

# Absolute Change Pivot Rule for the Simplex Algorithm

Kittiphong Chankong, Boonyarit Intiyot, and Krung Sinapiromsaran

**Abstract**—The simplex algorithm is a widely used method for solving a linear programming problem (LP) which is first presented by George B. Dantzig. One of the important steps of the simplex algorithm is applying an appropriate pivot rule, the rule to select the entering variable. An effective pivot rule can lead to the optimal solution of LP with the small number of iterations. In a minimization problem, Dantzig's pivot rule selects an entering variable corresponding to the most negative reduced cost. The concept is to have the maximum improvement in the objective value per unit step of the entering variable. However, in some problems, Dantzig's rule may visit a large number of extreme points before reaching the optimal solution. In this paper, we propose a pivot rule that could reduce the number of such iterations over the Dantzig's pivot rule. The idea is to have the maximum improvement in the objective value function by trying to block a leaving variable that makes a little change in the objective function value as much as possible. Then we test and compare the efficacy of this rule with Dantzig's original rule.

**Index Terms**—linear programming; simplex algorithm; pivot rule; absolute change pivot rule.

## I. INTRODUCTION

LINEAR programming (LP) is widely used for modeling and solving optimization problems in many industries. An LP model includes an objective function subject to a finite number of linear equality and inequality constraints. To solve an LP problem, we need to consider the computational complexity that depends on the number of constraints and variables. The popular algorithm for solving LP problems is the well-known simplex algorithm which is presented by George B. Dantzig [1], in 1963.

One of the important steps of the simplex algorithm is a pivot rule, the rule that is used for selecting the entering variable. An effective rule can lead to the solution of LP with small number of iterations. Dantzig's original rule is the standard pivot rule but this rule is efficient only for LP with small sizes. Moreover Dantzig's rule can not prevent cycling in linear programming [2] and takes a lot of iterations in some case. Klee and Minty [3] exhibited the worst case running time of simplex algorithm using Dantzig's pivot rule. To avoid this weakness, there are many research studies trying to improve simplex algorithm, via the pivot rule by reducing the number of iterations and the solution time. In 1977, Forrest and Goldfarb [4] presented a way to reduce the number of iterations which was called "Steepest-edge rule".

Manuscript received January 7, 2014; revised February 5, 2014. This work was supported in part by the Applied Mathematics and Computational Science Program, Faculty of Science and Graduate School, Chulalongkorn University.

K. Chankong, B. Intiyot, K. Sinapiromsaran are with the Applied Mathematics and Computational Science Program, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, Thailand e-mail: Chankong.kc@gmail.com, Boonyarit.i@chula.ac.th, Krung.s@chula.ac.th.

Later, other rules followed such as Devex rule by Harris [5] and the largest-distance pivot rule by Pan [6].

In this paper, we propose the new pivot rule called absolute change pivot rule. The idea is trying to block a leaving variable that makes a little change in the objective function value as much as we can. If we can prevent such variables to leave the basis, it could make the objective function value improved further than using a regular Dantzig's pivot rule and therefore lead to fewer number of iterations. We report the computational results by testing and comparing the number of iterations from this new rule to Dantzig's original pivot rule.

This paper is divided in to five sections. Section 1 gives a brief introduction; Section 2 describes the preliminaries of linear programming, simplex algorithm and pivot rule. Section 3 explain the main idea of our pivot rule and show Klee and Minty problem [3] for an example. Section 4 deals with the computational results by testing and comparing the number of iterations from this new rule to another rule and conclusion has been drawn at the end.

## II. PRELIMINARIES

### A. The simplex method

Consider the linear programming (LP) problem in the standard form, where  $A \in \mathbb{R}^{m \times n}$  ( $m < n$ ),  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$  and  $\text{rank}(A) = m$ .

$$\begin{aligned} \text{Minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned} \quad (1)$$

After possibly rearranging the column of  $A$ , let  $A = [B \ N]$  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 nonbasic matrix. We denote basic and nonbasic index set by  $I_B$  and  $I_N$  respectively. Let  $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$  be the solution of the equation  $Ax = b$ , where  $x_B = B^{-1}b$  and  $x_N = 0$  is called a basic solution of the system. 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

$$z_0 = c_B^T B^{-1}b. \quad (2)$$

Now let  $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$  denote the set of basic and nonbasic variables for the given basis. Then feasibility requires that  $x_B \geq 0$  and  $x_N \geq 0$ . We denote the  $j^{\text{th}}$  column of  $A$  by  $A_j$ . Then we can rewrite the system  $Ax = b$  as:

$$Bx_B + Nx_N = b. \quad (3)$$

Then

$$x_B = \bar{b} - \sum_{j \in I_N} (y_j x_j) \quad (4)$$

where  $\bar{b} = B^{-1}b$  and  $y_j = B^{-1}A_j$ . Let  $z$  be the objective function value, we get

$$z = z_0 - \sum_{j \in I_N} (z_j - c_j)x_j \quad (5)$$

where  $z_j = c_B^T B^{-1}A_j$  for each nonbasic variable. The negative reduced cost is obtained by  $z_j - c_j$ . The key result exhibits that the optimal solution is achieved if the index set

$$J = \{j \mid z_j - c_j > 0, j \in I_N\} \quad (6)$$

is empty.

### B. The simplex algorithm

Consider the algorithm of the simplex method with Dantzig's pivot rule for solving linear programming problem of the system (1).

**Initialization Step :** Choose a starting basic feasible solution with the basis  $B$  and the associated nonbasic  $N$ .

**Main Step :**

- Step1. Determine the entering variable from the nonbasic variables: By Dantzig's rule choose  $x_k$  by  $z_k - c_k = \max\{z_j - c_j \mid j \in I_N\}$ .
- Step2. If  $z_k - c_k \leq 0$ , then  $\begin{bmatrix} x_B \\ x_N \end{bmatrix}$  is an optimal solution. Stop the algorithm.
- Step3. Determine the leaving variable from the basic variables by the minimum ratio test.
- Step4. Perform the pivot operation using the entering and the leaving variable, and go to Step 1.

### C. The simplex method in tableau format

Suppose that we have a starting basic feasible solution  $x$  with basis  $B$ . The linear programming problem can be represented as follows:

$$\begin{aligned} \text{Minimize } & z \\ \text{subject to } & z - c_B^T x_B - c_N^T x_N = 0 \end{aligned} \quad (7)$$

$$\begin{aligned} Bx_B + Nx_N &= b \\ x_B, x_N &\geq 0. \end{aligned} \quad (8)$$

From equation (8) we have

$$x_B + B^{-1}Nx_N = B^{-1}b. \quad (9)$$

Multiplying (9) by  $c_B^T$  and adding to equation (7), we get

$$z + (c_B^T B^{-1}N - c_N^T)x_N = c_B^T B^{-1}b. \quad (10)$$

Set,  $x_N = 0$ , and from equation (9) and (10) we get  $x_B = B^{-1}b$  and  $z = c_B^T B^{-1}b$ . Also, from (9) and (10) we can conveniently represent the current basic feasible solution with basis  $B$  in the following tableau.

The tableau format, reports the value of the objective function  $z = c_B^T B^{-1}b$ , the basic variables  $x_B = B^{-1}b$  and the cost row  $c_B^T B^{-1}N - c_N^T$ , which consists of the  $z_j - c_j$  values for the nonbasic variables.  $\forall j, z_j - c_j \leq 0$ , then LP is at optimal solution. If  $x_k$  increases, then the vector  $y_k = B^{-1}A_k$ , which is stored in the tableau in rows 1

	$z$	$x_B$	$x_N$	RHS	
$z$	1	0	$c_B^T B^{-1}N - c_N^T$	$c_B^T B^{-1}b$	Row 0
$x_B$	0	$I$	$B^{-1}N$	$B^{-1}b$	Rows 1 through $m$

through  $m$  under the variable  $x_k$ , will determine how much  $x_k$  can be increased. If  $y_k \leq 0$ , then  $x_k$  can be increased indefinitely without being blocked, and the optimal objective value is unbounded. Conversely, if  $y_k \not\leq 0$ , that is, if  $y_k$  has at least one positive component, then the increase in  $x_k$  will be blocked by one of the current basic variables, which drops to zero. The minimum ratio test determines the blocking variable.

### D. Pivot rule

In terms of the geometric motivation of the simplex method, a pivot operation is equivalent to moving from a basic feasible solution to an adjacent basic feasible solution. If we want to pivot at the non-negative and nonzero pivot element, steps of pivot operations are as follows. First, select a nonbasis variable in the column  $k$  as a pivot column, the variable corresponding to the pivot column enters the set of basic variable is called **entering variable**. Second, divide all of element in the row  $r$  that associate with the selected nonbasic column by its reciprocal to change this element to 1. The entries in the column that containing the number 1 have to change to 0 by row operation method, then the column  $k$  becomes to the one of column of identity matrix. So the entering variable is now a basic variable which is replacing the variable in row  $r$ , the variable being replacing leaves the set of basic variable is called the **leaving variable**.

The method that use to select an entering variable in the simplex algorithm is called as **pivot rule**. The best pivot rule would move along the path with the shortest number of visited nodes from the starting point to an optimal solution. It is well known that Dantzig's pivot rule [1] is the first rule to select the entering variable. If  $J \neq \emptyset$ , the entering variable  $x_k$  based on Dantzig's pivot rule is selected by the most negative reduced cost as follows

$$k = \operatorname{argmax}\{z_j - c_j \mid j \in J\}. \quad (11)$$

After the pivot column has been selected, then we consider only the positive entries in the pivot column  $k$ . If all entries in pivot column  $k$  is not positive then the problem is unbounded. If the pivot column is  $k$ , then the pivot row  $r$  is chosen so that

$$r = \operatorname{argmin} \left\{ \frac{\bar{b}_j}{y_{jk}} \mid y_{jk} > 0, j \in \{1, \dots, m\} \right\}$$

This is called the **minimum ratio test**.

## III. ABSOLUTE CHANGE PIVOT RULE

To select the entering variable, Dantzig's pivot rule consider only the most negative reduced cost. It is possible that we can select an entering variable that can improve the objective function value as much as possible. Hence we propose a pivot rule to improve the objective value by trying to avoid the leaving variable that causes small change in the objective function. We call it *absolute change pivot rule*.

A. The concept of absolute change pivot rule

First, this rule looks for the row with the minimum right-hand-side. The motivation behind this is that, given an entering variable, the basic variable associated with this row will have a tendency to become zero first and, as the result, tends to become the leaving variable. By avoiding having this variable leaving the basis, we can increase the value of the entering variable further. To prevent such variable from leaving the basis, we look for an entering variable that has zero or negative value in that row so that the minimum ratio is not applicable for that row. If there is more than one candidate for such entering variable, we look for the row with the next minimum right-hand-side and repeat the process until we have only one candidate or until we cannot find a row with zero or negative value anymore. If we still end up with more than one entering candidate, we select the one with the most negative reduced cost. In summary, this rule heuristically selects the entering variable that can improve the objective value more.

Simplex algorithm with absolute change pivot rule

- Step1. Check  $z_j - c_j \leq 0$  for all  $j \in J$ , then  $\begin{bmatrix} x_B \\ x_N \end{bmatrix}$  is an optimal solution. Stop the algorithm.
- Step2. Determine the entering variable by using absolute change pivot rule.
  - i. Set  $CI_B = I_B$ . Let  $J = \{j \mid z_j - c_j > 0, j \in I_N\}$
  - ii. Select index  $r$  such that  $r = \text{argmin}\{\bar{b}_i \mid \text{for } i \in I_B\}$
  - iii. Let  $\hat{J} = \{j \in J \mid A_{rj} \leq 0\}$ . If  $\hat{J} \neq \emptyset$ , let  $J = \hat{J}$ .
    - If  $|\hat{J}| = 1$ , go to (iv). Else, remove  $i$  from  $CI_B$ . Go to (ii).
  - iv. Else, choose  $x_k$  by  $z_k - c_k = \max\{z_j - c_j \mid j \in J\}$
- Step3. Determine the leaving variable from the basic variables by the minimum ratio test.
- Step4. Perform the pivot operation using the entering and the leaving variable, and go to Step 1.

B. Illustration of the method

The proposed pivot rule is shown with two examples, one from the Klee and Minty problem and another one is randomly generated linear programming problem.

Example 1. Klee and Minty problem

In 1972, Klee and Minty showed a collection of LP problems that the worst-case complexity of the simplex method with Dantzig’s pivot rule is exponential time. The collection of LP is given by

$$\begin{aligned} \text{Minimize} \quad & - \sum_{j=1}^n 10^{n-j} x_j, \\ \text{subject to} \quad & 2 \sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1}, i = 1, \dots, n, \\ & x \geq 0, i = 1, \dots, n. \end{aligned} \tag{12}$$

The simplex method with Dantzig’s pivot rule requires  $2^n - 1$  iterations to solve the Klee and Minty problem.

The following examples are presented to show the efficiency of the proposed pivot rule. For simplicity, we use Klee and Minty problem with  $n = 3$ .

Consider the following problem:

$$\begin{aligned} \text{Minimize} \quad & -100x_1 - 10x_2 - x_3, \\ \text{subject to} \quad & x_1 \leq 1, \\ & 20x_1 + x_2 \leq 100, \\ & 200x_1 + 20x_2 + x_3 \leq 10000, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

The problem above can be written in tableau format, where  $x_4, x_5, x_6$  are the slack variables, as follows:

	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	RHS
$z$	1	100	10	1	0	0	0	0
$x_4$	0	1	0	0	1	0	0	1
$x_5$	0	20	1	0	0	1	0	100
$x_6$	0	200	20	1	0	0	1	10000

From the tableau we can see that the right-hand-side entries are already sorted from smallest to largest values. Consider the first row ( $x_4$ ), since the element in the second and third column are zero then  $x_2$  and  $x_3$  can be the entering variables.  $x_1$  is not considered as a candidate because the element in the first column is non negative. Since we still have two candidates, the second row has to be considered. The second row has zero value at the third column so  $x_3$  is a candidate while  $x_2$  is no longer a candidate since its entry is positive. As the result, the entering variable is  $x_3$ . From the minimum ratio test we get  $x_6$  as the leaving variable. After pivot operation, we obtain the optimal solution which is shown in the tableau as follows.

	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	RHS
$z$	1	-100	-10	0	0	0	-1	-10000
$x_4$	0	1	0	0	1	0	0	1
$x_5$	0	20	1	0	0	1	0	100
$x_3$	0	200	20	1	0	0	1	10000

For this example the optimal solution is  $x_1 = 0, x_2 = 0$  and  $x_3 = 10000$ . The optimal value is 10000 with the number of iteration is 1.

Example 2. Consider the following randomly generated linear programming problem :

$$\begin{aligned} \text{Minimize} \quad & -50x_1 - 2x_2 - 46x_3 - 40x_4 - 15x_5, \\ \text{subject to} \quad & 15x_1 - 3x_2 + 22x_3 + 3x_4 - 4x_5 \leq 1467, \\ & 17x_1 + 11x_2 + 23x_3 + 19x_4 - 28x_5 \leq 1733, \\ & 10x_1 - 18x_2 + 21x_3 + 28x_4 + 6x_5 \leq 1758, \\ & -49x_1 + 6x_2 + 36x_3 + 34x_4 - 2x_5 \leq 606, \\ & -33x_1 + 25x_2 + 48x_3 - 14x_4 + 12x_5 \leq 1365, \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

Let  $x_6, x_7, x_8, x_9, x_{10}$  be the slack variables associated with all of constraints in example 2, respectively. The initial simplex tableau for example 2 is:

	z	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	RHS
z	1	50	2	46	40	15	0	0	0	0	0	0
$x_6$	0	15	-3	22	3	-4	1	0	0	0	0	1467
$x_7$	0	17	11	23	19	-28	0	1	0	0	0	1733
$x_8$	0	10	-18	21	28	6	0	0	1	0	0	1758
$x_9$	0	-49	6	36	34	-2	0	0	0	1	0	606
$x_{10}$	0	-33	25	48	-14	12	0	0	0	0	1	1365

From the tableau above if we follow the algorithm, after the first iteration we get  $x_1$  as the entering variable and  $x_6$  as the leaving variable. After pivoting, the simplex tableau becomes:

	z	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	RHS
z	1	0	12	-27.333	30	28	-3.333	0	0	0	0	-4890
$x_1$	0	1	-0.2	1.467	0.2	-0.3	0.067	0	0	0	0	97.8
$x_7$	0	0	14.4	-1.933	15.6	-23	-1.133	1	0	0	0	70.4
$x_8$	0	0	-16	6.333	26	8.7	-0.667	0	1	0	0	780
$x_9$	0	0	-3.8	107.867	43.8	-15	3.267	0	0	1	0	5392.2
$x_{10}$	0	0	18.4	96.4	-7.4	3.2	2.2	0	0	0	1	4592.4

After the second iteration,  $x_5$  is the entering variable and  $x_8$  is the leaving variable. The result simplex tableau becomes:

	z	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	RHS
z	1	0	64.3	-48.039	-55	0	-1.154	0	-3.269	0	0	-7440
$x_1$	0	1	-0.7	1.661	1	0	0.046	0	0.031	0	0	121.8
$x_7$	0	0	-29	15.215	86	0	-2.938	1	2.708	0	0	2182.4
$x_5$	0	0	-1.8	0.731	3	1	-0.077	0	0.115	0	0	90
$x_9$	0	0	-32	118.877	89	0	2.108	0	1.738	1	0	6754.2
$x_{10}$	0	0	24.3	94.062	-17	0	2.446	0	-0.369	0	1	4304.4

In the last iteration of this example,  $x_2$  is the entering variable and  $x_{10}$  is the leaving variable. The last simplex tableau is:

	z	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	RHS
z	1	0	0	-296.9	-10.0	0	-7.6	0	-2.3	0	-2.6	-18827.6
$x_1$	0	1	0	4.34	0.52	0	0.12	0	0.02	0	0.03	121.8
$x_7$	0	0	0	127.14	65.77	0	-0.03	1	2.27	0	1.19	2182.4
$x_5$	0	0	0	7.88	1.71	1	0.11	0	0.09	0	0.08	90
$x_9$	0	0	0	241.22	66.89	0	5.29	0	1.26	1	1.30	6754.2
$x_2$	0	0	1	3.87	-0.69	0	0.10	0	-0.02	0	0.04	4304.4

This simplex tableau is the optimal tableau with optimal solution  $x_1 = 121.8$ ,  $x_2 = 4304.4$ ,  $x_3 = 0$ ,  $x_4 = 0$  and  $x_5 = 90$ . The optimal value is 18827.589 with the number of iterations is 3 while the simplex method with Dantzig's pivot rule uses 5 iterations to achieve the optimal solution.

Note that if the tableau does not contain zero entries or entries with negative value in the coefficient matrix, this rule is simply the Dantzig's. Therefore, to take advantage of this pivot rule, we should consider only the problems with some zero or negative entries that correspond to entering columns in the coefficient matrix.

TABLE I  
THE AVERAGE NUMBER OF ITERATIONS.

No.	Problem size		Average no. of iterations	
	m	n	DP	ACP
1	10	10	6.83	7.21
2	15	15	13.43	13.93
3	20	20	20.45	21.78
4	25	25	26.98	28.19
5	30	30	33.63	36.08
6	35	35	44.51	44.58
7	40	40	54.50	58.36
8	45	45	73.42	68.97
9	50	50	81.57	82.08
10	55	55	91.85	89.85
11	60	60	113.22	104.38
12	70	70	151.34	127.58
13	80	80	188.38	164.93
14	90	90	247.45	199.45
15	100	100	281.50	224.50
16	120	120	418.72	308.35
17	140	140	531.75	365.30
18	160	160	688.50	445.53
19	180	180	863.95	533.43
20	200	200	1041.03	634.63
21	250	250	1656.65	877.98
22	300	300	2367.47	1164.78
23	400	400	4219.51	1730.47
Average			574.64	318.80

#### IV. COMPUTATIONAL EXPERIMENTS

In this section, we present the computational results of simplex algorithm with absolute change pivot rule. Absolute change pivot rule was tested by solving randomly generated linear programming problems of various sizes. We compare the number of iterations of this pivot rule with Dantzig's pivot rule.

The programming language used was Python. All codes were run under an Oracle VM VirtualBox version 4.3.4r91027 by software Sage [7] version 5.7 with base memory 512 MB. The computer system processor is Intel(R) Core(TM)i5 CPU M 460 @2.53GHz, 4.00 GB of memory, and 64-bit Window 8.1 Operating System.

##### A. Problem generation

All randomly generated linear programming problems are minimization problems and are generated according to the following specifications: The cost vector  $c$  is generated with  $c_i \in [-10, 10]$ . The matrix  $A$  is generated with  $A_{ij} \in [-10, 10]$ . To guarantee a feasible problem, we generate the right-hand-side  $b$  by generating a feasible solution  $x$  with  $x_i \in [0, 10]$  and then  $b$  is calculated by  $b = Ax$ . All constraints are in the form  $\leq$ .

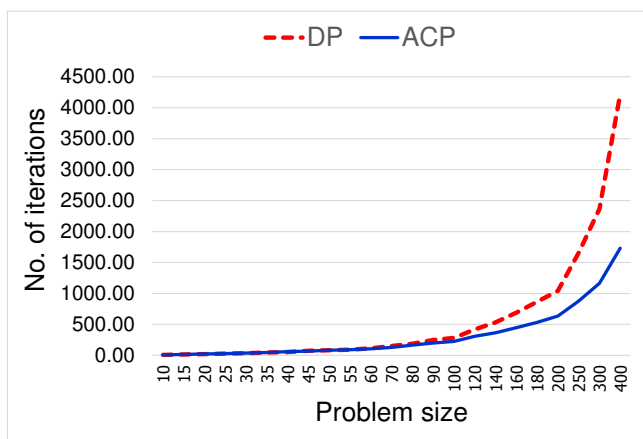


Fig. 1. Comparison of the simplex algorithm with DP and ACP.

The sides of problems are varied as shown in table I. For each size, we generate 50 problems and find the mean results for each method.

### B. Comparison

To measure the performance of the absolute change pivot rule, we compare this rule with Dantzig's original pivot rule. The performance measures used for comparison is the number of iterations (pivot). Note that DP is simplex algorithm with Dantzig's pivot rule and ACP is simplex algorithm with absolute change pivot rule.

Table I shows the comparison between the average number of iterations from solving LP by the simplex algorithm with DP and ACP. This table also shows that the average number of iterations from ACP pivot rule is less than the one from DP. ACP pivot rule achieves less number of iterations when the number of constraints and variable in the problem is higher. Moreover, in figure 1 we found that the number of iteration between ACP and DP is not significantly different in small problem while in large-scale problems ACP have a better performance.

## V. SUMMARY OF RESULTS

We proposed a pivot rule called the absolute change pivot rule. The idea of this rule is to have the maximum improvement in the objective value per unit step of the entering variable. We believe that the proposed algorithm could reduce the number of such iteration over the Dantzig's pivot rule. Table I offers a summary of the average number of iteration of each method. We conclude that the simplex algorithm using the absolute change pivot rule is very fast for solving linear programming problems. Moreover, absolute change pivot rule performs very well on a Klee and Minty problems.

For future works, we may create a new pivot algorithm that have more efficient than absolute change pivot rule and can apply to any linear programming problems.

### ACKNOWLEDGMENT

The authors would like to thank the Applied Mathematics and Computational Science Program, Department of Math-

ematics and Computer Science, Faculty of Science, Chulalongkorn University for providing facilities and resources for this research.

### REFERENCES

- [1] G. B. Dantzig, *Linear Programming and Extensions*. Princeton: Princeton University Press, 1963.
- [2] S. H. Bazara M., Jarvis J., *Linear programming and network flows*, 2nd ed. New York: John Wiley, 1990.
- [3] V. Klee and G. Minty, *How good is the simplex algorithm? in inequalities*. New York: Academic Press, 1972.
- [4] G. D. Forrest J., "A practical steepest-edge simplex algorithm for linear programming," *Mathematical Programming*, vol. 57, pp. 341–374, 1992.
- [5] H. P.M.J., "Pivot selection methods of the devex lp code," *Mathematical Programming*, vol. 5, pp. 1–28, 1973.
- [6] P.-Q. Pan, "A largest-distance pivot rule for the simplex algorithm," *European Journal of Operational research*, vol. 187, pp. 393–402, 2008.
- [7] W. Stein and D. Joyner, "Sage: System for algebra and geometry experimentation," *ACM SIGSAM Bulletin*, vol. 39, no. 2, pp. 61–64, 2005.