

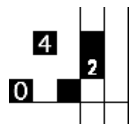
Steven R. Dunbar  
Department of Mathematics  
203 Avery Hall  
University of Nebraska-Lincoln  
Lincoln, NE 68588-0130  
<http://www.math.unl.edu>  
Voice: 402-472-3731  
Fax: 402-472-8466

## Topics in Probability Theory and Stochastic Processes Steven R. Dunbar

---

Worst Case and Average Case Behavior of the Simplex  
Algorithm

---



## Rating

Mathematicians Only: prolonged scenes of intense rigor.

---



## Question of the Day

1. What would be the implications of knowing that worst-case running time of the Simplex Algorithm is exponential in the problem size?
  2. What would knowing the average running time of the Simplex Algorithm tell us about actual practice?
- 



## Key Concepts

- 1.
  - 2.
  - 3.
- 



## Vocabulary

- 1.
  - 2.
  - 3.
- 



# Mathematical Ideas

## The Klee-Minty Example

In 1972, Klee and Minty, [3] considered the linear optimization problem:

$$\begin{aligned} & \max y_n \\ & \text{subject to } 0 \leq y_1 \leq 1 \\ & \quad \epsilon y_{j-1} \leq y_j \leq 1 - \epsilon y_{j-1}, \quad j = 2, \dots, n \\ & \quad y_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

where  $\epsilon \in (0, 1/2)$ . The change of variables

$$x_1 = y_1, x_j = (y_j - \epsilon y_{j-1})/\epsilon^{j-1}, j = 2, \dots, n$$

transforms the problem to

$$\begin{aligned} & \max \sum_{k=1}^n x_k \\ & \text{subject to } 2 \sum_{k=1}^{j-1} x_k + x_j \leq (1/\epsilon)^{j-1}, \quad 1 \leq j \leq n \\ & \quad x_j \geq 0, 1 \leq j \leq n. \end{aligned}$$

which is now in canonical form.

*Proof.* Note that the change of variables

$$x_1 = y_1, x_j = (y_j - \epsilon y_{j-1})/\epsilon^{j-1}$$

is equivalent to the inverse change of variables

$$y_1 = x_1, y_j = \epsilon^{j-1} \sum_{k=1}^j x_k.$$

This may be seen by writing each linear change of variables in matrix form and verifying the corresponding matrices are inverses of each other. Alternatively, substitute the expression for  $y_j$  in terms of the  $x_k$  into the expression for  $x_j$  in terms of  $y_j$ . Obtain

$$x_j = (\epsilon^{j-1} \sum_{k=1}^j x_k - \epsilon \cdot \epsilon^{j-2} \sum_{k=1}^{j-1} x_k) / \epsilon^{j-1} = x_j$$

verifying the inverse relation.

Then the objective function  $\max y_n$  becomes  $\max \epsilon^n \sum_{k=1}^n x_k$  which is equivalent to the rescaled  $\max \sum_{k=1}^n x_k$ . The constraint  $\epsilon y_{j-1} \leq y_j \leq 1 - \epsilon y_{j-1}$  becomes equivalent to

$$\epsilon \cdot \epsilon^{j-2} \sum_{k=1}^{j-1} x_k \leq \epsilon^{j-1} \sum_{k=1}^j x_k \leq 1 - \epsilon \cdot \epsilon^{j-2} \sum_{k=1}^{j-1} x_k.$$

This can be rewritten as

$$0 \leq \epsilon^{j-1} x_j \leq 1 - 2\epsilon^{j-1} \sum_{k=1}^{j-1} x_k.$$

Since all variables are required to be positive, the inequality

$$0 \leq \epsilon^j x_j + 2\epsilon^j \sum_{k=1}^{j-1} x_k$$

is automatic, so the constraint becomes

$$\epsilon^{-1} j x_j + 2\epsilon^{j-1} \sum_{k=1}^{j-1} x_k \leq 1.$$

This proof is the solution of Problem 8.5, page 427 of Bazaraa, Jarvis, Sherali, [1]. □

If we pivot using Dantzig's rule that the element in the vector of modified costs which is most negative determines the entering variable  $j$ , then the Simplex Algorithm will go through  $2^n - 1$  pivots on this example. The fairly straightforward but tedious proof is in [1]. The idea is that for this canonical problem the Simplex Algorithm systematically traces through the  $2^n$  vertices of a feasible set which is a distorted cube, first the  $2^{n-1}$  vertices on the “face” in the  $y_n = 0$  plane, then subsequently through the  $2^{n-1}$  vertices in the “opposite face”.

Following [2], another formulation of the Klee-Minty example due to Chvátal is achieved with the change of variables  $x_1 = y_1$ ,  $x_j = (-1/\epsilon)y_{j-1} +$

$y_j$ . When  $\epsilon = 1/10$  the problem becomes

$$\begin{aligned} & \max \sum_{k=1}^n 10^{n-k} x_k \\ & \text{subject to } 2 \sum_{k=1}^{j-1} 10^{j-k} x_k + x_j \leq 100^{j-1}, \quad 1 \leq j \leq n \\ & \quad x_j \geq 0, 1 \leq j \leq n. \end{aligned}$$

For example, the  $n = 3$  case is

$$\begin{aligned} & \max 100x_1 + 10x_2 + x_3 \\ & \text{subject to } x_1 \leq 1 \\ & \quad 20x_1 + x_2 \leq 100 \\ & \quad 200x_1 + 20x_2 + x_3 \leq 10,000 \\ & \quad x_1, x_2, x_3 \geq 0 \end{aligned}$$

which is quite similar to the first linear optimization problem with a change of scale. This formulation makes the distortion of the unit cube more evident. The initial basic feasible solution is  $(0, 0, 0, 1, 100, 10,000)$ .

According to Bazaraa, Jarvis and Sherali, [1], in 1973 Jeroslow showed the existence of problem classes that take an exponential number of pivots with the maximum improvement entering criterion as well. According to Karloff [2], Chvátal proved in 1978 that the Simplex Algorithm requires exponentially many pivots in the worst case if Bland's pivoting selection rule is used. The root of the problem is in the local viewpoint of the Simplex Algorithm whereby decisions are based on the local combinatorial structure of the polytope and the motion is restricted to an edge path.

According to Spielman and Teng [5], "almost all existing pivot rules are known to have exponential worst-case complexity".

## Empirical Results

Karloff, [2], claims that in practice for the Simplex Algorithm "the number of pivots seems to be between  $4m$  and  $6m$  for Phase I and Phase II together. Rarely does either phase require more than  $10m$  pivots. As  $n$  grows, the number of pivots seems to grow slowly, perhaps logarithmically with  $n$ ".

Bazaraa, Jarvis and Sherali, [1], claim that the Simplex Algorithm “has been empirically observed to take roughly  $3m/2$  iterations and seldom more than  $3m$  iterations”. By regression analysis, the performance of the Simplex Algorithm is described by  $Km^{2.5}nd^{0.33}$  where  $d$  is the density of the matrix  $A$ .

The Mathworld.com Simplex Method. article says the “simplex method is very efficient in practice, generally taking  $2m$  to  $3m$  iterations at most (where  $m$  is the number of equality constraints)”, citing references.

All these results point to the observation that practically speaking, not only does the Simplex Algorithm run in polynomial time, it runs in a small multiple of the number of constraints! The question now naturally arises as to whether there is an explanation for that practical effective running time in spite of the fact that exceptional examples actually run in an exponential power of the number of variables.

## A Markov Chain Model of the Simplex Algorithm

Consider the following linear optimization problem in standard form:

$$\begin{aligned} &\min \mathbf{c}\mathbf{x}, \\ &\text{subject to } A\mathbf{x} = \mathbf{b}, \\ &\mathbf{x} \geq \mathbf{0}. \end{aligned}$$

$A$  is an  $m \times n$  matrix of real numbers,  $\mathbf{c} = (c_1, \dots, c_n)$  and  $\mathbf{b} = (b_1, \dots, b_m)$  are constant vectors of real numbers, and  $\mathbf{x} = (x_1, \dots, x_n)$  is the  $n$  vector of non-negative values that is to be determined to minimize the linear objective  $\mathbf{c}\mathbf{x} = \sum_{j=1}^n c_j x_j$ . We know from the elementary theory of linear optimization that the minimum occurs at an extreme point of the feasible region  $A\mathbf{x} = \mathbf{b}$ ,  $\mathbf{x} \geq \mathbf{0}$ .

Under the natural assumptions that  $n > m$  and that the matrix  $A$  is full rank, we can find feasible extreme point solutions of the constraint set by setting  $n - m$  of the entries in  $\mathbf{x}$  to be 0. Note that there can be as many as

$$N = \binom{n}{m}$$

possible extreme points.

## A Heuristic Markov Chain Model

Here is a simple Markov Chain model for how the Simplex Algorithm moves from extreme point to extreme point on the feasible region. Suppose that if at some time the algorithm is at the  $i$ th best extreme point, then after the next pivot operation of the simplex Algorithm the resulting extreme point is equally likely to be any of the remaining  $i-1$  best. This is a probability model that covers all pivoting rules. That is, we model the transition from extreme point to subsequent extreme point as a Markov chain for which  $P_{11} = 1$  and

$$P_{ij} = \frac{1}{i-1}, \quad j = 1, \dots, i-1; 1 < i \leq N$$

and naturally,  $P_{ij} = 0$  for all other indices.

Let  $T_i$  denote the number of transitions needed to go from state  $i$  to state 1. Obtain a recursive formula for  $\mathbb{E}[T_i]$  by conditioning on the initial transformation:

$$\mathbb{E}[T_i] = 1 + \frac{1}{i-1} \sum_{j=1}^{i-1} \mathbb{E}[T_j]$$

Note that  $\mathbb{E}[T_1] = 0$ . Then recursively,

$$\mathbb{E}[T_2] = 1 + \frac{1}{1}(0) = 1,$$

$$\mathbb{E}[T_3] = 1 + \frac{1}{2}(0 + 1) = 1 + \frac{1}{2},$$

$$\mathbb{E}[T_4] = 1 + \frac{1}{3}(0 + 1 + 1 + 1/2) = 1 + 1/2 + 1/3,$$

$$\mathbb{E}[T_5] = 1 + \frac{1}{4}(0 + 1 + 1 + 1/2 + 1 + 1/2 + 1/3) = 1 + 1/2 + 1/3 + 1/4.$$

Then we can inductively guess and prove that

$$\mathbb{E}[T_i] = \sum_{k=1}^{i-1} \frac{1}{k} = H_{i-1}$$

where  $H_n = \sum_{k=1}^n \frac{1}{k}$  is the  $n$ th harmonic number.

*Proof.* Use the strong induction hypothesis that  $\mathbb{E}[T_j] = H_{j-1}$  for all  $j < i$ .

Then

$$\begin{aligned}
\mathbb{E}[T_i] &= 1 + \frac{1}{i-1} \sum_{j=1}^{i-1} \mathbb{E}[T_j] \\
&= 1 + \frac{1}{i-1} \sum_{j=1}^{i-1} H_{j-1} \\
&= 1 + \frac{1}{i-1} \sum_{j=1}^{i-2} H_j \\
&= 1 + \frac{1}{i-1} \sum_{j=1}^{i-2} \sum_{k=1}^j \frac{1}{k} \\
&= 1 + \frac{1}{i-1} \sum_{k=1}^{i-2} \sum_{j=k}^{i-2} \frac{1}{k} \\
&= 1 + \frac{1}{i-1} \left( \sum_{k=1}^{i-2} 1 + \sum_{k=2}^{i-2} \sum_{j=k}^{i-2} \frac{1}{k} \right) \\
&= 1 + \frac{1}{i-1} \left( \sum_{k=1}^{i-2} 1 + \sum_{k=2}^{i-2} \frac{i-k-1}{k} \right) \\
&= 1 + \frac{1}{i-1} + \frac{1}{i-1} \sum_{k=2}^{i-2} \left( 1 + \frac{i-k-1}{k} \right) \\
&= 1 + \frac{1}{i-1} + \frac{1}{i-1} \sum_{k=2}^{i-2} \frac{i-1}{k} \\
&= 1 + \frac{1}{i-1} + \sum_{k=2}^{i-2} \frac{1}{k} \\
&= H_{i-1}
\end{aligned}$$

□

From well-known inequalities bounding  $H_{i-1}$ , (Mathworld.com Harmonic Number.) we observe that

$$\log(i-1) + \gamma + \frac{1}{2i} < \mathbb{E}[T_i] < \log(i-1) + \gamma + \frac{1}{2(i-1)}$$



From that we can deduce that  $\mathbb{E}[T_i] \sim \log(i - 1)$  in general. Therefore  $\mathbb{E}[T_N] \sim \log(N - 1) \sim \log(N) \sim m[1 + \log(n/m - 1)]$  by Stirling's approximation applied to  $\binom{n}{m}$  (see below for more derivation) as an asymptotic bound on the expected value for the number of pivot operations required.

Note that this asymptotic result is roughly consistent with Karloff's observation above.

### Approximate Distribution of Number of Pivots

We can obtain an approximate distribution for  $T_N$  for large  $N$  using the Central Limit Theorem. Continue the assumption that if at some time the algorithm is at the  $i$ th best extreme point, then after the next pivot operation of the Simplex Algorithm the resulting extreme point is equally likely to be any of the remaining  $i - 1$  best. Let

$$I_j = \begin{cases} 1, & \text{if the Simplex Algorithm ever enters state } j \\ 0 & \text{otherwise} \end{cases}$$

be an indicator random variable. Then  $T_N = \sum_{k=1}^{N-1} I_k$ .

**Proposition 1.**  $I_1, \dots, I_{N-1}$  are independent and  $\mathbb{P}[I_j = 1] = 1/j$  for  $1 \leq j \leq N - 1$ .

*Proof.* Let  $j$  be fixed. Given  $I_{j+1} \dots I_N$ , let  $k = \min\{i : i > j, I_i = 1\}$  indicate the lowest numbered state, greater than  $j$ , that is entered. That is, we know that the process enters state  $k$ , and that the next state to be entered is any of the states  $1, 2, \dots, j$ . Hence, as the next state to be entered from state  $k$  is equally likely to be any of the lower number states  $1, 2, \dots, k - 1$ , we see that

$$\mathbb{P}[I_j = 1 | I_{j+1}, \dots, I_N] = \frac{1/(k - 1)}{j/(k - 1)} = 1/j$$

Hence  $\mathbb{P}[I_j = 1] = 1/j$  and independence follows since the preceding conditional probability does not depend on  $I_{j+1}, \dots, I_N$ .  $\square$

**Corollary 1.** 1.  $\mathbb{E}[I_j] = 1/j$ ,

2.  $\mathbb{E}[T_N] = \sum_{k=1}^{N-1} 1/k = H_{N-1}$ ,

3.  $\text{Var}[T_N] = \sum_{k=1}^{N-1} (1/k)(1 - 1/k)$ ,

4. For  $N$  large,  $T_N$  has approximately a normal distribution with mean  $\log(N)$  and variance  $\log(N)$ .

*Proof.* 1. Automatic, since  $\mathbb{P}[I_j = 1] = 1/j$ .

2. Automatic by the representation  $T_N = \sum_{k=1}^{N-1} I_k$ .

3. Automatic by the representation  $T_N = \sum_{k=1}^{N-1} I_k$  and the independence of the  $I_j$ .

4. Follows from the Central Limit Theorem (basic version with independent random variables having finite variance) and the standard fact that

$$\log(N) < \sum_{k=1}^{N-1} 1/k < 1 + \log(N-1).$$

□

Note that if  $n$ ,  $m$  and  $n-m$  are all large, then by Stirling's approximation we have

$$N = \binom{n}{m} \sim \frac{n^{n+1/2}}{(n-m)^{(n-m)+1/2} \cdot m^{m+1/2} \sqrt{2\pi}}.$$

Therefore

$$\begin{aligned} \log(N) &\sim \left(m \frac{n}{m} + 1/2\right) \log(m(n/m)) \\ &\quad - (m(n/m - 1) + 1/2) \log(m(n/m - 1)) - (m + 1/2) \log(m) - \end{aligned}$$

or

$$\log(N) \sim m \left( \frac{n}{m} \log \left( \frac{n/m}{n/m - 1} \right) + \log(n/m - 1) \right).$$

Now since  $\lim_{x \rightarrow \infty} x \log(x/(x-1)) = 1$ , it follows that

$$\log(N) \sim m [1 + \log(n/m - 1)].$$

## Sources

The information on the Klee-Minty example and the empirical observations of the number of pivots in practice is adapted from Bazaraa, Jarvis and Sherali, [1] and Karloff, [2]. The Markov Chain Model of the Simplex Algorithm is slightly adapted from Ross's Introduction to Probability Models, [4].

---



## Problems to Work for Understanding

---



## Reading Suggestion:

## References

- [1] Mokhtar S. Bazaraa, John J. Jarvis, and Hanif. D. Sherali. *Linear Programming and Network Flows*. Wiley Interscience, 3rd edition, 2005.
- [2] Howard Karloff. *Linear Programming*. Birkhäuser, 1991. Linear Optimization.
- [3] Victor Klee and G. J. Minty. How good is the simplex algorithm. In O. Shisha, editor, *Inequalities - III*, pages 159–175. Academic Press, 1972.
- [4] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 9th edition edition, 2006.
- [5] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):77–84, October 2009.



## Outside Readings and Links:

- 1.
- 2.
- 3.
- 4.

---

I check all the information on each page for correctness and typographical errors. Nevertheless, some errors may occur and I would be grateful if you would alert me to such errors. I make every reasonable effort to present current and accurate information for public use, however I do not guarantee the accuracy or timeliness of information on this website. Your use of the information from this website is strictly voluntary and at your risk.

I have checked the links to external sites for usefulness. Links to external websites are provided as a convenience. I do not endorse, control, monitor, or guarantee the information contained in any external website. I don't guarantee that the links are active at all times. Use the links here with the same caution as you would all information on the Internet. This website reflects the thoughts, interests and opinions of its author. They do not explicitly represent official positions or policies of my employer.

Information on this website is subject to change without notice.

Steve Dunbar's Home Page, <http://www.math.unl.edu/~sdunbar1>

Email to Steve Dunbar, `sdunbar1` at `unl dot edu`

Last modified: Processed from L<sup>A</sup>T<sub>E</sub>X source on January 20, 2011