More Simplex Pivoting Rules and Sensitivity Analyses

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http://www.stanford.edu/˜yyye

(LY Chapters 3 and 5.2)

How Good is the Greedy Pivoting Rule

Very good on average, but what about in the worse case ...?

When the simplex method is used to solve a linear program (LP), the number of iterations to solve the problem starting from a basic feasible solution (BFS) 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.

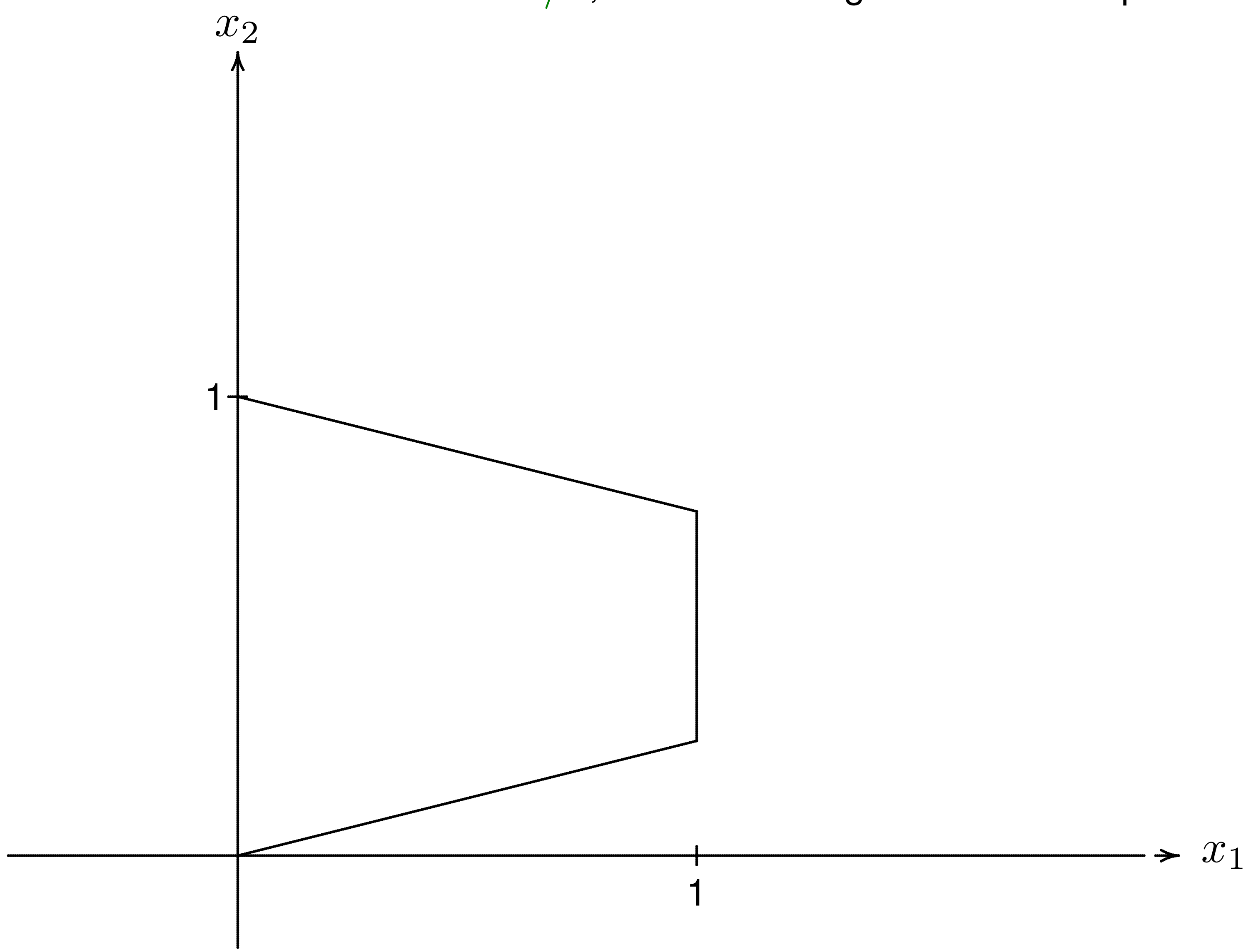
Klee and Minty Example

Consider

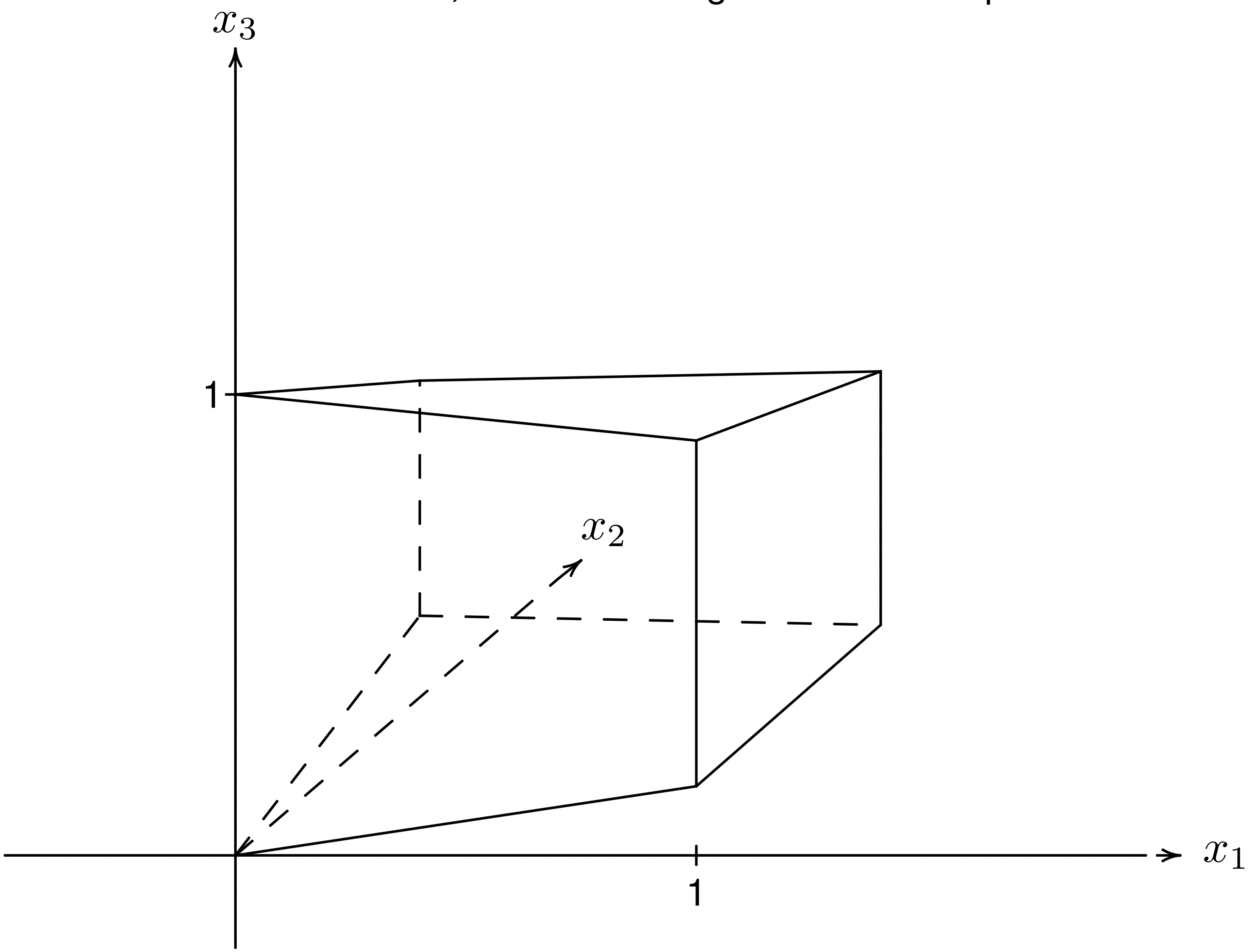
$$\begin{array}{lll} \max & x_n & \\ \text{subject to} & x_1 \geq 0 & \\ & x_1 \leq 1 & \\ & x_j \geq \epsilon\, x_{j-1} & j = 2, \ldots, n \\ & x_j \leq 1 - \epsilon\, x_{j-1} & j = 2, \ldots, n \end{array}$$

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.

In the case of $n = 2$ and $\epsilon = 1/4$, the feasible region of the example looks like



For the case where $n = 3$, the feasible region of the example looks like



The formulation above does not immediately reveal the standard form representation of the problem. Instead, we consider the following Klee-Minty example:

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

The problem above[a] is easily cast as a LP 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]Note that there is no need to express this problem in terms of powers of 10. Using any constant $C > 1$ would yield the same result (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). By adding slack variables s_1, s_2 and s_3, we get a problem in standard form. Then the system has $m = 3$ equations and $n = 6$ nonnegative variables.

We obtain the following initial tableau. The bullets below the tableau indicate the columns that are basic.

T^0

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	1
1	100	10	1	0	0	0	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
				•	•	•	

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 0, 0, 1, 100, 10000)$$

T^1

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	1
1	0	10	1	–100	0	0	–100
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
	●				●	●	

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (1, 0, 0, 0, 80, 9800)$$

T^2

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
	•	•				•	

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (1, 80, 0, 0, 0, 8200)$$

T^3

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
		●		●		●	

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 100, 0, 1, 0, 8000)$$

T^4

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
		•	•	•			

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 100, 8000, 1, 0, 0)$$

T^5

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
	•	•	•				

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (1, 80, 8200, 0, 0, 0)$$

T^6

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
	●		●		●		

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (1, 0, 9800, 0, 80, 0)$$

T^7

$-z$	x_1	x_2	x_3	s_1	s_2	s_3	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
			•	•	•		

So we conclude

$$(x_1, x_2, x_3, s_1, s_2, s_3) = (0, 0, 10000, 1, 100, 0)$$

is an optimal solution and that the corresponding 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 LP (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.

An Interesting Connection

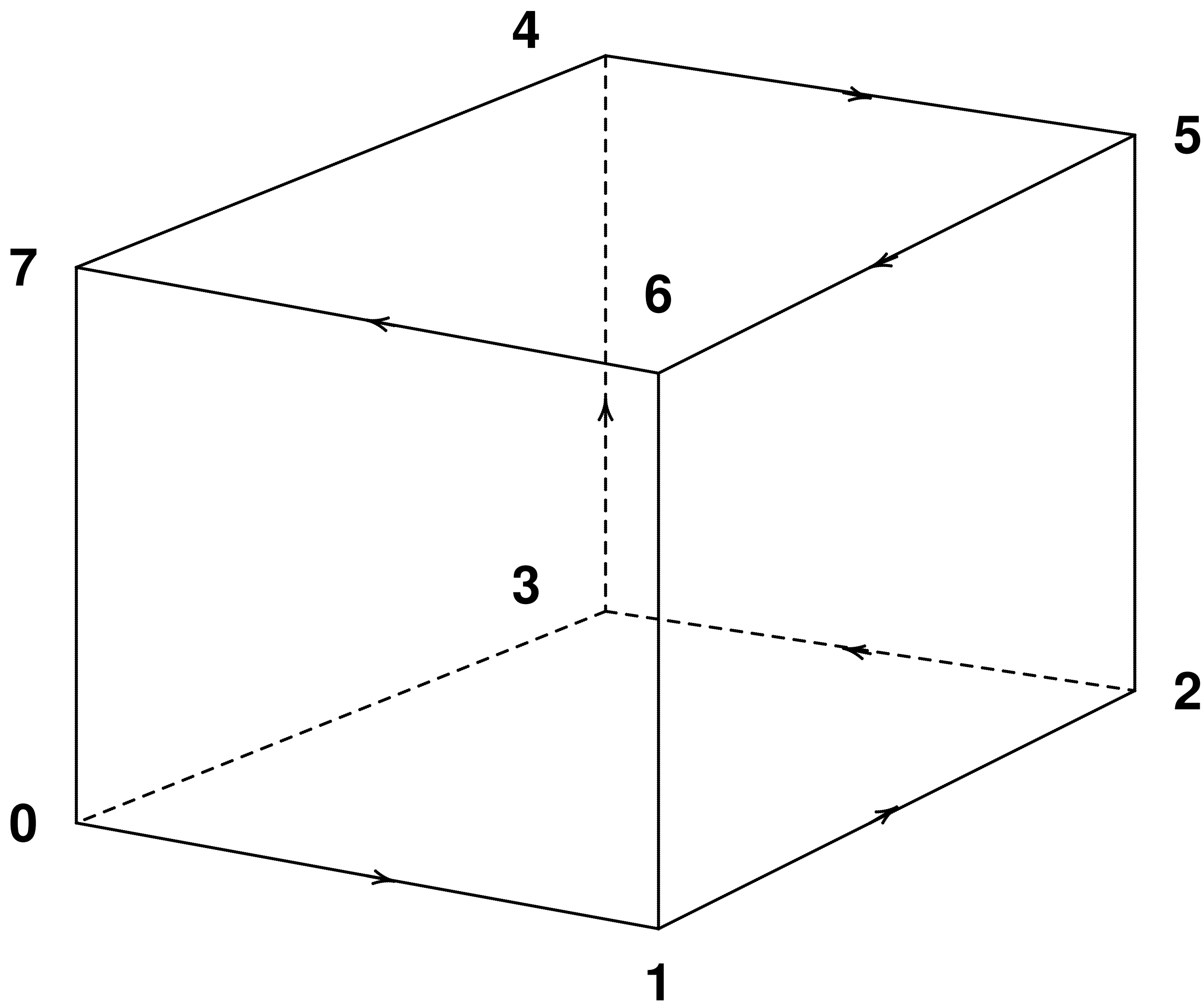
Consider the eight vectors $v^k = (v_1^k, v_2^k, v_3^k)$ for $k = 0, 1, \ldots, 7$ where

$$v_j^k := \begin{cases} 1 & \text{if } x_j \text{ is basic in tableau } k \\ 0 & \text{otherwise} \end{cases}$$

Looking at the previous eight tableaus $\mathrm{T}^0, \mathrm{T}^1, \ldots, \mathrm{T}^7$, we see that

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

Now suppose we regard these vectors as the coordinates of the vertices of the 3-cube $[\,0,1]^3$.



4
5
7
6
3
2
0
1

The figure above illustrates the fact that the sequence of vectors v^k corresponds to a path on the edges of the 3-cube. The path visits each vertex of the cube once and only once. Such a path is said to be Hamiltonian.

There is an amusing recreational literature that connects Hamiltonian path with certain puzzles. See Martin Gardner, "Mathematical games, the curious properties of the Gray code and how it can be used to solve puzzles," *Scientific American* 227 (August 1972) pp. 106-109. See also, S.N. Afriat, *The Ring of Linked Rings*, London: Duckworth, 1982.

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

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

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 #	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 LP. Among these are:

1. Perturbation of the right-hand side (RHS).
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 1 *Under Bland's pivot selection rule, the Simplex Algorithm cannot cycle.*

Sketch of Proof

Let initial tableau (we omit the RHS vector $\mathbf{b}$ here)

$$\mathcal{A} = \begin{bmatrix} 1 & \mathbf{c}^T \\ 0 & A \end{bmatrix},$$

where the column index from 0 to n and row index from 0 to m. 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. Also 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.

Let

$$\mathcal{A}' = \begin{bmatrix} 1 & (\mathbf{r}')^T \\ 0 & \bar{A}' \end{bmatrix}.$$

Denote the tableau that first specifies q as the pivot column, which means that x_q is the entering variable at $\mathcal{A}'$.

Let $\mathbf{y} = (1; \mathbf{r}')$. By virtue of the definition of q and the rule that results in the choice of q, we have

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

Note that the $(n+1)$-vector $\mathbf{y}$ belongs to the row space of $\mathcal{A}$ or $\mathcal{A}'$ or any subsequent tableau.

Now x_q must also leave the basis, say immediately after some tableau

$$\mathcal{A}'' = \begin{bmatrix} 1 & (\mathbf{r}'')^T \\ 0 & \bar{A}'' \end{bmatrix}$$

where basic variable index set $B'' = (j_1, j_2, ..., j_m)$ with $q = j_r$. Let t denote the entering variable to replace x_q. We define another $(n+1)$-vector

$\mathbf{v} = (v_0, v_1, \ldots, v_n)$ as follows:

$$v_0 = r''_t < 0, \quad \mathbf{v}_{B''} = \bar{A}''_{\cdot t}, \quad v_t = -1, \quad v_j = 0 \quad \text{else.}$$

Note that $v_q = \bar{A}''_{rt} > 0$ since x_q is the outgoing variable. Note that $\mathcal{A}''\mathbf{v} = \mathbf{0}$ so that it is also in the null-space of $\mathcal{A}'$, which implies $\mathbf{y} \cdot \mathbf{v} = 0$. By construction $y_0 v_0 = v_0 < 0$ so that $y_j v_j > 0$ for some $j \geq 1$.

Since $y_j \neq 0$, x_j must be nonbasic at $\mathcal{A}'$; since $v_j \neq 0$, x_j must be a basic variable at $\mathcal{A}''$ or $j = t$. Accordingly, $j \in \tau$, and hence $j \leq q$. But by construction again, $y_q < 0 < v_q$ which implies that $y_q v_q < 0$ so that $j \neq q$.

Furthermore, (1) implies that $y_j > 0$, so $v_j > 0$. Thus, $j \neq t$ since $v_t = -1$ from (2). Let $j = j_p$ for some p. Then $\bar{A}''_{pt} = v_j > 0$ and $\bar{\mathbf{b}}''_p = 0$.

But these contradict the assumption that x_q is outgoing at $\mathcal{A}''$, since $j < q$ and by Bland's rule j should be the outgoing variable. This means that cycling cannot occur when Bland's Rule is applied.

Optimal-Value Function of Data under Nondegenerary

Consider the following convex function:

$$\begin{array}{rll} OV(\mathbf{b}) = & \text{minimize} & \mathbf{c}^T\mathbf{x} \\ & \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & & \mathbf{x} \geq \mathbf{0}. \end{array}$$

$$\nabla OV(\mathbf{b}) = \mathbf{y}^*.$$

And also consider the following concave function:

$$\begin{array}{rll} OV(\mathbf{c}) = & \text{minimize} & \mathbf{c}^T\mathbf{x} \\ & \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & & \mathbf{x} \geq \mathbf{0}. \end{array}$$

$$\nabla OV(\mathbf{c}) = \mathbf{x}^*.$$

Sensitivity Analyses: Parametric LP

The the RHS vector becomes $\mathbf{b} + \lambda\mathbf{d}$ or the objective coefficient vector becomes $\mathbf{c} + \lambda\mathbf{g}$, where the parameter λ belongs to an interval.

Denote this problem by LP(λ):

$$\begin{array}{lll} \text{LP}(\lambda) & \text{minimize} & (\mathbf{c} + \lambda\mathbf{g})^{\mathbf{T}}\mathbf{x} \\ & \text{subject to} & A\mathbf{x} = \mathbf{b} + \lambda\mathbf{d}, \\ & & \mathbf{x} \geq \mathbf{0}. \end{array}$$

Geometrical Observations

1. If $\mathbf{b}$ is replaced by $\mathbf{b} + \lambda\mathbf{d}$, as λ varying the point $\mathbf{b} + \lambda\mathbf{d}$ moves away from $\mathbf{b}$ in the direction $\mathbf{d}$ (depending on the sign of λ). This raises the question of whether or not the LP problem remains (primal) feasible at the current basis.

2. We know that for the function $\mathbf{c}^T\mathbf{x}$, the vector $\mathbf{c}$ denotes the direction of steepest ascent. Thus, parameterizing the cost function according to the rule $\mathbf{c} + \lambda\mathbf{g}$ changes the gradient/slope, the normal direction of the objective hyperplane. This raises the question of whether or not the LP problem remains (dual) feasible at the current basis.

Getting Started

Let us consider λ around 0.

A key question in these parametric problems is: how much can the parameter λ be changed before the current optimal basic solution with basis set B of LP(0) is lost?

Theorem 2 *The optmal basis set B of LP(0) remains optimal for LP(λ) if and only if*

$$A_B^{-1}(\mathbf{b}+\lambda\mathbf{d}) \geq \mathbf{0} \quad \textit{and} \quad (\mathbf{c}+\lambda\mathbf{g}) - \mathbf{A^T}(\mathbf{A_B^T})^{-1}(\mathbf{c}+\lambda\mathbf{g})_{\mathbf{B}} \geq \mathbf{0}.$$

This will establish an interval on λ in which the optmal basis of LP(0) remains optimal.

Ceteris Paribus Analysis: RHS

The problem before us is to find (for each $i = 1, \ldots, m$) the range of values of the scalar λ for which the basis A_B remains optimal for the new RHS $\mathbf{b} + \lambda \mathbf{e}_i$, where $\mathbf{e}_i$ is the vector of all zero except 1 in the ith position.

A_B remains optimal if

$$\mathbf{0} \le A_B^{-1}(\mathbf{b} + \lambda \mathbf{e}_i) = \bar{\mathbf{b}} + \lambda(A_B^{-1}\mathbf{e}_i),$$

where $A_B^{-1}\mathbf{e}_i$ is simply the i-th column of A_B^{-1}.

Then the new optimal objective value is changed from the old one by $\lambda \cdot y_i^*$ where $\mathbf{y}^*$ is the optimal shadow price vector of LP(0):

$$\mathbf{c}_B^T A_B^{-1}(\mathbf{b}+\lambda\mathbf{e}_i) = (\mathbf{y}^*)^T(\mathbf{b}+\lambda\mathbf{e}_i) = (\mathbf{y}^*)^T\mathbf{b}+\lambda\cdot(\mathbf{y}^*)^T\mathbf{e}_i = (\mathbf{y}^*)^T\mathbf{b}+\lambda\cdot y_i^*.$$

Ceteris Paribus Analysis: Objective Coeffs of Nonbasic Variables

The problem before us is to find the range of values of the scalar λ for which the basis B remains optimal for the new $\mathbf{c} + \lambda \mathbf{e}_j$ where $j \in N$. A_B remains optimal if

$$(\mathbf{c} + \lambda \mathbf{e}_j)_N - A_N^T (A_B^T)^{-1} \mathbf{c}_B = \mathbf{r}_N + (\lambda \mathbf{e}_j)_N \geq \mathbf{0}.$$

Thus, as long as $\lambda \geq -r_j$ (the current reduced cost value of $j$the variable), then the optimal primal and dual solution remain unchanged.

Ceteris Paribus Analysis: Objective Coeffs of Basic Variables

The problem before us is to find the range of values of the scalar λ for which the basis B remains optimal for the new $\mathbf{c} + \lambda \mathbf{e}_j$, where $j \in B$. A_B remains optimal if

$$\mathbf{c}_N - A_N^T (A_B^T)^{-1} (\mathbf{c} + \lambda \mathbf{e}_j)_B = \mathbf{r}_N - \lambda \bar{A}_N^T (\mathbf{e}_j)_B \geq \mathbf{0}.$$

where $\bar{A}_N^T (\mathbf{e}_j)_B$ is simply the row of $\bar{A}$ corresponding to basic variable j excluding basic columns.

Then the new optimal objective value is changed from the old one by $\lambda \cdot x_j^*$ (where $\mathbf{x}^*$ is the optimal solution of LP(0)):

$$\begin{aligned}(\mathbf{c} + \lambda \mathbf{e}_j)_B^T A_B^{-1} \mathbf{b} = (\mathbf{c} + \lambda \mathbf{e}_j)_B^T \mathbf{x}_B^* = (\mathbf{c} + \lambda \mathbf{e}_j)^T \mathbf{x}^* = \mathbf{c}_B^T \mathbf{x}_B^* + \lambda \mathbf{e}_j^T \mathbf{x}^* \\ = \mathbf{c}_B^T \mathbf{x}_B^* + \lambda \cdot x_j^*.\end{aligned}$$