

# Complexity of Pivot Algorithms

Qi Wang, Sean Kelley

December, 2020

## Abstract

Abstract...

## 1 Introduction

Simplex method was invented by Dantzig in 1947 [4] to solve the Linear Optimization (LP) problem. It was tableau pivoting based method and pivoting under certain rules. The rules can be flexible so many simplex variants have been developed afterwards. Another framework of pivot algorithm is criss-cross method, which was proposed by Zions at 1969 [5] and then [8], [2] present finite criss-cross version independently around 1980s. Some variant pivot methods appeared by proposing different pivots rules. We will describe them in the following section. **Mention Short simplex paths**

Although simplex method was generally efficient in practical, theoretically, Klee and Minty [3] in 1970s showed the worst case that, a specific type of LO problem, a variant of simplex method visited all vertices (exponential steps) until solve the problem to be optimal. The feasible region of such problems was call Klee-Minty cube whose corners have been "squashed". We will demonstrate some examples in the following section. For now, it is still an unaccomplished problem that to design a pivot algorithm and prove that the number of pivot steps is bounded by polynomial of number of variables and constraints. The paper is structured as follows: in Section 2 we define the LO problem, list the notation and (**might present some algorithms**); in section 3 we discuss the complexity of pivot algorithms and give a brief conclusion in the final section.

## 2 The pivot algorithm for LO

In this paper, we consider the primal Linear optimization problem of the standard form

$$\begin{aligned}
 \min \quad & c^T x \\
 \text{s.t.} \quad & Ax = b \\
 & x \geq 0 \\
 & x \in \mathbb{R}^n, \ c \in \mathbb{R}^n, \ A \in \mathbb{R}^{m \times n}, \ b \in \mathbb{R}^m
 \end{aligned} \tag{1}$$

And the dual problem is

$$\begin{aligned}
 \min \quad & b^T y \\
 \text{s.t.} \quad & A^T y + s = c \\
 & s \geq 0 \\
 & y \in \mathbb{R}^m, \ s \in \mathbb{R}^n
 \end{aligned} \tag{2}$$

Table 1: Notation

$n$	number of variables
$m$	number of constraints (assume $m \leq n$ )
$B$	set of index of basis solution $x_B$ , $ B  = m$
$N$	set of index of nonbasis solution $x_N$ , $ N  = n - m$

## 3 Complexity

### 3.1 Historical Review

As we will go on to see in this article, much of our conversation will revolve around advancements in pivoting rules that led to performance increases for pivot algorithms. Before getting there, however, we would like to begin by discussing a few key improvements to other aspects of pivot algorithms that contributed to research focus in this space. We will start by going all the way back to the beginning.

#### 3.1.1 Dantzig's Simplex Algorithm

As mentioned in our opening, the major advancement that opened the space of pivot algorithms was the development of the Primal Simplex Algorithm

(referred to hereon as Simplex) by Dantzig, and his work will serve as our baseline for the complexity of pivot algorithms as we go on to compare them moving forward. In order to understand the complexity of the algorithm Dantzig proposed, let's take a moment to refresh ourselves on its operations. Although mentioned in greater detail in [4], the important details to recall are as follows. We begin by setting up the following system of equations where the equality exists between the first and second columns:

0	$-c_B^T x_B$	$-c_N^T x_N$	$x_0$
$b$	$A_B x_B$	$A_N x_N$	0

Then with a couple of algebraic operations, we get the following tableau:

$-c_B^T A_B^{-1} b$	0	$(c_B^T A_B^{-1} A_N - c_N^T) x_N$	$x_0$
$x_B$	$I$	$A_B^{-1} A_N x_N$	0

Simplex tells us to then pivot through bases until we find one that is either optimal or inconsistent. To make a pivot, Dantzig said to do the following:

- Find the most positive element in the top row. Denote its index as  $q$  ( $n - m$  operations)
- Find the smallest  $x_i / \tau_{iq} : \tau_{iq} > 0$ , where  $\tau_{iq}$  is the  $i$ th element of the  $q$ th column in our tableau. If there is a tie, arbitrarily pick one of the tying elements ( $3m$  operations)
- Swap  $i$  and  $q$  in their respective index sets.
- Multiply  $A_B^{-1}$  by  $A_N$  ( $m^2 * (n - m)$  operations)
- Add  $c_B^T A_B^{-1} A_N$  to  $-c_N^T$  ( $m^2 + m(n - m) + (n - m)$  operations)

We see then this takes us to a total of  $m^2(n - m) + m^2 + m(n - m) + m + 2n = O(m^2 n)$  operations for the pivot.

Now as we showed in class [9], we know that so long as there is no degeneracy in the problem, simplex cannot cycle, so the number of pivots is bounded by the number of possible bases, which was  $\binom{n}{m}$ . Therefore, we know Dantzig's version of simplex to have complexity  $O(\binom{n}{m} m^2 n)$  for nondegenerate problems.

### 3.1.2 Bland's Rule

Now for Dantzig's simplex, we assumed that a problem would have no degeneracy, which is a rather high standard for using the algorithm in practice. To further complicate the matter, degeneracy leads to cycling in that version of simplex, which means the algorithm could run infinitely and never converge. To get around that issue then, many pivot rules have been proposed which help simplex avoid cycles. Arguably none of these such pivoting rules are more famous than Bland's Rule, which updates the above steps of simplex in the following way:

- instead of picking the most positive top row element to enter the basis, pick the first positive top row element.
- instead of arbitrarily selecting a basis row matching the minimal ratio, select the minimal ratio basis row with least index.

As shown in [1], these updates to Dantzig's simplex method ensure that we never cycle and make all linear programs converge and finite. Thus all linear programs now have complexity  $O(\binom{n}{m}m^2n)$ .

### 3.1.3 Revised Simplex Method

With simplex having a finite worst case across all problems, it is worthwhile to look at ways now to lower that upper bound. One of the first places the optimization community had success in doing this was with speeding up the process of making a pivot, which has come to be known as the Revised Simplex Method. Again, Revised Simplex differs from Dantzig's Simplex in two key ways [11]:

- instead of calculating all of the objective row for each pivot, just calculate enough columns until we get a positive entry, pivoting on that (up to  $n - m$  operations)
- instead of multiplying  $A_B^{-1}$  by all of  $A_N$ , just multiply by the column corresponding to our first found positive entry ( $m^2$  operations).

It is worth highlighting at this point that [11] shows Revised Simplex yields the same solution as Dantzig's Simplex but saves us from making many unnecessary calculations. This change brings our pivot to  $2m^2 + m(n - m) + m + 2n = O(mn)$  operations (the same as in Dantzig's Simplex but swap  $m^2(n - m)$  for  $m^2$ ).

### 3.2 Worst case

Given  $m$  constraints and  $n$  variables, there are at most  $\binom{n}{m}$  possible basis, which is the biggest upper bound for the number of iterations for pivot algorithms. When  $m = \frac{n}{2}$ ,  $\binom{n}{m}$  get its maximum  $\binom{n}{\frac{n}{2}} \approx \sqrt{\frac{2}{\pi n}} 2^n$ . Klee and Minty defined a type of LO problems which some pivot algorithms may visit all vertices before finally finding the optimal solution. Detailed introduction of Klee Minty cube can be referred to [10]. Here we demonstrate some examples. Klee Minty cube is defined as following LO problem 3. The coefficients might be different in different articles.

$$\begin{aligned}
 & \max \sum_{j=1}^n 2^{n-j} x_j \\
 & \text{s.t. } 2 \sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq 5^i, \quad i = 1, \dots, n \\
 & \quad x_j \geq 0, \quad j = 1, \dots, n
 \end{aligned} \tag{3}$$

When  $n = 2$  and  $n = 3$  the 2-d and 3-d Klee Minty LO problems are

$$\begin{array}{ll}
 \max & 2x_1 + x_2 \\
 \text{s.t.} & x_1 \leq 5 \\
 & 4x_1 + x_2 \leq 25 \\
 & x_1 \geq 0, \quad x_2 \geq 0.
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & 4x_1 + 2x_2 + x_3 \\
 \text{s.t.} & x_1 \leq 5 \\
 & 4x_1 + x_2 \leq 25 \\
 & 8x_1 + 4x_2 + x_3 \leq 125 \\
 & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.
 \end{array}$$

The feasible region of 2-d and 3-d Klee Minty problems are shown in Figure 1. They look like "squashed" square and cube, which are very ill-conditioned. We apply primal simplex method (a) least-index version and (b) Dantzig (the nonbasis with biggest reduced cost enters) pivot rules to demonstrate the number of pivot steps. The tableau result of least-index pivot rule is shown in Table 2. The results of Dantzig pivot rule and least-index rule just coincide in this example because in first iteration, the nonbasis with the most negative reduced cost is  $x_1$  which is the entering nonbasis by least-index rule, too.

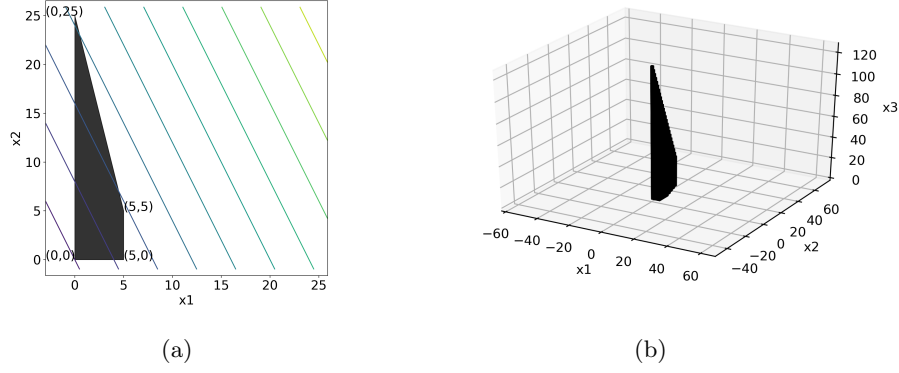


Figure 1: (a) Feasible region of 2-d Klee Minty Cube with contour lines (b) Feasible region of 3-d Klee Minty Cube

Table 2: Tableau of primal simplex method (least index version), the element with \* is the pivot element.

	rhs	$x_1$	$x_2$	$s_1$	$s_2$	
$z$	0	2	1	0	0	at vetex (0,0)
$s_1$	5	1*	0	1	0	
$s_2$	25	4	1	0	1	
<hr/>						
$z$	-10	0	1	-2	0	at vetex (5,0)
$x_1$	5	1	0	1	0	
$s_2$	5	0	1*	-4	1	
<hr/>						
$z$	-15	0	0	2	-1	at vetex (5,5)
$x_1$	5	1	0	1*	0	
$x_2$	5	0	1	-4	1	
<hr/>						
$z$	-25	-2	0	0	-1	at vetex (0,25)
$s_1$	5	1	0	1	0	
$x_2$	25	4	1	0	1	
<hr/>						

By taking a look on the constraints, the first constraint says that  $x_1$  is no bigger than 5. With this in mind, the second constraint says that  $x_2$  has an upper bound of about 25, depending on how big  $x_1$  is. Similarly, the third constraint says that  $x_3$  is roughly no bigger than 125 (again, this statement needs some adjustment depending on the sizes of  $x_1$  and  $x_2$ ). Therefore, the constraints are approximately just a set of upper bounds, which means that

the feasible region is virtually a stretched  $n$ -dimensional hypercube, which has  $2^n$  vertices. And Klee and Minty [7] showed that Dantzig pivot rule with primal simplex algorithm will conduct  $2^n - 1$  pivot steps.

### 3.3 Kitahara and Mizuno analysis

In 2011, Kitahara and Mizuno [6] proved the complexity can be bounded by a polynomial of  $m, n$  and a special ratio. They utilized the analysis of Ye (2010) [12] for Markov Decision Problem. We will summarize the key results of Kitahara and Mizuno [6] in this section.

We define some new notations. A BFS is said to be a basic feasible solution for primal LO problem (1). Let  $\delta$  and  $\gamma$  be the minimum and the maximum values of all the positive elements of all BFSs. That is, for any BFS  $\hat{x} \in \mathbb{R}^n$ , if  $\hat{x}_j \neq 0$ ,  $j \in \{1, \dots, n\}$  where the subscript  $j$  represent the  $j$ th entry of  $\hat{x}$ , we have

$$\delta \leq \hat{x}_j \leq \gamma. \quad (4)$$

The most important result of Kitahara and Mizuno [6] is the following theorem and corollary.

**Theorem 1.** *When we apply the simplex method with the most negative rule or the best improvement rule for LO 1 having optimal solutions, we encounter at most*

$$n \lceil m \frac{\gamma}{\delta} \log(m \frac{\gamma}{\delta}) \rceil \quad (5)$$

*different basic feasible solutions.*

**Corollary 1.** *If the primal problem is nondegenerate, the simplex method finds an optimal solution in at most  $n \lceil m \frac{\gamma}{\delta} \log(m \frac{\gamma}{\delta}) \rceil$  iterations.*

The proof can be referred to Kitahara and Mizuno [6].

Therefore the bound (5) is a polynomial of  $n, m$  and  $\frac{\gamma}{\delta}$  with assumptions that the primal LO 1 has optimal solutions and is nondegenerate. In practical, the ratio of  $\frac{\gamma}{\delta}$  is not easy to get prior to solving the problem. However, for some specific form of LO,  $\frac{\gamma}{\delta}$  can be bounded using LO coefficients.

#### 3.3.1 LO with a totally unimodular matrix $A$ and integral $b$

**Definition 1.** *A matrix  $A$  is totally unimodular if every square submatrix has determinant 0, -1 or 1. In particular, this implies that all entries are 0, -1 or 1.*

Totally unimodular matrices are very well behaved, because they always define polytopes with integer vertices, as long as the right-hand side is integer-valued. Many practical problem can be modelled using totally unimodular matrices, such as the node arc incidence matrix of a directed graph.

With totally unimodular  $A$  and integral  $b$ , all the elements of any BFS are integers, so  $\delta \geq 1$ . And for a BFS  $x = (x_B, x_N)$ ,  $x_N = 0$  and  $x_B = A_B^{-1}b \geq 0$ , since  $A_B$  is totally unimodular,  $A_B^{-1}$  is unimodular [6] and all the elements of  $A_B^{-1}$  are  $-1, 0$  or  $1$ . Thus for any  $j \in B$  we have  $x_j \leq \|b\|_1$ , which implies  $\gamma \leq \|b\|_1$ . Thus  $\frac{\gamma}{\delta} \leq \frac{\|b\|_1}{1}$ . Thus by Theorem 1, we have

**Corollary 2.** *Assume that the constraint matrix  $A$  of LO (1) is totally unimodular and the constraint vector  $b$  is integral. When we apply the simplex method with the most negative rule or the best improvement rule for LO (1), we encounter at most  $n \lceil m \|b\|_1 \log(m \|b\|_1) \rceil$  different basic feasible solutions. Moreover if the LO (1) is nondegenerate, this is the most iterations to find optimal solution.*

### 3.3.2 Markov decision problem

In [6], Kitahara and Mizuno also extended the theory for Markov decision problem (MDP), where the number of possible actions is two, is formulated as

$$\begin{aligned} \min \quad & c_1^T x_1 + c_2^T x_2 \\ \text{s.t.} \quad & (I - \theta P_1)x_1 + (I - \theta P_2)x_2 = e, \\ & x_1, x_2 \geq 0 \end{aligned} \tag{6}$$

where  $I$  is the  $m \times m$  identity matrix,  $P_1, P_2$  are  $m \times m$  Markov matrices (Markov transition matrices),  $\theta$  is a discount rate, and  $e$  is the vector of all ones. MDP has following properties.

1. MDP is nondegenerate.
2. The minimum value of all the positive elements of BFSs is greater than or equal to 1, i.e.,  $\delta \geq 1$ .
3. The maximum value of all the positive elements of BFSs is less than or equal to  $\frac{m}{1-\theta}$ , i.e.,  $\gamma \leq \frac{m}{1-\theta}$

Therefore, again by Theorem 1, Kitahara and Mizuno [6] obtained the similar result to Ye [12]



**Corollary 3.** *The simplex method for solving MDP finds an optimal solution in at most  $\lceil m \frac{m^2}{1-\theta} \log(\frac{m^2}{1-\theta}) \rceil$  iterations, where  $n = 2m$ .*

**TODO** I plan to run some numerical tests on random generated LO problem, klee-minty?, and unimodular matrix problem by solvers and simplex method with difference pivot rules, to see whether iterations given by solvers agree with the theory.

## 4 Conclusion

Conclusion...

## References

- [1] Robert Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977.
- [2] Yow-Yieh Chang. Least-index resolution of degeneracy in linear complementarity problems. Technical report, STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH, 1979.
- [3] Wikipedia Contributors. Klee–minty cube, 11 2020.
- [4] George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13:339–347, 1951.
- [5] K. Fukuda and T. Terlaky. Criss-cross methods: A fresh view on pivot algorithms. *Mathematical programming*, 79(1-3):369–395, 1997.
- [6] Tomonari Kitahara and Shinji Mizuno. A bound for the number of different basic solutions generated by the simplex method. *Mathematical Programming*, 137(1-2):579–586, 2013.
- [7] V. Klee and G.J. Minty. How good is the simplex algorithm. *Inequalities*, 3(3):159–175, 1972.
- [8] T. Terlaky. A finite crisscross method for oriented matroids. *Journal of Combinatorial Theory, Series B*, 42(3):319–327, 1987.
- [9] Tamas Terlaky. Introduction to mathematical optimization, 2015.

- [10] Robert J Vanderbei. *Linear programming: foundations and extensions*, volume 285. Springer Nature, 2020.
- [11] Harvey Wagner. A comparison of the original and revised simplex methods. *Operations Research*, 5(3):103–107, 1957.
- [12] Yinyu Ye. The simplex method is strongly polynomial for the markov decision problem with a fixed discount rate, 2010.

## 5 Outline Suggestion

what are pivot algorithms? include point that there are two parts - selecting the pivot and making the pivot

Why did pivot algorithms come about?

what kinds are there? simplex criss cross

who developed them? when?

how do they differ (include problem assumptions/set up and iteration differences)?

what were their pivot calculation (everything) and pivot choosing rules (e.g. dantzig's rule - just most negative reduced cost/element in objective row)

what were their complexity?  $O(\text{something})$  when nondegenerate in worst case degeneracy could still create infiniteness/nonconvergence

what is degeneracy?

why does degeneracy matter (how does it affect complexity)?

how did we get around degeneracy? zadeh's/cunningham's rules blands rule

What did this bring complexity to? same as before, now just covering a much wider array of problems now all were finite/convergent

How could we improve this? either improve the pivot operation or the pivot selection

How could we improve the pivot operations? Revised simplex method

What is the revised simplex method? Who worked on this and when?

What did this bring complexity to?

This became the basis for modern solvers using pivot algorithms (get confirmed by gurobi/cplex/xpress/coin) mention improvements made by pre-processing, but don't go into detail - not sure if they are problem specific or general to pivot algorithms

Why worry about improving pivot selection? pivot algorithms usually average polynomial time complexity but klee-minty showed that worst case complexity can actually be achieved (what is klee-minty)

What were some early "improvements" to pivot selection? Greatest Improvement (calculates which column reduces objective the most) Steepest edge (normalize reduced cost with the norm of its column) (I think these were actually slower in the average case because when you have a lot of variables the quick calculations on bland's rule win out)

Ok this is where I am stuck - these things cover us nicely until 1980's and our papers pick up 2010's. I feel like we should have something to fit the in between in terms of advancement of computational complexity of pivot algorithms