The Simplex Method

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

LY, Chapters 2.3-2.4, 4.1-4.2

Geometry of Linear Programming (LP)

Consider the following LP problem in the standard inequality form:

$$
\begin{array}{llll}
\text{maximize} & x_1 & +2x_2 & \\
\text{subject to} & -x_1 & & \leq 0 \\
 & & x_2 & \leq 1 \\
 & x_1 & +x_2 & \leq 1.5 \\
 & x_1 & & \leq 1 \\
 & & -x_2 & \leq 0.
\end{array}
$$

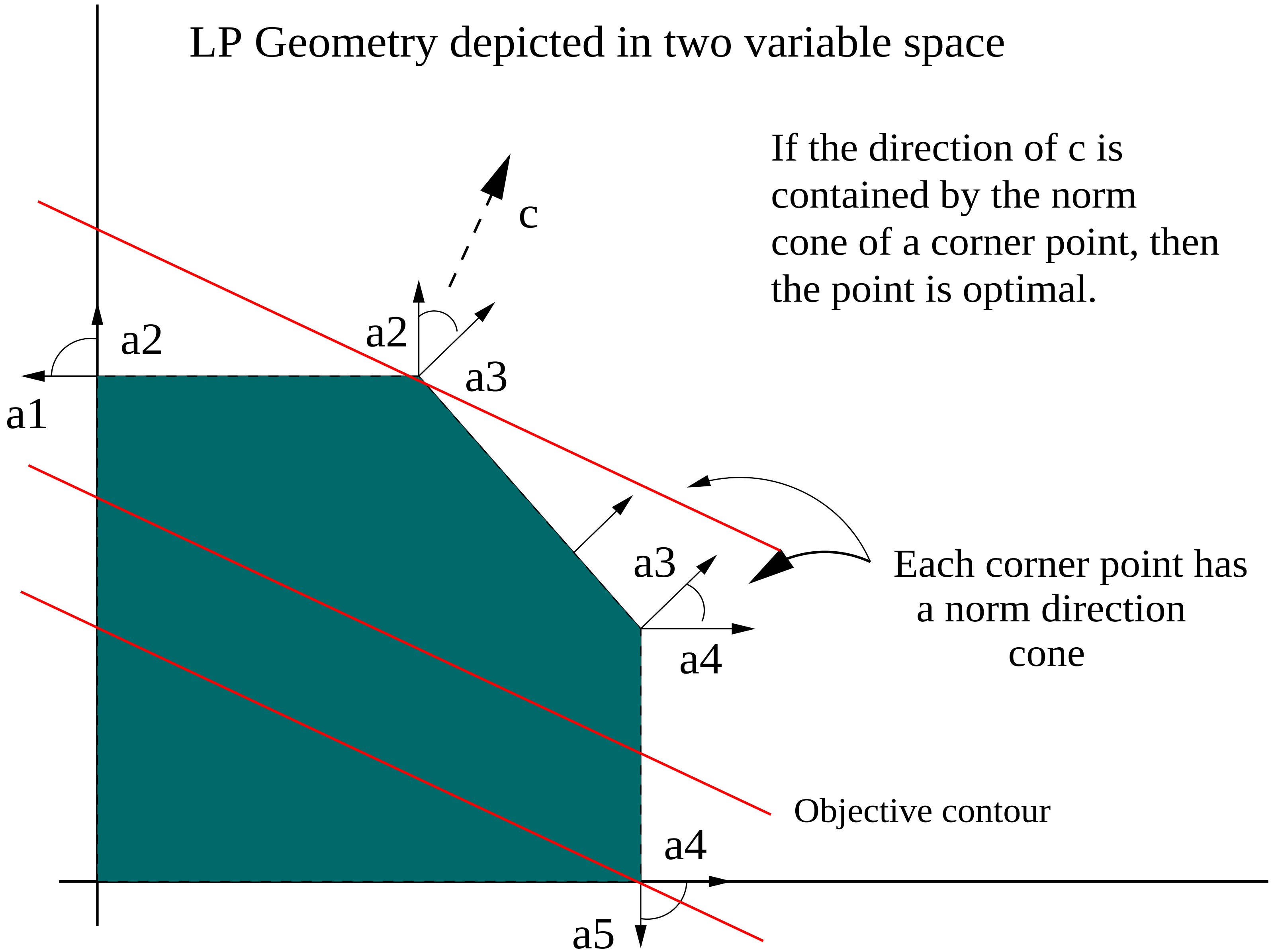

Figure 1: Feasible region with objective contours

Optimality Test of LP in Inequality Form

Consider an LP with m variables and n linear inequality constraints.

- A Corner Point is an intersection point of the hyperplanes of m linearly-independent inequality constraints.
- These constraints are called active or binding constraints at the corner solution.
- Two corner solutions are adjacent if they differ by one active constraint.
- **Theorem 1** *For LP in the standard form, a Corner Point is maximal if and only if the objective vector is a conic combination of the normal direction vectors of the m hyperplanes.*

History of the Simplex Method

George B. Dantzig's Simplex Method for LP stands as one of the most significant algorithmic achievements of the 20th century. It is now over 50 years old and still going strong.

The basic idea of the simplex method to confine the search to corner points of the feasible region (of which there are only finitely many) in a most intelligent way. In contrast, interior-point methods will move in the interior of the feasible region, hoping to by-pass many corner points on the boundary of the region.

The key for the simplex method is to make computers see corner points; and the key for interior-point methods is to stay in the interior of the feasible region.

From Geometry to Algebra

- How to make computer recognize a corner point?
- How to make computer tell that two corners are neighboring?
- How to make computer terminate and declare optimality?
- How to make computer identify a better neighboring corner?

LP Standard (Equality) Form

$$
\begin{array}{rl}
\text{minimize} & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
\text{subject to} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\
& a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m, \\
& x_j \geq 0, \quad j = 1, 2, \ldots, n.
\end{array}
$$

Equivalently,

$$
\begin{array}{rrl}
(LP) & \text{minimize} & \mathbf{c}^T\mathbf{x} \\
& \text{subject to} & A\mathbf{x} = \mathbf{b}, \\
& & \mathbf{x} \geq \mathbf{0}.
\end{array}
\qquad
\begin{array}{rrl}
(LD) & \text{maximize} & \mathbf{b}^T\mathbf{y} \\
& \text{subject to} & A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \\
& & \mathbf{s} \geq \mathbf{0}.
\end{array}
$$

Reduction to the Standard Form

- Eliminating ”free” variables: substitute with the difference of two nonnegative variables

$$x = x^+ - x^- \quad \text{where} \quad x^+, x^- \geq 0.$$

- Eliminating inequalities: add slack variables

$$\mathbf{a}^T\mathbf{x} \leq b \quad \Leftrightarrow \quad \mathbf{a}^T\mathbf{x} + s = b, \quad s \geq 0,$$

$$\mathbf{a}^T\mathbf{x} \geq b \quad \Leftrightarrow \quad \mathbf{a}^T\mathbf{x} - s = b, \quad s \geq 0.$$

- Eliminating upper bounds: move them to constraints $x \leq 3 \quad \Leftrightarrow \quad x + s = 3, \quad s \geq 0.$
- Eliminating nonzezro lower bounds: shift the decision variables $x \geq 3 \quad \Rightarrow \quad x := x - 3.$
- Change $\max\ \mathbf{c}^T\mathbf{x}$ to $\min\ -\mathbf{c}^T\mathbf{x}$.

An LP Example in Standard Form

$$\begin{array}{llllllll}
\text{minimize} & -x_1 & -2x_2 & & & & \\
\text{subject to} & x_1 & & +x_3 & & & =1 \\
& & x_2 & & +x_4 & & =1 \\
& x_1 & +x_2 & & & +x_5 & =1.5 \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0.
\end{array}$$

Basic and Basic Feasible Solution (BFS)

In the LP standard form, let's assume that we selected m linearly independent columns, denoted by the index set B from A and solve

$$A_B \mathbf{x}_B = \mathbf{b}$$

for the m-vector $\mathbf{x}_B$. By setting the variables $\mathbf{x}_N$ of $\mathbf{x}$ corresponding to the remaining columns of A equal to zero, we obtain a solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$.

Then $\mathbf{x}$ is said to be a basic solution to (LP) with respect to basis A_B. The components of $\mathbf{x}_B$ are called basic variables and those of $\mathbf{x}_N$ are called nonbasic variables.

Two basic solutions are adjacent if they differ by exactly one basic (or nonbasic) variable. If a basic solution satisfies $\mathbf{x}_B \geq \mathbf{0}$, then $\mathbf{x}$ is called a basic feasible solution (BFS), and it is an extreme point of the feasible region.

If one or more components in $\mathbf{x}_B$ has value zero, $\mathbf{x}$ is said to be degenerate.

Geometry vs Algebra

Theorem 2 *Consider the polyhedron in the standard LP form. Then a basic feasible solution and a corner point are equivalent; the former is algebraic and the latter is geometric.*

In the LP example:

Basis	3,4,5	1,4,5	3,4,1	3,2,5	3,4,2	1,2,3	1,2,4	1,2,5
Feasible?	$\surd$	$\surd$		$\surd$		$\surd$	$\surd$	
x_1, x_2	$0, 0$	$1, 0$	$1.5, 0$	$0, 1$	$0, 1.5$	$.5, 1$	$1, .5$	$1, 1$

Neighboring Basic Solutions:

- Two basic solutions are neighboring or adjacent if they differ by exactly one basic (or nonbasic) variable.
- A basic feasible solution is optimal if no "better" neighboring BFS exists, or the direction to each neighbor BFS is non-improving.

Theorem 3 *(The Fundamental Theorem of LP in Algebraic form)*[a] *Given (LP) and (LD) where A has full row rank m,*

i) *if there is a feasible solution, there is a basic feasible solution (Carathéodory's theorem);*

ii) *if there is an optimal solution, there is an optimal basic solution.*

The simplex method is to proceed from one BFS (a corner point of the feasible region) to an adjacent or neighboring one by choosing exactly one of non-basic variables to increase its value, in such a way as to continuously improve the value of the objective function.

We now prove ii) in the next slide.

[a] Text p.21

Rounding to an Optimal Basic Feasible Solution for LP

1. Start with any optimal solution $\mathbf{x}^0 = \mathbf{x}^*$ after removing all variables with the zero value, thus $\mathbf{x}^0 > \mathbf{0}$. Let A^0 be the remaining constraint matrix corresponding to $\mathbf{x}^0$ such that $A^0\mathbf{x}^0 = \mathbf{b}$. If the columns of A^0 is linearly independent, then $\mathbf{x}^0$ is already a BFS (if $|\mathbf{x}^0| < m$ then fill any rest of independent columns to make it a basis so that the BFS is degenerate). Otherwise, let $k = 0$ and go to the next step.

2. Find any vector $\mathbf{d}$ such that $A^k\mathbf{d} = \mathbf{0},\ \mathbf{d} \neq \mathbf{0}$, and let $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha\mathbf{d}$ where α is chosen so that $\mathbf{x}^{k+1} \geq \mathbf{0}$ and at least one entry of $\mathbf{x}^{k+1}$ equals 0.

3. Remove all zero entries from $\mathbf{x}^{k+1}$ and let A^{k+1} be the remaining constraint matrix corresponding to $\mathbf{x}^{k+1}$ such that $A^{k+1}\mathbf{x}^{k+1} = \mathbf{b}$.

4. If the columns of A^{k+1} is linearly independent, then stop with the BFS $\mathbf{x}^{k+1}$; otherwise set $k := k + 1$ and return to Step 2.

The final output would be a BFS, and it should be optimal because it remains complementary to any optimal dual vector that is complementary to the initial $\mathbf{x}^*$.

Optimality Test of the Current BFS I

Suppose the basis of a BFS is A_B and the rest is A_N. One can express $\mathbf{x}_B$ (dependent variables) in terms of $\mathbf{x}_N$ (independent variables) using the equality constraint:

$$A_B^{-1}A\mathbf{x} = A_B^{-1}\mathbf{b}, \quad (\text{or } \mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N)$$

where the independent $\mathbf{x}_N$ represent the degree of freedom variables.

Then the objective function can be equivalently reduced to

$$\begin{aligned} \mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N &= \mathbf{c}_B^T A_B^{-1}\mathbf{b} - \mathbf{c}_B^T A_B^{-1}A_N\mathbf{x}_N + \mathbf{c}_N^T\mathbf{x}_N \\ &= \mathbf{c}_B^T A_B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1}A_N)\mathbf{x}_N. \end{aligned}$$

The reduced objective depends on independent variables in $\mathbf{x}_N$ only, where

$$\mathbf{r} = \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1}A$$

is called the reduced gradient or cost-vector of $\mathbf{x}$.

Note that the reduced $\mathbf{r}_B = \mathbf{c}_B^T - \mathbf{c}_B^T A_B^{-1}A_B = \mathbf{0}$.

Optimality Test of the Current BFS II

The reduced gradient vector $\mathbf{r} \in R^n$ can be written by

$$\mathbf{r} = \mathbf{c} - \bar{A}^T \mathbf{c}_B = \mathbf{c} - A^T (A_B^{-1})^T \mathbf{c}_B = \mathbf{c} - A^T \mathbf{y},$$

where $\mathbf{y} = (A_B^{-1})^T \mathbf{c}_B$ is called the shadow price or dual vector corresponding to the current BFS.

Theorem 4 *If $\mathbf{r}_N \geq \mathbf{0}$ (equivalently $\mathbf{r} \geq \mathbf{0}$) at a BFS with basic variable set B, then the BFS $\mathbf{x}$ is a primal optimal basic solution and $\mathbf{y}$ is a dual optimal basic solution, both with A_B being an optimal basis.*

The proof is simply from the primal feasibility, dual feasibility and complementarity where dual slack vector $\mathbf{s} = \mathbf{r}$.

In fact, it can also be seen from the necessary and sufficient conditions for

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \;\; \text{s.t.} \;\; \mathbf{x} \geq \mathbf{0}$$

are that $\nabla f(\mathbf{x}) \geq \mathbf{0}$ and it is complementary to $\mathbf{x} \geq \mathbf{0}$, as long as f is a differentiable convex function.

In the LP Example, let the basic variable set $B = \{3, 4, 5\}$ so that

$$A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I$$

and

$$A_B^{-1} = I; \quad \mathbf{c}_B = (0;\ 0;\ 0)$$

$$\mathbf{y}^T = (0,\ 0,\ 0) \quad \text{and} \quad \mathbf{r}^T = (-1,\ -2,\ 0,\ 0,\ 0)$$

The corresponding optimal corner solution is $(x_1, x_2) = (0,\ 0)$ in the original problem, and it is not optimal.

By increasing either of (x_1, x_2), one can further improve the objective value.

However, in the LP Example, let the basic variable set $B = \{1, 2, 3\}$ so that

$$A_B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

and

$$A_B^{-1} = \begin{pmatrix} 0 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & -1 \end{pmatrix}$$

$$\mathbf{y}^T = (0,\ -1,\ -1) \quad \text{and} \quad \mathbf{r}^T = (0,\ 0,\ 0,\ 1,\ 1)$$

The corresponding optimal corner solution is $(x_1, x_2) = (0.5,\ 1)$ in the original problem, and it is optimal.

Changing Basis for the LP Example

With the initial basic variable set $B = \{3, 4, 5\}$, let us choose x_1 (entering variable) to increase. Then we have

$$\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \frac{3}{2} \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} x_1.$$

How much can we increase x_1 while the current basic variables remain feasible (or non-negative)?

In order to keep all variables nonnegative, the largest value is $x_1 = 1$. Then we see x_3 becomes zero and it is called outgoing variable. This leads to a new basis $B = \{1, 4, 5\}$

The largest increase of the entering variable can be determined by minimum ratio test (MRT) procedure.

Minimum Ratio Test (MRT)

- Select the entering variable x_e with its reduced cost $r_e < 0$;
- If column $(A_B)^{-1}A_{.e} \le 0$, then Objective is Unbounded
- Minimum Ratio Test (MRT): the largest increase is θ where

$$\theta := \min\left\{\frac{((A_B)^{-1}\mathbf{b})_i}{(A_B)^{-1}A_{.e}} : (A_B)^{-1}A_{.e} > 0\right\}$$

- When the entering variable reaches θ, select any current basic variables who reaches zero value.
- If there are multiple basic variables reach zero value, the new BFS will be degenerate (one of its basic variable has value 0). Pretending it is $\epsilon > 0$ but arbitrarily small and continue the process.

The Simplex Algorithm

0. Initialize with a minimization problem in feasible canonical form with respect to a basic index set B. Let N denote the complementary index set.

1. Test for termination: Compute $\mathbf{x}_B = (A_B)^{-1}\mathbf{b} \geq \mathbf{0}$, $\mathbf{y}^T = \mathbf{c}_B^T(A_B)^{-1}$, and $\mathbf{r} = \mathbf{c} - A^T\mathbf{y}$.

2. Select
$$r_e = \min_{j \in N}\{r_j\}.$$
If $r_e \geq 0$, stop. The solution is optimal.

3. Otherwise determine whether the vector $(A_B)^{-1}A_{.e}$ contains a positive entry. If not, the objective function is unbounded below – terminate; otherwise, let x_e be the entering basic variable.

4. Determine the outgoing: execute the MRT to determine the outgoing variable x_o.

5. Update basis: update B and A_B and return to Step 1.

How Good is the Simplex Method

Very good on average, but the worse case ...?

When the simplex method is used to solve a linear program the number of iterations to solve the problem starting from a basic feasible solution is typically a small multiple of m, e.g., between $2m$ and $3m$.

At one time researchers believed—and attempted to prove—that the simplex algorithm (or some variant thereof) always requires a number of iterations that is bounded by a polynomial expression in the problem size.

Klee and Minty Example

Consider

$$\begin{array}{lll} \max & x_n & \\ \text{subject to} & x_1 \geq 0 & \\ & x_1 \leq 1 & \\ & x_j \geq \epsilon\, x_{j-1} & j = 2, \ldots, n \\ & x_j \leq 1 - \epsilon\, x_{j-1} & j = 2, \ldots, n \end{array}$$

where $0 < \epsilon < 1/2.$ This presentation of the problem emphasizes the idea (see the figures below) that the feasible region of the problem is a perturbation of the n-cube.

The formulation above does not immediately reveal the standard form representation of the problem. Instead, we consider a different one, namely

$$
\begin{array}{lrll}
\max & \sum_{j=1}^{n} 10^{n-j} x_j & & \\
\text{subject to} & 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i & \le 100^{i-1} & i = 1, \ldots, n \\
& x_j & \ge 0 & j = 1, \ldots, n
\end{array}
$$

The problem above[a] also be used is easily cast as a linear program in standard form. Unfortunately, it is less apparent how to exhibit the relationship between its feasible region and a perturbation of the unit cube.

[a] It should be noted that there is no need to express this problem in terms of powers of 10. Using any constant $C > 1$ would yield the same effect (an exponential number of pivot steps).

Example

$$\begin{array}{lrcrcrcr}
\max & 100x_1 & + & 10x_2 & + & x_3 & & \\
\text{subject to} & x_1 & & & & & \leq & 1 \\
& 20x_1 & + & x_2 & & & \leq & 100 \\
& 200x_1 & + & 20x_2 & + & x_3 & \leq & 10,000
\end{array}$$

In this case, we have three constraints and three variables (along with their non-negativity constraints). After adding slack variables, we get a problem in standard form. The system has $m = 3$ equations and $n = 6$ nonnegative variables. In tableau form, the problem is

T^0

	$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
	1	100	10	1	0	0	0	0
T^0	0	1	0	0	1	0	0	1
	0	20	1	0	0	1	0	100
	0	200	20	1	0	0	1	10,000
					•	•	•	

The bullets below the tableau indicate the columns that are basic.

	$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
	1	0	10	1	–100	0	0	–100
T^1	0	1	0	0	1	0	0	1
	0	0	1	0	–20	1	0	80
	0	0	20	1	–200	0	1	9,800
		•				•	•	

T^2

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	0	1	100	–10	0	–900
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	0	1	200	–20	1	8,200
	●	●				●	

T^3

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	–100	0	1	0	–10	0	–1,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	–200	0	1	0	–20	1	8,000
		•		•		•	

T^4

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	100	0	0	0	10	–1	–9,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	–200	0	1	0	–20	1	8,000
		•	•	•			

T^5

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	0	0	–100	10	–1	–9,100
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	0	1	200	–20	1	8,200
	•	•	•				

T^6

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	–10	0	100	0	–1	–9,900
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	20	1	–200	0	1	9,800
	●		●		●		

T^7

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	–100	–10	0	0	0	–1	–10,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	200	20	1	0	0	1	10,000
			•	•	•		

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 0, 10^4, 1, 10^2, 0)$$

is optimal and that the objective function value is $10,000.$

Along the way, we made $2^3 - 1 = 7$ pivot steps. The objective function made a strict increase with each change of basis.

Remark. The instance of the linear program (1) in which $n = 3$ leads to $2^3 - 1$ pivot steps when the greedy rule is used to select the pivot column. The general problem of the class (1) takes $2^n - 1$ pivot steps. To get an idea of how bad this can be, consider the case where $n = 50$. Now $2^{50} - 1 \approx 10^{15}$. In a year with 365 days, there are approximately 3×10^7 seconds. If a computer were running continuously and performing T iterations of the Simplex Algorithm per second, it would take approximately

$$\frac{10^{15}}{3T \times 10^8} = \frac{1}{3T} \times 10^8 \quad \text{years}$$

to solve the problem using the Simplex Algorithm with the greedy pivot selection rule.

Resolving Cycling in the Simplex Algorithm

In a system of rank m, a (basic) solution that uses fewer than m columns to represent the right-hand side vector is said to be degenerate. Otherwise, it is called nondegenerate.

A basic feasible solution will be nondegenerate if and only if its m basic variables are positive.

Why is degeneracy a problem? The Simplex Algorithm can cycling (an infinite repetition of a finite sequence of bases) when a degenerate basic feasible solution crops up in the course of executing the algorithm, unless a suitable rule is employed to break the ties. Fortunately, there are rules to overcome this problem.

Cycling Example

$$
\begin{array}{lrcrcrcrrrl}
\min & -2x_1 & - & 3x_2 & + & x_3 & + & 12x_4 & & & \\
\text{s.t.} & -2x_1 & - & 9x_2 & + & x_3 & + & 9x_4 & +x_5 & & =0 \\
 & \frac{1}{3}x_1 & + & x_2 & - & \frac{1}{3}x_3 & - & 2x_4 & & +x_6 & =0 \\
 & x_1, & & x_2, & & x_3, & & x_4, & x_5, & x_6 & \geq 0
\end{array}
$$

Initially, the basic variables are $\{x_5, x_6\}$ and it is in the canonical form. The pivot sequence shown in the table below leads back to the original system after 6 pivots.

Pivot number	1	2	3	4	5	6
Basic var. out	x_6	x_5	x_2	x_1	x_4	x_3
Basic var. in	x_2	x_1	x_4	x_3	x_6	x_5

Methods for Resolving Cycling

There are several methods for resolving degeneracy in linear programming. Among these are:

1. Perturbation of the right-hand side.
2. Lexicographic ordering.
3. Application of Bland's pivot selection rule.

Bland's Rule

It is a double least-index rule consisting of the following two parts:

(i) Among all candidates for the entering column (i.e., those with $r_j < 0$), choose the one with the smallest index, say e.

(ii) Among all rows i for which the minimum ratio test results in a tie, choose the row r for which the corresponding basic variable has the smallest index, j_r.

Theorem 5 *Under Bland's pivot selection rule, the Simplex Algorithm cannot cycle.*

Sketch of Proof

We think of the initial data as being expressed in a tableau of $m+1$ rows and $n+2$ columns (indexed from 0 to $n+1$) which we write as

$$\mathcal{A} = \left[\begin{array}{ccc} 1 & \mathbf{c}^T & 0 \\ 0 & A & \mathbf{b} \end{array}\right].$$

One row (the 0^{th}, $\mathcal{A}_{0\bullet}$) and column (the 0^{th}, $\mathcal{A}_{\bullet 0}$) pertain the variable x_0 we wish to optimize. The column indexed by $n+1$ is the right-hand side of the system of equations (augmented by an equation for the objective function).

Let $\bar{\mathcal{A}}$ denote the first $n+1$ columns of $\mathcal{A}$, i.e., with column $\mathcal{A}_{\bullet n+1}$ deleted. Analogous notations will be used for pivotal transforms of $\bar{\mathcal{A}}$.

Now if cycling occurs, there is a set τ of indices $j \in \{1, \ldots, n\}$ such that x_j becomes basic during cycling. Clearly τ has only a finite number of elements, so it has a largest element which we denote by q. Let $\mathcal{A}'$ denote the tableau that first specifies q as the pivot column. This means that x_q becomes a basic variable in the next tableau.

Let $\mathbf{y} = (y_0, y_1, \ldots, y_n) = \bar{\mathcal{A}}'_{0\bullet}$. By virtue of the definition of q and the rule that results in the choice of

q, we have

$$y_0 = 1, \quad y_j \geq 0 \quad 1 \leq j < q, \quad y_q < 0. \tag{1}$$

Note that the $(n+1)$-vector y belongs to the row space of $\bar{A}$. Now x_q must also leave the basis at some tableau $\mathcal{A}''$. Let $x_q = x_{j_r}$, and let t denote the pivot column when x_q becomes nonbasic. Define the $(n+1)$-vector $\mathbf{v} = (v_0, v_1, \ldots, v_n)$ as follows:

$$v_{j_i} = \bar{a}''_{it} \quad i = 0, 1, \ldots, m, \quad v_t = -1, \quad v_j = 0 \quad \text{else.} \tag{2}$$

Note that $v_0 = v_{j_0} = \bar{a}''_{0t} < 0$, $v_q = \bar{a}''_{rt} > 0$, and $\mathbf{v}$ is in the null space of $\bar{A}$. Thus, $\mathbf{y} \cdot \mathbf{v} = 0$, and by construction $y_0 v_0 < 0$. Hence $y_j v_j > 0$ for some $j \geq 1$. Since $y_j \neq 0$, x_j must be nonbasic in $\mathcal{A}'$; since $v_j \neq 0$, then either x_j is basic in $\mathcal{A}''$ or else $j = t$. Accordingly, $j \in \tau$, and hence $j \leq q$. By the construction again, $y_q < 0 < v_q$ which implies that $y_q v_q < 0$ hence $j < q$.

Furthermore, (1) implies that $y_j > 0$, so $v_j > 0$. Next we observe that $v_t = -1$ implies $j \neq t$. All these lead to the conclusion that x_j is currently basic in $\mathcal{A}''$. Let $j = j_p$ for some p. Then $v_j = \bar{a}''_{pt} > 0$.

Note that during the cycling the right-hand-side vector $\bar{\mathbf{b}}$ does not change and the values of all variables in τ are fixed at 0. This implies $\bar{b}''_p = 0$. We have established that $j = j_p, \ \bar{a}''_{pt} > 0$ and $\bar{b}''_p = 0$. But this contradicts the assumption that x_q is removed from the basic set corresponding to tableau $\mathcal{A}''$, since $j < q$ and by Bland's rule j should be removed. This means that cycling cannot occur when Bland's Rule is applied.

Remark. This elegant degeneracy resolution rule has the drawback that it may result in pivot choices that do not *significantly* improve the objective function value. It may also force the selection of dangerously small pivot elements.