms is very fast; there is still room for this further improvement. More generally, most of the techniques applied in advanced implementations of simplex algorithm can probably, with some changes, be incorporated into the exterior point simplex algorithm. The possible improvements will be the subject of our future work.

References

[1] H.Chen, P.M.Paralos and M.A.Saunder, (1994)The simplex algorithm with a new primal and dual pivot rule, Oper. Res. Lett. 16,121-127.

[2] G. B. Dantzig, (1963) Linear Programming and Extensions, Princeton University Press, Princeton, New York.

[3] K.Paparrizos, N.Samaras and G.Stephanides, (2003)A new efficient primal dual simplex algorithm, 1383-1399.

[4] K.Fukuda and T.Terlaky, (1992)Criss-Cross Methods: A Fresh View on Pivot Algorithms, Mathematical Programming, (Series B) 79, 369-396.

[5] Yan Zizong and Fei Pusheng, (2004)Current Equivalent Facet Algorithm for Linear Programming Problem, Mathematica numerica sinica(In Chinese), Vol.26, No.4,437-444.

[6] Yan Zizong, Fei Pusheng and Wang Xiaoli, (2004) Basic Theory of Current Equivalent Facet Algorithm for Linear Programming Problem (In Chinese), Journal of Yangtze university, to appear