

New Optimal Pivot Rule for the Simplex Algorithm

Jean Bosco Etoa Etoa

Department of Economic and Management Sciences, University of Yaounde II, Soa, Cameroon

Email: jbetao_etoa@hotmail.com, jbetao@gmail.com

How to cite this paper: Etoa, J.B.E. (2016)

New Optimal Pivot Rule for the Simplex Algorithm. *Advances in Pure Mathematics*, 6, 647-658.

<http://dx.doi.org/10.4236/apm.2016.610054>

Received: May 14, 2016

Accepted: September 13, 2016

Published: September 16, 2016

Copyright © 2016 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The purpose of this paper is to introduce a new pivot rule of the simplex algorithm. The simplex algorithm first presented by George B. Dantzig, is a widely used method for solving a linear programming problem (LP). One of the important steps of the simplex algorithm is applying an appropriate pivot rule to select the basis-entering variable corresponding to the maximum reduced cost. Unfortunately, this pivot rule not only can lead to a critical cycling (solved by Bland's rules), but does not improve efficiently the objective function. Our new pivot rule 1) solves the cycling problem in the original Dantzig's simplex pivot rule, and 2) leads to an optimal improvement of the objective function at each iteration. The new pivot rule can lead to the optimal solution of LP with a lower number of iterations. In a maximization problem, Dantzig's pivot rule selects a basis-entering variable corresponding to the most positive reduced cost; in some problems, it is well-known that Dantzig's pivot rule, before reaching the optimal solution, may visit a large number of extreme points. Our goal is to improve the simplex algorithm so that the number of extreme points to visit is reduced; we propose an optimal improvement in the objective value per unit step of the basis-entering variable. In this paper, we propose a pivot rule that can reduce the number of such iterations over the Dantzig's pivot rule and prevent cycling in the simplex algorithm. The idea is to have the maximum improvement in the objective value function: from the set of basis-entering variables with positive reduced cost, the efficient basis-entering variable corresponds to an optimal improvement of the objective function. Using computational complexity arguments and some examples, we prove that our optimal pivot rule is very effective and solves the cycling problem in LP. We test and compare the efficiency of this new pivot rule with Dantzig's original pivot rule and the simplex algorithm in MATLAB environment.

Keywords

Linear Programming, Simplex Algorithm, Pivot Rules, Optimal Pivot Rule

1. Introduction

Linear programming (LP) has been one of the most dynamic areas of applied mathe-

matics in the last sixty years. LP was solved in the late 1960s by Dantzig's simplex method [1]. But, many variants of the simplex method were eventually proved to have exponential worst-case performance [2]. To solve efficiently a LP problem, we need to consider the pivot rule and the computational complexity that depend on the number of constraints and variables. One of the important steps of the simplex algorithm is of course the pivot rule that is used for selecting the basis-entering variable. An effective rule consists of computing the optimal solution of a LP with a small number of iterations. Dantzig's simplex method still seems to be the most efficient procedure for a great majority of practical problems, especially for small size problems. But Dantzig's original pivot rule cannot prevent cycling in linear programming and takes a lot of iterations in some cases [3]. To prevent this weakness, many research studies tried to improve the simplex algorithm, via the pivot rule by reducing the number of iterations and the solution time [4]-[6]. Unfortunately, most papers concerning simplex pivot rules have not been receiving much attention, even among researchers in the field of linear programming. Moreover, a very large part of these researches was presented in terms of oriented matroid programming and frequently not specialized to pivot rules for linear programming. Also, some of the other results were obtained as a side result (extreme case) of some interior point methods. Due to this, a lot of results remained unknown to researchers only working on the simplex method. T. Terlaky and S. Zhang [7] discussed the various pivot rules of the simplex method and its variants that have been developed until 1993, starting from the appearance of Bland's minimal index rules [8]. Their paper was mainly concerned with finiteness properties of simplex type pivot rules. Also there are rich research results concerning pivot rules for specially structured linear programming problems, like network linear programming, assignment problems, etc. Most recently, K. Chankong *et al.* [9] proposed a new pivot rule called absolute change pivot rule. The idea is trying to block a basis-leaving variable that makes a little change in the objective function value as much as possible. Some computational results are reported, comparing the number of iterations from this new rule to Dantzig's original pivot rule.

In this paper, we propose an original pivot rule called ***optimal pivot rule***. The idea is to have an optimal improvement of the value of the objective function for any iteration: from the variables with positive reduced cost, we have a set of basis-entering variables; the efficient basis-entering variable is chosen from this set and corresponds to an optimal improvement of the objective function; this makes the objective function value to increase faster than when a regular Dantzig's pivot rule is used, and therefore lead to fewer number of iterations. The optimal pivot rule can prevent cycling in the simplex algorithm. We report the computational results by testing and comparing the number of iterations from this new rule to Dantzig's original pivot rule in MATLAB environment.

The rest of the paper is organized as follows: Section 2 describes the preliminaries of linear programming, simplex algorithm and pivot rule. Section 3 explains the main idea of our optimal pivot rule; we show that the new pivot rule prevents cycling in simplex algorithm. We use simple computational complexity facts to prove that the new optimal pivot rule is efficient. Section 4 deals with the computational results by testing and

comparing the speed and the number of iterations from this new pivot rule to classical simplex rule and conclusions drawn.

2. Preliminaries: Dantzig's Pivot Rule

In this paper, we consider the linear programming (LP) problem in the standard form:

$$\begin{aligned} & \text{Maximize } c^T x \\ & \text{subject to } \begin{cases} Ax = b \\ x \geq 0, \end{cases} \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ ($m < n$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $\text{rank}(A) = m$.

After possibly rearranging the column of A , let $A = [A \ B]$ where B is an $m \times m$ invertible matrix and N is $m \times (n - m)$ matrix. Here, B is called the basic matrix and N the associated non basic matrix. Basic and non basic index set are represented by J_B and J_N respectively. Consider the equation $Ax = b$, and let $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$ be the solution where $x_B = B^{-1}b$ and $x_N = 0$ is called a basic solution of the system. The constraints

$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

can be rewrite as

$$\begin{cases} Bx_B + Nx_N = b \\ x_B \geq 0, x_N \geq 0. \end{cases} \tag{2}$$

If $x_B \geq 0$, x is called a basic feasible solution of the system. Suppose that a basic feasible solution of the system (1) is $\begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix}$ whose objective value z_0 is given by

$$z_0 = c_B^T B^{-1}b.$$

Then

$$x_B = \bar{b} - \sum_{j \in J_N} Y_j x_j$$

where $\bar{b} = B^{-1}b$ and $Y_j = B^{-1}A_j$. We denote the j^{th} column of A by A_j and the j^{th} column of $B^{-1}N$ by Y_j .

Let z be the objective function value, we get

$$z = z_0 - \sum_{j \in J_N} \bar{c}_j x_j$$

where $\bar{c}_j = c_j - z_j$ represents the reduced cost, with

$$z_j = c_B^T B^{-1}A_j = c_B^T Y_j.$$

The main result exhibits that the optimal solution is achieved if the index set

$$J = \{j \in J_N / \bar{c}_j > 0\}$$

is empty. If the index set J is not empty, let

$$e = \arg \max \{\bar{c}_j / j \in J\}.$$

If the index set

$$\{i \in \{1, 2, \dots, m\} / Y_{(i,e)} > 0\}$$

is empty, then the LP (1) is not bounded, and it has no solution.

Let

$$s = \arg \min \left\{ \frac{\bar{b}_i}{Y_{(i,e)}}, Y_{(i,e)} > 0 \right\}.$$

By Dantzig’s rules, the index of the basis-entering variable is e and the index of basis-leaving variable is s . The pivot operation uses $Y_{(s,e)}$.

The tableau format of the simplex method follows:

Table 1 format reports the value of the objective function $z_0 = c_B^T B^{-1} b$, the basis variables $x_B = B^{-1} b$, the reduced cost row, which consist of

$$\bar{c}_j = c_j - z_j = c_j - c_B^T B^{-1} A_j$$

for non basic variables. $\forall j, \bar{c}_j > 0$, the LP is at optimal solution. If x_e increases, then the vector $Y_e = B^{-1} A_e$, which is stored in the tableau in row 1 through m under variable x_e , will determine how much x_e can increase. If $Y_e \leq 0$, then x_e can be increased indefinitely, and the optimal objective value is unbounded. Conversely, if $Y_e \not\leq 0$, that is, if Y_e has at least one positive component, then the increase in x_e , from a pivot rule on $Y_{(s,e)} > 0$ results to an increase of the value of the objective function. The optimal pivot rule determines the non basic variable x_e , and the pivot $Y_{(s,e)}$ that compute the optimal increase of the value of the objective function.

In Bland’s Rule, choose the basis-entering variable x_e , such that e is the smallest index with $j \in J_N$. Also choose the basis-leaving variable index s with the smallest index (in case of ties in the ratio test). This rule solves the cycling problem.

3. Optimal Pivot Rules

A key factor in the performance of the simplex method is the rule used to decide which index j (with $\bar{c}_j > 0$) should enter in the basis after each pivot. It is well-known that the time spent in checking $\bar{c}_j > 0$, for each j , is $O(m)$, and if we check all possible j s, the total time is at most $O(km)$. This compares with the $O(km)$ time needed to complete the rest of the pivot, where k is the number m of pivots performed since we last computed B^{-1} .

However, the selection of a pivot rule not only will affect the performance of each

Table 1. The simplex tableau format.

c	c_B	c_N		
	x_B	x_N	-	
c_B	I_m	$Y = B^{-1} A_N$	$\bar{b} = B^{-1} b$	Rows 1 through m
z	0	$c_B^T Y_j$	$z_0 = c_B^T B^{-1} b$	Row $m + 1$
\bar{c}_j	0	$c_j - z_j$	-	

pivot, but also the total number of pivots needed to reach the optimum (if it exists). For each j (with $\bar{c}_j > 0$), the time spent in checking a pivot

$$t = \arg \min \left\{ \frac{\bar{b}_i}{y_{(i,j)}}, y_{(i,j)} > 0 \right\}$$

is $O(m)$. For all possible j 's, the total time to check a most $(n - m)$ pivots is $O(mn)$. At each step in a simplex algorithm, pivoting requires the most important computing time; it consists to compute B^{-1} . This requires a $O(m^3)$ time, which is greater than the total number of time to check all possible j 's pivots. Reducing the number of pivots (number of iterations in the simplex algorithm) accelerate the speed of the simplex algorithm to compute an optimal solution when exists. One may argue that this optimal pivot rule needs even more computation. The efficiency of the optimal pivot rule results from this simple computational complexity fact.

Let

$$\theta_{(t,j)} = \frac{\bar{b}_i}{y_{(t,j)}}.$$

The simplex algorithm with optimal pivot rule follows.

Step 1. Let $e = \arg \max \{ \bar{c}_j, j \in J_N \}$. Stop the algorithm if:

- 1) $\bar{c}_e \leq 0$, or all $j \in J_N$, then $[x_{B_k}, x_{N_k}]$ is an optimal solution.
- 2) if $\exists j \in J_N / \bar{c}_j > 0$ and $y_{(i,j)} \leq 0$ for all $i = 1, 2, \dots, m$, the LP is not bounded.

Stop the algorithm.

Step 2. Determine the basis-entering and the basis-leaving variables by using optimal change pivot rule:

For all $j \in J_N$ (with $\bar{c}_j > 0$), let $J_{N0} \subseteq J_N$ such as $j \in J_{N0}$ if

$$t = \arg \min \left\{ \frac{\bar{b}_i}{y_{(i,j)}}, y_{(i,j)} > 0 \right\}$$

exists.

Let

$$\theta_{(s,e)} = \max \left\{ \bar{c}_j \theta_{(t,j)} : j \in J_{N0} \text{ and } t = \arg \min \left\{ \frac{\bar{b}_i}{y_{(i,j)}}, y_{(i,j)} > 0 \right\} \right\}.$$

The index of the basis-entering variable is e and the index of basis-leaving variable is s .

Step 3. Perform the pivot operation using the basis-entering and the basis-leaving variable, and go to **Step 1**.

Definition: A pivot is degenerate if the objective function value does not change from 2 consecutives pivots. A cycle is a sequence of pivots that returns to the dictionary from which the cycle began.

Note: Every pivot in a cycle must be degenerate.

Theorem 1 (termination with optimal pivot rule) If the simplex method uses optimal pivot rule, it terminates in finite time with optimal solution, and more over there is no cycling.

Proof: Suppose the simplex method is implemented with optimal pivot rule and consider two consecutive bases B_k and $B_{k+1}, k \geq 1$. Let $J_{N_k} = \{j \in J_N / \bar{c}_j > 0\}$ be the set of variables with positive reduced cost. Any $x(j), j \in J_{N_k}$ can improve the value of the objective function. Let $t = \arg \min \left\{ \frac{\bar{b}_i}{y_{(i,j)}}, y_{(i,j)} > 0 \right\}$ and $\theta_{(t,j)} = \frac{\bar{b}_i}{y_{(t,j)}}$. The improvement of the objective function is $\bar{c}_j \theta_{(t,j)}$. If the solution is degenerate, we have $\theta_{(t,j)} = 0$ for some $j \in J_{N_k}$, in particular if $j = \arg \max \{\bar{c}_j / j \in J_{N_k}\}$ (Dantzig's rule); what causes cycling.

$$\text{Now, let } \theta_{(s,e)} = \max \left\{ \bar{c}_j \theta_{(t,j)} : j \in J_{N_k} \text{ and } t = \arg \min \left\{ \frac{\bar{b}_i}{y_{(i,j)}}, y_{(i,j)} > 0 \right\} \right\}.$$

If $\theta_{(s,e)} \bar{c}_e > 0$ and let $c_{B_k} x_{B_k}$ and $c_{B_{k+1}} x_{B_{k+1}}$ the values of the objective function and the corresponding solutions x_{B_k} and $x_{B_{k+1}}$. We have $c_{B_{k+1}} x_{B_{k+1}} = c_{B_k} x_{B_k} + \theta_{(s,e)} \bar{c}_e \Rightarrow c_{B_k} x_{B_k} < c_{B_{k+1}} x_{B_{k+1}}$. The solutions from two consecutive bases B_k and $B_{k+1}, k \geq 1$ cannot remain the same and there is no possible cycle.

Conjecture 1. If $\theta_{(s,e)} \bar{c}_e = 0$, necessarily $\theta_{(t,j)} \bar{c}_j = 0 \forall j \in J_{N_k} \Rightarrow \bar{b}_i = 0$ and the current solution can never be improved by the simplex algorithm. Hence the LP does not have a solution.

An illustration of the Optimal Pivot Rule

The proposed pivot rule is shown with two examples.

Example 1. Beale's cycling problem

Consider the following linear programming problem:

$$\begin{aligned} \text{maximize } z &= \frac{3}{4}x_1 - 150x_2 + \frac{1}{50}x_3 - 6x_4 \\ \text{subject to } &\begin{cases} \frac{1}{4}x_1 - 60x_2 - \frac{1}{25}x_3 + 9x_4 \leq 0 \\ \frac{1}{2}x_1 - 90x_2 - \frac{1}{50}x_3 + 3x_4 \leq 0 \\ x_3 \leq 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases} \end{aligned}$$

Here, in **Table 2**, $\max \bar{c}_j = \bar{c}_1 = \frac{3}{4}$; but $\bar{c}_1 \theta_{(1,1)} = \bar{c}_1 \theta_{(2,1)} = 0$.

Table 2. The initial simplex tableau (example 1).

	x	x_1	x_2	x_3	x_4	x_1	x_2	x_3	\bar{b}
x_B	c_B	3/4	-150	1/50	-6	0	0	0	
x_1	0	1/4	-60	-1/25	9	1	0	0	0
x_2	0	1/2	-90	-1/50	3	0	1	0	0
x_3	0	0	0	1	0	0	0	1	1
z_j		0	0	0	0	0	0	0	0
$\bar{c}_j = c_j - z_j$		3/4	-150	1/50	-6	0	0	0	

The optimal pivot rule: $\theta_{(\bar{3},3)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,3)}}, Y_{(i,3)} > 0 \right\}$ and $\bar{c}_3 \theta_{(\bar{3},3)} = 1/50 > 0$. The basis-entering variable is x_3 and the basis-leaving variable is $x_{\bar{3}}$.

From **Table 3**, we have only $\bar{c}_2 > 0$. $\theta_{(\bar{2},2)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,2)}}, Y_{(i,2)} > 0 \right\}$. The basis-entering variable is x_2 and the basis-leaving variable is $x_{\bar{2}}$.

After only 3 iterations, we have the optimal solution on **Table 4** with no cycling. 7 iterations are required to solve this problem with Bland’s pivot rules.

Example 2. Consider the following LP program

$$\text{maximize } z = 4x_1 + 5x_2 + 3x_3$$

$$\text{subject to } \begin{cases} x_1 + x_4 = 1000 \\ x_2 + x_5 = 500 \\ x_3 + x_6 = 1500 \\ 3x_1 + 6x_2 + 2x_3 + x_7 = 6700 \\ x_i \geq 0, i = 1, 2, 3, 4. \end{cases}$$

We solve this LP using optimal pivot rule.

$$\text{Here, } \max \bar{c}_j = \bar{c}_2 = 5; \theta_{(5,2)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,2)}}, Y_{(i,2)} > 0 \right\} = 500 \text{ and } \bar{c}_2 \theta_{(5,2)} = 2500.$$

The optimal pivot rule on **Table 5**:

Table 3. The second simplex tableau (example 1).

	x	x_1	x_2	x_3	x_4	$x_{\bar{1}}$	$x_{\bar{2}}$	$x_{\bar{3}}$	\bar{b}
$x_{\bar{3}}$	$c_{\bar{3}}$	3/4	-150	1/50	-6	0	0	0	
$x_{\bar{1}}$	0	1/4	-60	0	9	1	0	1/25	1/25
$x_{\bar{2}}$	0	1/2	-90	0	3	0	1	1/50	1/50
x_3	1/50	0	0	1	0	0	0	0	1
	z_j	0	0	1/50	0	0	0	0	1/50
	$\bar{c}_j = c_j - z_j$	3/4	-150	0	-6	0	0	0	

Table 4. The optimal simplex tableau (example 1).

	x	x_1	x_2	x_3	x_4	$x_{\bar{1}}$	$x_{\bar{2}}$	$x_{\bar{3}}$	\bar{b}
$x_{\bar{3}}$	$c_{\bar{3}}$	3/4	-150	1/50	-6	0	0	0	
$x_{\bar{1}}$	0	0	-15	0	15/2	1	-1/2	3/100	3/100
x_1	3/4	1	-180	0	6	0	2	1/25	1/25
x_3	1/50	0	0	1	0	0	0	0	1
	z_j	3/4	-135	1/50	9/2	0	3/2	3/100	1/20
	$\bar{c}_j = c_j - z_j$	0	-15	0	-21/2	0	-3/2	-3/100	

$$\bar{c}_1 = 4; \theta_{(4,1)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,1)}}, Y_{(i,1)} > 0 \right\} = 1000 \quad \text{and} \quad \bar{c}_1 \theta_{(4,1)} = 4000.$$

$$\bar{c}_3 = 3; \theta_{(6,3)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,3)}}, Y_{(i,3)} > 0 \right\} = 1500 \quad \text{and} \quad \bar{c}_3 \theta_{(6,3)} = 4500.$$

$$\bar{c}_3 \theta_{(6,3)} = \max \{ \bar{c}_1 \theta_{(4,1)}, \bar{c}_2 \theta_{(5,2)}, \bar{c}_3 \theta_{(6,3)} \} = 4500$$

The basis-entering variable is x_3 and the basis-leaving variable is x_6 (the basis-entering variable corresponds here to the minimal reduced cost, but with an optimal growth of the value of the objective function).

Using the classical simplex pivot rule, the basis-entering variable is x_3 (corresponding to the maximal reduced cost) and the basis-leaving variable is x_5 . The increase of the objective function is 2500.

$$\bar{c}_1 = 4; \theta_{(4,1)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,1)}}, Y_{(i,1)} > 0 \right\} = 1000 \quad \text{and} \quad \bar{c}_1 \theta_{(4,1)} = 4000.$$

$$\bar{c}_2 = 5; \theta_{(5,2)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,2)}}, Y_{(i,2)} > 0 \right\} = 3700/6 \quad \text{and} \quad \bar{c}_2 \theta_{(5,2)} = \frac{18500}{6}.$$

$$\bar{c}_1 \theta_{(4,1)} = \min \{ \bar{c}_1 \theta_{(4,1)}, \bar{c}_2 \theta_{(5,2)} \} \quad (\text{see Table 6}).$$

The basis-entering variable is x_1 and the basis-leaving variable is x_4 (the basis-entering variable corresponds here to the minimal reduced cost, but with an optimal growth

Table 5. The initial simplex tableau (example 2).

	x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	\bar{b}
x_B	c_B	4	5	3	0	0	0	0	
x_4	0	1	0	0	1	0	0	0	1000
x_5	0	0	1	0	0	1	0	0	500
x_6	0	0	0	1	0	0	1	0	1500
x_7	0	3	6	2	0	0	0	1	6700
	z_j	0	0	0	0	0	0	0	0
	$\bar{c}_j = c_j - z_j$	4	5	3	0	0	0	0	

Table 6. The second simplex tableau (example 2).

	x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	\bar{b}
x_B	c_B	4	5	3	0	0	0	0	
x_4	0	1	0	0	1	0	0	0	1000
x_5	0	0	1	0	0	1	0	0	500
x_6	3	0	0	1	0	0	1	0	1500
x_7	0	3	6	0	0	0	-2	1	3700
	z_j	0	0	3	0	0	3	0	4500
	$\bar{c}_j = c_j - z_j$	4	5	0	0	0	-3	0	

of the value of the objective function). Then, we have **Table 7**.

$$\bar{c}_2 = 5 > 0; \theta_{(7,2)} = \min \left\{ \frac{\bar{b}_i}{Y_{(i,2)}}, Y_{(i,2)} > 0 \right\} = 700/6. \text{ The basis-entering variable is } x_2$$

and the basis-leaving variable is x_7 .

We have an optimal solution on **Table 8** after 3 iterations.

$$\bar{x}_1 = 1000, \bar{x}_2 = \frac{350}{3}, \bar{x}_5 = \frac{1150}{3}, \bar{x}_6 = 1500 \text{ and } \bar{z} = 27250/3.$$

Dantzig's pivot rule computed the optimal solution of this LP with 6 iterations.

4. Computational Experiments

In this section, we present the computational results of modified simplex algorithm with optimal pivot rule. Optimal pivot rule was tested by solving randomly generated linear programming problems of various sizes using the MATLAB codes. We compare the number of iterations of this pivot rule with Dantzig's pivot rule. The computer system processor is Intel (R) Core (TM) i7 3770S CPU @ 3.1 GHz, 8.00 GB of memory, and 64-bit Window 8.1 Operating System.

4.1. Problem Generation

For LP problems considered here, data are randomly generated using MATLAB generator. We consider the LP problem whose formulation is given by

Table 7. The third simplex tableau (example 2).

	x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	\bar{b}
x_B	c_B	4	5	3	0	0	0	0	
x_1	4	1	0	0	1	0	0	0	1000
x_5	0	0	1	0	0	1	0	0	500
x_6	3	0	0	1	0	0	1	0	1500
x_7	0	0	6	0	-3	0	-2	1	700
z_j		4	0	3	4	0	3	0	8500
$\bar{c}_j = c_j - z_j$		0	5	0	-4	0	-3	0	

Table 8. The optimal simplex tableau (example 2).

	x	x_1	x_2	x_3	x_4	x_5	x_6	x_7	\bar{b}
x_B	c_B	4	5	3	0	0	0	0	
x_1	4	1	0	0	1	0	0	0	1000
x_5	0	0	0	0	1/2	1	1/3	-1/6	1150/3
x_6	3	0	0	1	0	0	1	0	1500
x_2	5	0	1	0	-1/2	0	-1/3	1/6	350/3
z_j		4	5	3	3/2	0	4/3	5/6	27,250/3
$\bar{c}_j = c_j - z_j$		0	0	0	-3/2	0	-4/3	-5/6	

$$\begin{aligned} & \text{Maximize } c^T x \\ & \text{subject to } \begin{cases} Ax \leq b \\ x \geq 0, \end{cases} \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$ ($m < n$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $\text{rank}(A) = m$.

We use MATLAB generator to build all data: SPRAND ($m, n, \text{density}$) is a random, m -by- n , sparse matrix with approximate density $m \times n$ uniformly distributed nonzero entries. The density used is $p\%$.

$$\begin{aligned} A &= \text{round}(\text{sprandn}(m, n, p)) + 0.5 \\ p &= 80 \text{ when } n < 500 \text{ and } p = 50 \text{ when } n \geq 500. \end{aligned}$$

The data of b are generated according to RANDN ($1, m$) which is a vector with random entries. The data of c are generated according to RANDN ($n, 1$) which is vector with random entries.

$$b[1:m-1] = \text{abs}(\text{round}(1.5 \times \text{randn}([1, m]))) + 1$$

and

$$c = \text{abs}(\text{round}(2 \times \text{randn}([1, n]))) - 1.$$

We add an additional ones entries constraint in the matrix A , $\sum_{j=1}^n x_j \leq n$ and $b(m) = n$ to obtain a bounded problem.

4.2. Comparison

To measure the performance of our new optimal pivot rule, we compare the optimal pivot rule with Dantzig’s original pivot rule, written in a MATLAB environment programming. The optimal pivot rule is also compare to the simplex method included in MATLAB *optimset toolbox*. The performance measures used for comparison are the number of iterations (pivots) and the CPU time. Note that DPR is simplex algorithm with Dantzig’s pivot rule and OPR is simplex algorithm with optimal pivot rule, SML is simplex in MATLAB.

Table 9 shows the comparison between the average number of iterations and the CPU time from solving LP by the simplex algorithm with DPR, OPR and SML: the average number of iterations and the CPU time from OPR pivot rule is less than the one from DPI and SML. DPR pivot rule achieves less number of iterations when the number of constraints and variable in the problem is higher. Due to limitation of the simplex software in MATLAB platform, SML could not solve the problems with $n \geq 500$ and $m \geq 450$, with exit message*. But SML and DPR solved LP’s using almost the same number of iterations, but with a higher CPU time for SML.

*MATLAB message: “Exiting: Maximum number of iterations exceeded; increase options. MaxIter”: MATLAB could not solve the problem asking to increase the maximum number of iterations permitted.

5. Summary of Results and Conclusions

We proposed a new pivot rule called the optimal pivot rule. The idea of this rule is to

Table 9. The average number of iterations and the average CPU.

No.	Problem size		Optimal pivot		Dantzig's pivot			Simplex in MATLAB	
	<i>m</i>	<i>n</i>	Number of iterations	CPU	Number of iterations	CPU	Number of iterations	CPU	
1.	30	60	28	0.031	174	0.001	169	0.11	
2.	40	80	41	0.015	275	0.032	274	0.22	
3.	50	100	80	0.068	573	0.047	570	0.48	
4.	100	150	165	0.15	4468	0.82	4412	8.73	
5.	110	175	259	0.32	7150	1.52	7153	16.33	
6.	120	190	270	0.37	7399	1.84	7368	17.73	
7.	130	200	268	0.40	7500	2.23	7497	26.16	
8.	140	250	368	0.68	9582	3.63	9580	29.95	
9.	170	300	557	1.46	17,597	11.09	17,596	89.10	
10.	200	350	621	2.03	23,541	19.58	23,546	214.5	
11.	250	400	1084	4.78	32,408	37.28	32,407	482.4	
12.	300	450	1263	7.24	60,514	94.20	*	*	
13.	350	500	1576	12.9	86,002	267.51	*	*	
14.	400	600	1846	23.16	113,162	666.02	*	*	

compute an optimal improvement in the objective value per unit step of the basis-entering variable. From our experiments, the proposed pivot rule is faster and reduces the number of such iteration over the Dantzig's pivot rule the simplex algorithm. Tableau 9 offers a summary of the average number of iterations of each method. We conclude that the simplex algorithm using the optimal change pivot rule is very fast for solving linear programming problems when the size of the problem is large.

Using simple computational complexity facts, we proved that the new optimal pivot rule in the simplex algorithm is efficient. Moreover, we show that the optimal pivot rule solves the problem of cycling in the simplex algorithm.

6. Recommendations

In a future research, we will implement the optimal pivot rule to solve mathematical optimization problems whose algorithms are derived from simplex pivots like quadratic programming problem. The conjecture 1 stated in this article needs to be proven.

To prevent the warning message "Exiting: Maximum number of iterations exceeded, increase options. MaxIter" from the simplex in MATLAB platform, MATLAB developers should include our optimal pivot rule in the simplex method in that software, so that MATLAB will then be able to solve larger size LPs.

References

- [1] Dantzig, G.B. (1963) Linear Programming and Extensions. Princeton University Press, Princeton.

- [2] Klee, V. and Minty, G. (1972) How Good Is the Simplex Algorithm? In *Inequalities*. Academic Press, New York.
- [3] Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. (1990) *Linear Programming and Network Flows*. 2nd Edition, John Wiley, New York.
- [4] Forrest, J.J. and Goldfarb, D. (1992) Steepest-Edge Simplex Algorithm for Linear Programming. *Mathematical Programming*, **57**, 341-374.
<http://dx.doi.org/10.1007/BF01581089>
- [5] Harris, P.M.J. (1973) Pivot Selection Methods of the Devexlp Code. *Mathematical Programming*, **5**, 1-28. <http://dx.doi.org/10.1007/BF01580108>
- [6] Pan, P.-Q. (2008) A Largest-Distance Pivot Rule for the Simplex Algorithm. *European Journal of Operational Research*, **187**, 393-402. <http://dx.doi.org/10.1016/j.ejor.2007.03.026>
- [7] Terlaky, T. and Zhang, S. (1993) Pivot Rules for Linear Programming: A Survey on Recent Theoretical Developments. *Annals of Operations Research*, **46**, 203-233.
<http://dx.doi.org/10.1007/BF02096264>
- [8] Bland, R.G. (1977) New Finite Pivoting Rules for the Simplex Method. *Mathematics of Operations Research*, **2**, 103-107. <http://dx.doi.org/10.1287/moor.2.2.103>
- [9] Chankong, K., Intiyot, B. and Sinapiromsaran, K. (2014) Absolute Change Pivot Rule for the Simplex Algorithm. *Proceedings of the International MultiConference of and Computer Scientists*, Hong Kong, 12-14 March 2014, 1209-1213.



Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

