Complexity of Pivot Algorithms

Qi Wang, Sean Kelley

December 16, 2020

Overview

Analysis of Simplex

- Find our pivot column, $j \in I_N$
  (O(n) operations)
- Find our pivot row, $i \in I_B$
  (O(m) operations)
- Update our constraints by calculating $A_B^{-1}A$
  (O(m^2n) operations†)
- Update objective row by calculating $c_B^T A_B^{-1} A - c^T$
  (O(mn) operations†)

$\binom{n}{m}$ possible bases $\implies O(\binom{n}{m} m^2 n)$ complexity.
$\dagger$ assumes $m << n$

- Find our pivot column, $j \in I_N$
  ($O(n)$ operations)
- Update our constraints by calculating $A_B^{-1}A_j$
  ($O(m^2)$operations†)
- Find our pivot row, $i \in I_B$
  ($O(m)$ operations)
- Update objective row by calculating $c_B^T A_B^{-1} A - c^T$
  ($O(mn)$operations†)

$\binom{n}{m}$ possible bases $\implies$ $O(\binom{n}{m}mn)$ complexity. Can we pick better pivots?
$\dagger$ assumes $m << n$

Other Pivot Rules

- Best improvement rule: choose entering $j := \arg\max_{j \in N} (c_B^T A_B^{-1} A_j - c_j) \times t_j$ where $t_j = \min_{i \in \{B : (A_B^{-1} A_j)_i > 0\}} \frac{x_i}{(A_B^{-1} A_j)_i}$
- Steepest edge rule: choose entering $j := \arg\max_{j \in N} \frac{c_B^T A_B^{-1} A_j - c_j}{\|A_B^{-1} A_j\|}$
- Last-in-first-out rule (LIFO): choose the most recently moved variable.
- Most-often-selected-variable (MOSV): choose the variable that has been selected the largest amount of times before.

Bland's, LIFO, MOSV are all S-monotone index selection rules, which is finite with simplex method and criss-cross.

Worst Case - Klee Minty Cube

Klee-Minty Cube LO problem

$$
\begin{aligned}
\max \sum_{j=1}^{n} 2^{n-j} x_j \\
\text{s.t. } 2\sum_{j=1}^{i-1} 2^{i-j} x_j + x_i \le 5^i, \quad i = 1, \cdots, n \\
x_j \ge 0, \quad j = 1, \cdots, n
\end{aligned}
$$

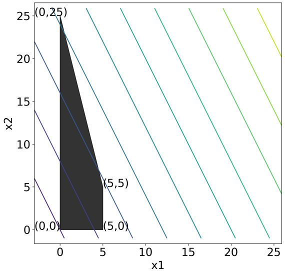
Primal simplex with Dantzig's rule may visit all vertices before finally finding the optimal solution ($2^n - 1$ iterations).

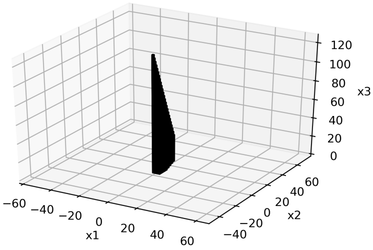
Worst Case - Klee Minty Cube

Klee-Minty Cube LO problem ($n = 2$ and $n = 3$)

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

$$\begin{aligned} \max \quad & 4x_1 + 2x_2 + x_3 \\ \text{s.t.} \quad & x_1 \le 5 \\ & 4x_1 + x_2 \le 25 \\ & 8x_1 + 4x_2 + x_3 \le 125 \\ & x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0. \end{aligned}$$





Solving 2-d Klee Minty Cube using Simplex method with Dantzig's rule

	rhs	x_1	x_2	s_1	s_2	
z	0	2	1	0	0	at vetex (0,0), x_1 enter, s_1 leave
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_2 enter, s_2 leave
x_1	5	1	0	1	0	
s_2	5	0	1^*	-4	1	
z	-15	0	0	2	-1	at vetex (5,5), s_1 enter, x_1 leave
x_1	5	1	0	1^*	0	
x_2	5	0	1	-4	1	
z	-25	-2	0	0	-1	at vetex (0,25), optimal
s_1	5	1	0	1	0	
x_2	25	4	1	0	1	

Kitahara and Mizuno Analysis[**kitahara2013bound**]

Notations:

- BFS: a basic feasible solution for primal standard LO problem.
- δ and γ: 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 $\hat{x}_j \neq 0, \; j \in \{1, \cdots, n\}$ where j represents the jth entry of $\hat{x}$, we have

$$\delta \leq \hat{x}_j \leq \gamma$$

Theorem 1

When applying simplex method with the Dantzig's rule or the best improvement rule for LO having optimal solutions, we encounter at most

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

different basic feasible solutions.

Kitahara and Mizuno Analysis

Corollary 2

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.*

In practice, the ratio of $\frac{\gamma}{\delta}$ is not easy to get prior to solving the problem. However, for some specific LP, $\frac{\gamma}{\delta}$ can be bounded by LP coefficients.

Definition 3

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.

E.g., the node-arc incidence matrix of a directed graph is a totally unimodular matrix.

$$A = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & +1 \\ +1 & 0 & -1 & -1 & 0 & 0 \\ 0 & +1 & +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 & +1 & -1 \end{pmatrix}$$

Kitahara and Mizuno Analysis

With totally unimodular A and integral b, all the elements of any BFS are integers, so

$$\delta \geq 1.$$

For a BFS $x = (x_B, x_N)$, $x_N = 0$ and $x_B = A_B^{-1} b \geq 0$, and A_B^{-1} are all 0,-1,1. Thus for any $j \in B$, we have $x_j \leq \|b\|_1$, so

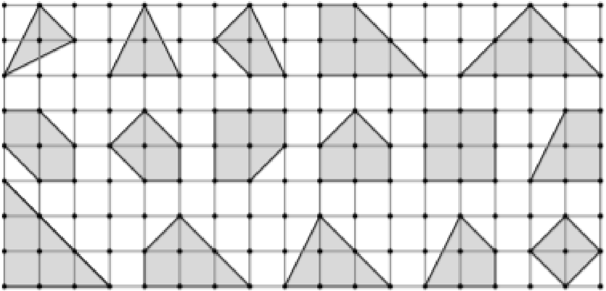
$$\gamma \leq \|b\|_1.$$

Thus $\frac{\gamma}{\delta} \leq \frac{\|b\|_1}{1}$.

Corollary 4

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

Analysis of Del Pia and Michini



Lattice Polytope: a polytope whose vertices are integral.

Analysis of Del Pia and Michini

Del Pia and Michini proved the following bounds on vertices visited when solving an LP over a lattice polytope.

- simplex algorithm: $O(nk\|c\|_\infty)$
- scaling algorithm: $O(nk \log \|c\|_\infty)$
- preprocessing algorithm: $O(n^4 k \log nk)$
- face-fixing algorithm: $O(n^2 k \log nk\alpha)$

Simplex Algorithm:
$x^* = \text{simplex}(P, c, x^0)$

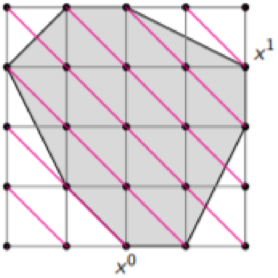
Analysis of Del Pia and Michini

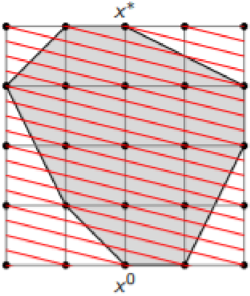
- simplex algorithm: $O(nk\|c\|_\infty)$
- scaling algorithm: $O(nk \log \|c\|_\infty)$
- preprocessing algorithm: $O(n^4 k \log nk)$
- face-fixing algorithm: $O(n^2 k \log nk\alpha)$

Scaling Algorithm:
$x^{n+1} = \text{simplex}(P, c^n, x^n)$

$$
\begin{aligned}
c &= (1,2,3,4,5,6,7) \\
c^0 &= (1,1,1,1,1,1,1) \\
c^1 &= (1,1,1,1,2,2,2) \\
c^2 &= (1,1,2,2,3,3,4) \\
c^3 &= (1,2,3,4,5,6,7)
\end{aligned}
$$

Below is a benefit of scaling.





Analysis of Del Pia and Michini

- simplex algorithm: $O(nk\|c\|_\infty)$
- scaling algorithm: $O(nk \log \|c\|_\infty)$
- preprocessing algorithm: $O(n^4 k \log nk)$
- face-fixing algorithm: $O(n^2 k \log nk\alpha)$

Preprocessing and Face-Fixing Algorithms:
Perturb c to get the most advantage out of the Scaling Algorithm.

Conclusion

- For $k = 1$, Kitahara is still exponential, but Del Pia and Michini are polynomial.
- This result has its caveats.
- A truly polynomial pivot algorithm is yet to be discovered.

References