Complexity of Pivot Algorithms

Qi Wang, Sean Kelley

December, 2020

Abstract

Abstract...

1 Introduction

Simplex method was invented by Dantzig in 1947 [3] 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 Zionts at 1969 [4] and then [5], [1] 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 [2] 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 Linear optimization problem of the standard form

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

$$x \in \mathbb{R}^n, \ c \in \mathbb{R}^n, \ A \in \mathbb{R}^{m \times n}, \ b \in \mathbb{R}^m$$

(1)

Table 1: Notation

|	
n	number of variables
m	number of constraints (assume $m \leq n$)

3 Complexity - worst case

Given m constraints and n variables, there are at most $\binom{n}{m}$ possible basis, which is a 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 many pivot algorithms will visit all the vertices to find the optimal solution. Detailed introduction of Klee Minty cube can be referred to [6]. Here we demonstrate some simple examples. For such problem
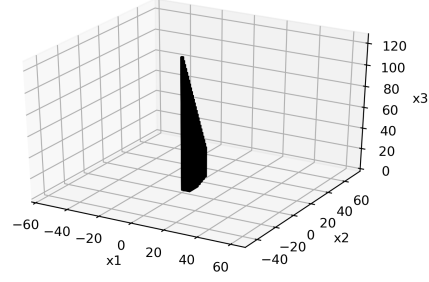
$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} 2^{n-j} x_j \\
\text{s.t.} \quad & 2\sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \leq 5^i && i = 1, \cdots, n \\
& x_j \geq 0 && j = 1, \cdots, n
\end{aligned}
$$

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

$$
\begin{aligned}
\max \quad & 2x_1 + x_2 \\
\text{s.t.} \quad & x_1 \leq 5 \\
& 4x_1 + x_2 \leq 25 \\
& x_1 \geq 0, \ x_2 \geq 0.
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & 4x_1 + 2x_2 + x_3 \\
\text{s.t.} \quad & x_1 \leq 5 \\
& 4x_1 + x_2 \leq 25 \\
& 8x_1 + 4x_2 + x_3 \leq 125 \\
& x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0.
\end{aligned}
$$
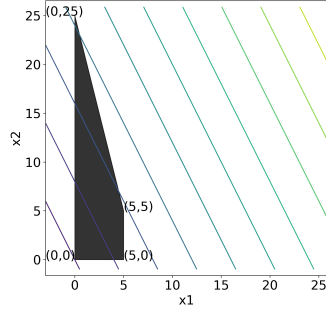
Figure 1: (a) Feasible region of 2-d Klee Minty Cube with contour lines (b) Feasible region of 3-d Klee Minty Cube

The feasible region of 2-d and 3-d Klee Minty problems are shown in Figure 1. It can be illustrated that 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. Note the results of Dantzig pivot rule and least-index rule coincide.

Table 2: Tableau of primal simplex method (least index version), the element with a * 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	
z	-10	0	1	-2	0	at vetex (5,0)
x_1	5	1	0	1	0	
s_2	5	0	1*	-4	1	
z	-15	0	0	2	-1	at vetex (5,5)
x_1	5	1	0	1*	0	
x_2	5	0	1	-4	1	
z	-25	-2	0	0	-1	at vetex (0,25)
s_1	5	1	0	1	0	
x_2	25	4	1	0	1	

4 Conclusion

Conclusion...

References

[1] Yow-Yieh Chang. Least-index resolution of degeneracy in linear complementarity problems. Technical report, STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH, 1979.

[2] Wikipedia Contributors. Klee–minty cube, 11 2020.

[3] George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13:339–347, 1951.

[4] K. Fukuda and T. Terlaky. Criss-cross methods: A fresh view on pivot algorithms. *Mathematical programming*, 79(1-3):369–395, 1997.

[5] T. Terlaky. A finite crisscross method for oriented matroids. *Journal of Combinatorial Theory, Series B*, 42(3):319–327, 1987.

[6] Robert J Vanderbei. *Linear programming: foundations and extensions*, volume 285. Springer Nature, 2020.

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. dantzigs rule - just most negative reduced cost/element in objective row)

what were their complexity? O(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 preprocessing, 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