Linear Programming

Lecture 7: Does the Simplex Algorithm Work?

What Can Go Wrong with Simplex Algorithm?

Initialization:

What Can Go Wrong with Simplex Algorithm?

Initialization: The simplex algorithm pivots between feasible dictionaries (equivalently, feasible tableaus). The pivoting process moves us from one BFS to another BFS having a greater objective value.

What Can Go Wrong with Simplex Algorithm?

Initialization: The simplex algorithm pivots between feasible dictionaries (equivalently, feasible tableaus). The pivoting process moves us from one BFS to another BFS having a greater objective value.

Hence, in order to pivot, we need an initial feasible dictionary.

What Can Go Wrong with Simplex Algorithm?

Initialization: The simplex algorithm pivots between feasible dictionaries (equivalently, feasible tableaus). The pivoting process moves us from one BFS to another BFS having a greater objective value.

Hence, in order to pivot, we need an initial feasible dictionary.

How do we obtain the first feasible dictionary?

What Can Go Wrong with Simplex Algorithm?

Iteration:

What Can Go Wrong with Simplex Algorithm?

Iteration: Can we always choose variables to enter and leave the basis in an unambiguous way?

What Can Go Wrong with Simplex Algorithm?

Iteration: Can we always choose variables to enter and leave the basis in an unambiguous way?

Can there be multiple choices or no choice?

What Can Go Wrong with Simplex Algorithm?

Iteration: Can we always choose variables to enter and leave the basis in an unambiguous way?

Can there be multiple choices or no choice?

Are there ambiguities in the choice of these variables, and if so, can they be satisfactorily?

What Can Go Wrong with Simplex Algorithm?

Termination:

What Can Go Wrong with Simplex Algorithm?

Termination: Does the simplex algorithm terminate after a finite number of pivots?

What Can Go Wrong with Simplex Algorithm?

Termination: Does the simplex algorithm terminate after a finite number of pivots?

Does it terminate at a solution when a solution exists?

What Can Go Wrong with Simplex Algorithm?

Termination: Does the simplex algorithm terminate after a finite number of pivots?

Does it terminate at a solution when a solution exists?

Does it terminate for unbounded problems?

What Can Go Wrong with Simplex Algorithm?

Termination: Does the simplex algorithm terminate after a finite number of pivots?

Does it terminate at a solution when a solution exists?

Does it terminate for unbounded problems?

Can it stall, or can it go on pivoting forever without ever *solving* the problem?

Choosing the Entering Variable

Assume we are given a feasible dictionary:

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , & \end{array} \qquad (D_B)$$

where $\widehat{b}_i \geq 0, \ i \in B$.

Choosing the Entering Variable

Assume we are given a feasible dictionary:

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\ z &= \widehat{z} + \sum_{j\in N} \widehat{c}_j x_j \;, \end{aligned} \qquad (D_B)$$

where $\widehat{b}_i \geq 0,\; i \in B$.

Entering Variable:

Choosing the Entering Variable

Assume we are given a feasible dictionary:

$$
\left.\begin{array}{rcll}
x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\
z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , &
\end{array}\right. \qquad (D_B)
$$

where $\widehat{b}_i \geq 0, \ i \in B$.

Entering Variable:
A nonbasic variable x_{j_0}, $j_0 \in N$ can enter the basis if $\widehat{c}_{j_0} > 0$.

Choosing the Entering Variable

Assume we are given a feasible dictionary:

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \quad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned} \qquad (D_B)$$

where $\widehat{b}_i \geq 0, \ i \in B$.

Entering Variable:
A nonbasic variable x_{j_0}, $j_0 \in N$ can enter the basis if $\widehat{c}_{j_0} > 0$.

There may be many such nonbasic variables, but all of them have the potential to increase the value of the objective variable z.

Choosing the Leaving Variable

$$
\begin{aligned}
x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\
z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ ,
\end{aligned}
$$

Suppose x_{j_0}, $j_0 \in N$ is the entering variable.

Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Suppose x_{j_0}, $j_0 \in N$ is the entering variable.

The leaving variable is that basic variable whose non-negativity places the greatest restriction on increasing the value of the entering variable x_{j_0}.

Choosing the Leaving Variable

$$
\begin{aligned}
x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \quad i \in B \\
z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ ,
\end{aligned}
$$

Suppose x_{j_0}, $j_0 \in N$ is the entering variable.

The leaving variable is that basic variable whose non-negativity places the greatest restriction on increasing the value of the entering variable x_{j_0}.

The leaving variable attains the minimum value among the ratios

$$
\frac{\widehat{b}_i}{\widehat{a}_{ij_0}} \quad \text{for} \quad \widehat{a}_{ij_0} > 0 \qquad i \in B.
$$

Choosing the Leaving Variable

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , & \end{array}$$

Suppose x_{j_0}, $j_0 \in N$ is the entering variable.

The leaving variable is that basic variable whose non-negativity places the greatest restriction on increasing the value of the entering variable x_{j_0}.

The leaving variable attains the minimum value among the ratios

$$\frac{\widehat{b}_i}{\widehat{a}_{ij_0}} \quad \text{for} \quad \widehat{a}_{ij_0} > 0 \qquad i \in B.$$

If x_{i_0}, $i_0 \in B$ is the leaving variable, then

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B,\ \widehat{a}_{ij_0} > 0 \right\}.$$

Hazards in Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Hazards in Choosing the Leaving Variable

$$
\begin{aligned}
x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\
z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ ,
\end{aligned}
$$

Two potential problems:

Hazards in Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

Hazards in Choosing the Leaving Variable

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , & \end{array}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \ \widehat{a}_{ij_0} > 0 \right\}.$$

Hazards in Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B,\ \widehat{a}_{ij_0} > 0 \right\}.$$

If (i) occurs, then we can increase the value of the entering variable x_{j_0}, as much as we want without violating feasibility.

Hazards in Choosing the Leaving Variable

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , & \end{array}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \ \widehat{a}_{ij_0} > 0 \right\}.$$

If (i) occurs, then we can increase the value of the entering variable x_{j_0}, as much as we want without violating feasibility.
That is, we can increase the value of z as much as we want.

Hazards in Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \ \widehat{a}_{ij_0} > 0 \right\}.$$

If (i) occurs, then we can increase the value of the entering variable x_{j_0}, as much as we want without violating feasibility.
That is, we can increase the value of z as much as we want.
Hence the LP is unbounded.

Unbounded LPs

Fact: If there exists $j_0 \in N$ in the dictionary D_B for which $\widehat{c}_{j_0} > 0$ and $\widehat{a}_{ij_0} \leq 0$ for all $i \in B$, then the LP

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b,\ 0 \leq x \end{array}$$

is unbounded, i.e., the optimal value is $+\infty$.

Unbounded LPs

$$
\begin{array}{lrrrcl}
\text{maximize} & x_1 & +x_2 & +x_3 & & \\
\text{subject to} & 3x_1 & +x_2 & -2x_3 & \leq & 5 \\
 & 4x_1 & +3x_2 & & \leq & 7 \\
 & \multicolumn{3}{l}{0 \leq x_1, x_2, x_3} & &
\end{array}
$$

Unbounded LPs

$$\begin{array}{lrrrcr} \text{maximize} & x_1 & +x_2 & +x_3 & & \\ \text{subject to} & 3x_1 & +x_2 & -2x_3 & \leq & 5 \\ & 4x_1 & +3x_2 & & \leq & 7 \\ & \multicolumn{3}{l}{0 \leq x_1, x_2, x_3} & & \end{array}$$

$$\left[\begin{array}{cccccc|c} 3 & 1 & -2 & 1 & 0 & 0 & 5 \\ 4 & 3 & 0 & 0 & 1 & 0 & 7 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Hazards in Choosing the Leaving Variable

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad & i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \ \widehat{a}_{ij_0} > 0 \right\}.$$

Hazards in Choosing the Leaving Variable

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , & \end{array}$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B,\ \widehat{a}_{ij_0} > 0 \right\}.$$

(ii) is **VERY BAD** for the simplex algorithm!

Hazards in Choosing the Leaving Variable

$$
\begin{aligned}
x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\
z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ ,
\end{aligned}
$$

Two potential problems:

(i) There is no $i \in B$ for which $\widehat{a}_{ij_0} > 0$.

(ii) There is more than one $i_0 \in B$ for which

$$
\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \ \widehat{a}_{ij_0} > 0 \right\} .
$$

(ii) is **VERY BAD** for the simplex algorithm!
We show why by example.

Example

$$\begin{array}{ll} \text{maximize} & 2x_1 - x_2 + 8x_3 \\ \text{subject to} & 2x_1 - 4x_2 + 6x_3 \leq 3 \\ & -x_1 + 3x_2 + 4x_3 \leq 2 \\ & 2x_3 \leq 1 \\ & 0 \leq x_1, x_2, x_3 \end{array}$$

Example

$$\begin{array}{lr} \text{maximize} & 2x_1 - x_2 + 8x_3 \\ \text{subject to} & 2x_1 - 4x_2 + 6x_3 \leq 3 \\ & -x_1 + 3x_2 + 4x_3 \leq 2 \\ & 2x_3 \leq 1 \\ & 0 \leq x_1, x_2, x_3 \end{array}$$

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{array}{|cccccc|c} 2 & -4 & 6 & 1 & 0 & 0 & 3 \\ -1 & 3 & 4 & 0 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 & 0 & 1 & 1 \\ \hline 2 & -1 & 8 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$z = 0$$

Note that any one of these rows could serve as the pivot row!

Example

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad z = 0$$

2	−4	6	1	0	0	3
−1	3	4	0	1	0	2
0	0	2	0	0	1	1
2	−1	8	0	0	0	0

Note that any one of these rows could serve as the pivot row!

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad z = 4$$

2	−4	0	1	0	−3	0
−1	3	0	0	1	−2	0
0	0	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$
2	−1	0	0	0	−4	−4

Note that by pivoting on this tableau we do not change the objective value

Example

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad z = 4$$

$$\begin{array}{|cccccc|c}
2 & -4 & 0 & 1 & 0 & -3 & 0 \\
-1 & 3 & 0 & 0 & 1 & -2 & 0 \\
0 & 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
\hline
2 & -1 & 0 & 0 & 0 & -4 & -4 \\
\hline
\end{array}$$

Note that by pivoting on this tableau we do not change the objective value

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad z = 4$$

$$\begin{array}{|cccccc|c}
1 & -2 & 0 & \frac{1}{2} & 0 & -\frac{3}{2} & 0 \\
0 & 1 & 0 & \frac{1}{2} & 1 & -\frac{7}{2} & 0 \\
0 & 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
\hline
0 & 3 & 0 & -1 & 0 & -1 & -4 \\
\hline
\end{array}$$

Note that we have not changed the point identified by this tableau

Example

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad z = 4$$

$$\begin{array}{|cccccc|c}
1 & -2 & 0 & \frac{1}{2} & 0 & -\frac{3}{2} & 0 \\
0 & 1 & 0 & \frac{1}{2} & 1 & -\frac{7}{2} & 0 \\
0 & 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
\hline
0 & 3 & 0 & -1 & 0 & -1 & -4 \\
\hline
\end{array}$$

Note that we have not changed the point identified by this tableau

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad z = 4$$

$$\begin{array}{|cccccc|c}
1 & 0 & 0 & \frac{3}{2} & 2 & -\frac{17}{2} & 0 \\
0 & 1 & 0 & \frac{1}{2} & 1 & -\frac{7}{2} & 0 \\
0 & 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
\hline
0 & 0 & 0 & -\frac{5}{2} & -3 & \frac{19}{2} & -4 \\
\hline
\end{array}$$

Again no change.

Example

$$x = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix} \qquad \begin{array}{|cccccc|c} 1 & 0 & 0 & \frac{3}{2} & 2 & -\frac{17}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 1 & -\frac{7}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \hline 0 & 0 & 0 & -\frac{5}{2} & -3 & \frac{19}{2} & -4 \\ \hline \end{array}$$

$z = 4$

Again no change.

$$x = \begin{pmatrix} \frac{17}{2} \\ \frac{7}{2} \\ 0 \end{pmatrix} \qquad \begin{array}{|cccccc|c} 1 & 0 & 17 & \frac{3}{2} & 2 & 0 & \frac{17}{2} \\ 0 & 1 & 7 & \frac{1}{2} & 1 & 0 & \frac{7}{2} \\ 0 & 0 & 2 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & \text{-}19 & -\frac{5}{2} & -3 & 0 & -\frac{27}{2} \\ \hline \end{array}$$

$z = \frac{27}{2}$

Finally, we break out to optimality.

Degeneracy

Observations:

Degeneracy

Observations:

(1) If on a given pivot, there is more than one choice of variable to leave the basis, then the subsequent tableau will set one or more of the basic variables equal to zero in the associated basic feasible solution.

Degeneracy

Observations:

(1) If on a given pivot, there is more than one choice of variable to leave the basis, then the subsequent tableau will set one or more of the basic variables equal to zero in the associated basic feasible solution.
A dictionary in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate dictionary".

Degeneracy

Observations:

(1) If on a given pivot, there is more than one choice of variable to leave the basis, then the subsequent tableau will set one or more of the basic variables equal to zero in the associated basic feasible solution.
A dictionary in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate dictionary".
Correspondingly, a tableau in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate tableau".

Degeneracy

Observations:

(1) If on a given pivot, there is more than one choice of variable to leave the basis, then the subsequent tableau will set one or more of the basic variables equal to zero in the associated basic feasible solution.
A dictionary in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate dictionary".
Correspondingly, a tableau in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate tableau".

(2) It is possible that a pivot on a degenerate dictionary (or tableau) does not change the associated basic feasible solution and the value of the objective variable z.

Degeneracy

Observations:

(1) If on a given pivot, there is more than one choice of variable to leave the basis, then the subsequent tableau will set one or more of the basic variables equal to zero in the associated basic feasible solution.
A dictionary in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate dictionary".
Correspondingly, a tableau in which one or more of the basic variables is set to zero in the associated basic feasible solution is called a "degenerate tableau".

(2) It is possible that a pivot on a degenerate dictionary (or tableau) does not change the associated basic feasible solution and the value of the objective variable z.
Such a pivot is called a "degenerate pivot".

Degeneracy

Observation (2) is particularly troublesome since it opens the door to the possibility of an infinite sequence of degenerate pivots never terminating with optimality.

Degeneracy

Observation (2) is particularly troublesome since it opens the door to the possibility of an infinite sequence of degenerate pivots never terminating with optimality.

Unfortunately, this *can* occur leading to the failure of the method. An example of the phenomenon is given in the text.

Degeneracy

Observation (2) is particularly troublesome since it opens the door to the possibility of an infinite sequence of degenerate pivots never terminating with optimality.

Unfortunately, this *can* occur leading to the failure of the method. An example of the phenomenon is given in the text.

Our goal is to understand how such a pathological situation can occur and then to devise methods to overcome the problem.

Cycling

Assume the algorithm is operating with iron clad pivoting rules.

Cycling

Assume the algorithm is operating with iron clad pivoting rules. That is, non-optimal feasible dictionaries have unique pivots.

Cycling

Assume the algorithm is operating with iron clad pivoting rules.
That is, non-optimal feasible dictionaries have unique pivots.

Example:

Cycling

Assume the algorithm is operating with iron clad pivoting rules. That is, non-optimal feasible dictionaries have unique pivots.

Example:

$$\begin{aligned} x_i &= \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j \qquad\qquad i \in B \\ z &= \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ , \end{aligned} \tag{D_B}$$

Largest-Coefficient Largest-Subscript Rule

Cycling

Largest-Coefficient Largest-Subscript Rule

- Choice of Entering Variable: Among all those variables x_j with $j \in N$ such that $\widehat{c}_j = \max\{\widehat{c}_k : k \in N\} > 0$ choose x_{j_0} so that j_0 is largest.

Cycling

Largest-Coefficient Largest-Subscript Rule

- Choice of Entering Variable: Among all those variables x_j with $j \in N$ such that $\widehat{c}_j = \max\{\widehat{c}_k : k \in N\} > 0$ choose x_{j_0} so that j_0 is largest.

- Choice of Leaving Variable: Among all those variables x_i with $i \in B$ such that $\frac{\widehat{b}_i}{\widehat{a}_{ij_0}} = \min\left\{\frac{\widehat{b}_k}{\widehat{a}_{kj_0}} : k \in B, \widehat{a}_{kj_0} > 0\right\}$ choose x_{i_0} so that i_0 is largest.

How Many Possible Bases?

How many possible ways are there to choose a set of basic indices?

How Many Possible Bases?

How many possible ways are there to choose a set of basic indices?

Since every basis must contain m variables and there are only $n+m$ variables altogether, the total number of possible sets of basic indices equals the number of possible ways to choose m distinct elements from a collection of $n+m$ objects.

How Many Possible Bases?

How many possible ways are there to choose a set of basic indices?

Since every basis must contain m variables and there are only $n+m$ variables altogether, the total number of possible sets of basic indices equals the number of possible ways to choose m distinct elements from a collection of $n+m$ objects.

$$\binom{n+m}{m} = \frac{(n+m)!}{m!n!}.$$

Cycling

infinite pivot sequence $\Rightarrow$ infinite dictionary sequence.

Cycling

infinite pivot sequence $\Rightarrow$ infinite dictionary sequence.

At least one dictionary, say D_1, has a basis B appearing twice.

Cycling

infinite pivot sequence $\Rightarrow$ infinite dictionary sequence.

At least one dictionary, say D_1, has a basis B appearing twice.
Suppose B is also the basis for D_{N+1}.

Cycling

$$\text{infinite pivot sequence} \Rightarrow \text{infinite dictionary sequence.}$$

At least one dictionary, say D_1, has a basis B appearing twice.
Suppose B is also the basis for D_{N+1}.
Let

$$\dots D_1, \dots, D_N, D_{N+1}, D_{N+2}, D_{N+2}, \dots$$

be the sequence of pivots where D_1 and D_{N+1} have the same basis.

Cycling

$$\text{infinite pivot sequence} \Rightarrow \text{infinite dictionary sequence.}$$

At least one dictionary, say D_1, has a basis B appearing twice.
Suppose B is also the basis for D_{N+1}.
Let

$$\ldots D_1, \ldots, D_N, D_{N+1}, D_{N+2}, D_{N+2}, \ldots$$

be the sequence of pivots where D_1 and D_{N+1} have the same basis.
If each basis is associated with a unique dictionary, then $D_1 = D_{N+1}$.

Cycling

infinite pivot sequence $\Rightarrow$ infinite dictionary sequence.

At least one dictionary, say D_1, has a basis B appearing twice.
Suppose B is also the basis for D_{N+1}.
Let

$$\ldots D_1, \ldots, D_N, D_{N+1}, D_{N+2}, D_{N+2}, \ldots$$

be the sequence of pivots where D_1 and D_{N+1} have the same basis.
If each basis is associated with a unique dictionary, then $D_1 = D_{N+1}$. But then,

$$D_2 = D_{N+2},\ D_3 = D_{N+3},\ \ldots,\ D_1 = D_{2N},\ D_2 = D_{2N+2},\ \ldots$$

Cycling

infinite pivot sequence $\Rightarrow$ infinite dictionary sequence.

At least one dictionary, say D_1, has a basis B appearing twice.
Suppose B is also the basis for D_{N+1}.
Let

$$\ldots D_1, \ldots, D_N, D_{N+1}, D_{N+2}, D_{N+2}, \ldots$$

be the sequence of pivots where D_1 and D_{N+1} have the same basis.
If each basis is associated with a unique dictionary, then $D_1 = D_{N+1}$. But then,

$$D_2 = D_{N+2},\ D_3 = D_{N+3},\ \ldots,\ D_1 = D_{2N},\ D_2 = D_{2N+2},\ \ldots$$

That is, the same sequence of dictionaries appear over and over again. If this occurs we say that the sequence of dictionaries cycles.

The Basis-Dictionary Correspondence

Fact : Every basis uniquely determines its associated dictionary.

Corollary:
The simplex algorithm fails to terminate if and only if it cycles. The simplex algorithms can only cycle between degenerate dictionaries (or tableaus) with each dictionary (or tableau) in the cycle being associated with the same basic feasible solution and objective value.

Proof

We show each basis yields a unique dictionary.

Proof

We show each basis yields a unique dictionary.

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j, & \qquad i \in B \\ z & = & \widehat{z}_i + \sum_{j \in N} \widehat{c}_j x_j & \end{array} \qquad (D_1)$$

and

$$\begin{array}{rcll} x_i & = & b_i^* - \sum_{j \in N} a_{ij}^* x_j, & \qquad i \in B \\ z & = & z^* + \sum_{j \in N} c_j^* x_j & \end{array} \qquad (D_2)$$

Two dictionaries with the same basis B.

Proof

We show each basis yields a unique dictionary.

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j, & \qquad i \in B \\ z & = & \widehat{z}_i + \sum_{j\in N} \widehat{c}_j x_j & \end{array} \qquad (D_1)$$

and

$$\begin{array}{rcll} x_i & = & b_i^* - \sum_{j\in N} a_{ij}^* x_j, & \qquad i \in B \\ z & = & z^* + \sum_{j\in N} c_j^* x_j & \end{array} \qquad (D_2)$$

Two dictionaries with the same basis B.
Show all coefficients are identical.

Proof

$$
\begin{array}{rcll}
& & (D_1) & \\
x_i & = & \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j, & i \in B \\
z & = & \widehat{z}_i + \sum_{j\in N} \widehat{c}_j x_j &
\end{array}
\qquad
\begin{array}{rcll}
& & (D_2) & \\
x_i & = & b_i^* - \sum_{j\in N} a_{ij}^* x_j, & i \in B \\
z & = & z^* + \sum_{j\in N} c_j^* x_j &
\end{array}
$$

D_1 and D_2 have identical solution sets.

Proof

$$\begin{array}{rcll} & & (D_1) & \\ x_i & = & \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j, & i \in B \\ z & = & \widehat{z}_i + \sum_{j\in N} \widehat{c}_j x_j & \end{array} \qquad \begin{array}{rcll} & & (D_2) & \\ x_i & = & b_i^* - \sum_{j\in N} a_{ij}^* x_j, & i \in B \\ z & = & z^* + \sum_{j\in N} c_j^* x_j & \end{array}$$

D_1 and D_2 have identical solution sets.
Let $j_0 \in N$ and set $x_{j_0} = t$ and $x_j = 0$ for $j \in N$, $j \neq j_0$.
Then

$$\begin{array}{rcll} \widehat{b}_i - \widehat{a}_{ij_0} t & = & x_i = b_i^* - a_{ij_0}^* t & \quad \text{for } i \in B \\ \widehat{z} + \widehat{c}_{j_0} t & = & z = z^* + c_{j_0}^* t. & \end{array}$$

Setting $t = 0$, we have

$$\widehat{b}_i = b_i^* \quad i \in B \quad \text{and} \quad \widehat{z} = z^*.$$

Then, setting $t = 1$, we have

$$\widehat{a}_{ij_0} = a_{ij_0}^* \qquad \text{for } i \in B.$$

Proof

$$
\begin{array}{ccc}
& (D_1) & \\
x_i & = & \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j, \quad i \in B \\
z & = & \widehat{z}_i + \sum_{j\in N} \widehat{c}_j x_j
\end{array}
\qquad
\begin{array}{ccc}
& (D_2) & \\
x_i & = & b_i^* - \sum_{j\in N} a_{ij}^* x_j, \quad i \in B \\
z & = & z^* + \sum_{j\in N} c_j^* x_j
\end{array}
$$

D_1 and D_2 have identical solution sets.
Let $j_0 \in N$ and set $x_{j_0} = t$ and $x_j = 0$ for $j \in N$, $j \neq j_0$.
Then

$$
\begin{aligned}
\widehat{b}_i - \widehat{a}_{ij_0} t &= x_i = b_i^* - a_{ij_0}^* t \qquad \text{for } i \in B \\
\widehat{z} + \widehat{c}_{j_0} t &= z = z^* + c_{j_0}^* t.
\end{aligned}
$$

Setting $t = 0$, we have

$$\widehat{b}_i = b_i^* \quad i \in B \quad \text{and} \quad \widehat{z} = z^*.$$

Then, setting $t = 1$, we have

$$\widehat{a}_{ij_0} = a_{ij_0}^* \qquad \text{for } i \in B.$$

Repeating for all $j \in N$.

Degeneracy and Cycling

We have established that the simplex algorithm can only fail to terminate if it cycles, and that it can only cycle in the presence of degeneracy. In order to assure that the simplex algorithm successfully terminates we need to develop a pivoting rule that avoids cycling.

Degeneracy and Cycling

We have established that the simplex algorithm can only fail to terminate if it cycles, and that it can only cycle in the presence of degeneracy. In order to assure that the simplex algorithm successfully terminates we need to develop a pivoting rule that avoids cycling.

There are many anti-cycling pivoting rules. We present the smallest subscript rule, also known as Bland's Rule.

Bland's Rule

$$
\begin{array}{rcll}
x_i & = & \widehat{b}_i - \sum_{j\in N} \widehat{a}_{ij} x_j & \qquad i \in B \\
z & = & \widehat{z} + \sum_{j\in N} \widehat{c}_j x_j \ . &
\end{array} \qquad (D_B)
$$

Bland's Rule

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ . & \end{array} \qquad (D_B)$$

Choice of entering variable: x_{j_0} for $j_0 \in N$ is the entering variable if $\widehat{c}_{j_0} > 0$ and $j_0 \leq j$ whenever $\widehat{c}_j > 0$.

Bland's Rule

$$\begin{array}{rcll} x_i & = & \widehat{b}_i - \sum_{j \in N} \widehat{a}_{ij} x_j & \qquad i \in B \\ z & = & \widehat{z} + \sum_{j \in N} \widehat{c}_j x_j \ . & \end{array} \qquad (D_B)$$

Choice of entering variable: x_{j_0} for $j_0 \in N$ is the entering variable if $\widehat{c}_{j_0} > 0$ and $j_0 \leq j$ whenever $\widehat{c}_j > 0$.

Choice of leaving variable: x_{i_0} for $i_0 \in B$ is the leaving variable if

$$\frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \min \left\{ \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} : i \in B, \widehat{a}_{ij_0} > 0 \right\}$$

and

$$i_0 \leq i \ \text{ whenever } \ \frac{\widehat{b}_{i_0}}{\widehat{a}_{i_0 j_0}} = \frac{\widehat{b}_i}{\widehat{a}_{ij_0}} \quad i \in B.$$

Bland's Rule

Theorem: [R.G. Bland (1977)]
The simplex algorithm terminates as long as the choice of variable to enter or leave the basis is made according to the smallest subscript rule.

Bland's Rule

Theorem: [R.G. Bland (1977)]
The simplex algorithm terminates as long as the choice of variable to enter or leave the basis is made according to the smallest subscript rule.

We abbreviate the *Bland's rule* to **SSR** (smallest subscript rule).

Proof

Prove Bland's rule prevents cycling.

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0\ .$$

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0 \ .$$

Since all the pivots in the cycle are degenerate, the minimum ratio is zero in each pivot, and so each dictionary in the cycle $D_0, \ldots, D_N$ identifies the same BFS.

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0 \ .$$

Since all the pivots in the cycle are degenerate, the minimum ratio is zero in each pivot, and so each dictionary in the cycle $D_0, \ldots, D_N$ identifies the same BFS.

If a variable enters (leaves) the basis in one of the dictionaries $D_0, \ldots, D_N$, then it must depart (return) from (to) the basis in another of these dictionaries.

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0 \ .$$

Since all the pivots in the cycle are degenerate, the minimum ratio is zero in each pivot, and so each dictionary in the cycle $D_0, \ldots, D_N$ identifies the same BFS.

If a variable enters (leaves) the basis in one of the dictionaries $D_0, \ldots, D_N$, then it must depart (return) from (to) the basis in another of these dictionaries.

We call the variables that leave the basis in any of the dictionaries $D_0, \ldots, D_N$ *fickle* since they move in and out of the basis.

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0 \ .$$

Since all the pivots in the cycle are degenerate, the minimum ratio is zero in each pivot, and so each dictionary in the cycle $D_0, \ldots, D_N$ identifies the same BFS.

If a variable enters (leaves) the basis in one of the dictionaries $D_0, \ldots, D_N$, then it must depart (return) from (to) the basis in another of these dictionaries.

We call the variables that leave the basis in any of the dictionaries $D_0, \ldots, D_N$ *fickle* since they move in and out of the basis.

Denote the set of fickle variables by $\mathcal{F}$.

Proof

Prove Bland's rule prevents cycling.

Assume to the contrary that a cycle exists:

$$D_0, D_1, \ldots, D_N = D_0 \ .$$

Since all the pivots in the cycle are degenerate, the minimum ratio is zero in each pivot, and so each dictionary in the cycle $D_0, \ldots, D_N$ identifies the same BFS.

If a variable enters (leaves) the basis in one of the dictionaries $D_0, \ldots, D_N$, then it must depart (return) from (to) the basis in another of these dictionaries.

We call the variables that leave the basis in any of the dictionaries $D_0, \ldots, D_N$ *fickle* since they move in and out of the basis.

Denote the set of fickle variables by $\mathcal{F}$.

All fickle variables must take the value zero in the BFS associated with this cycle since they take the value zero when they are not in the basis.

Proof

Let ℓ be the largest subscript in $\mathcal{F}$, and let

$$D \qquad \begin{aligned} x_i &= b_i - \sum_{j \notin B} a_{ij} x_j \,, \qquad i \in B \\ z &= v + \sum_{j \notin B} c_j x_j \end{aligned}$$

be a dictionary in the cycle where x_ℓ is leaving the basis and let x_e denote the entering variable: x_ℓ leaves D and x_e enters.

Proof

Let ℓ be the largest subscript in $\mathcal{F}$, and let

$$D \qquad \begin{aligned} x_i &= b_i - \sum_{j \notin B} a_{ij} x_j \,, \qquad i \in B \\ z &= v + \sum_{j \notin B} c_j x_j \end{aligned}$$

be a dictionary in the cycle where x_ℓ is leaving the basis and let x_e denote the entering variable: x_ℓ leaves D and x_e enters.

Since x_e is also fickle, $e < \ell$ (ℓ is largest in $\mathcal{F}$), $c_e > 0$, and $a_{\ell e} > 0$.

Proof

Let ℓ be the largest subscript in $\mathcal{F}$, and let

$$D \qquad \begin{aligned} x_i &= b_i - \sum_{j \notin B} a_{ij} x_j \,, \qquad i \in B \\ z &= v + \sum_{j \notin B} c_j x_j \end{aligned}$$

be a dictionary in the cycle where x_ℓ is leaving the basis and let x_e denote the entering variable: x_ℓ leaves D and x_e enters.

Since x_e is also fickle, $e < \ell$ (ℓ is largest in $\mathcal{F}$), $c_e > 0$, and $a_{\ell e} > 0$.

Let D^* be a dictionary in the cycle with x_ℓ entering the basis.

Proof

Let ℓ be the largest subscript in $\mathcal{F}$, and let

$$
\begin{aligned}
D \qquad\qquad\qquad x_i &= b_i - \sum_{j \notin B} a_{ij} x_j \,, \qquad i \in B \\
z &= v + \sum_{j \notin B} c_j x_j
\end{aligned}
$$

be a dictionary in the cycle where x_ℓ is leaving the basis and let x_e denote the entering variable: x_ℓ leaves D and x_e enters.

Since x_e is also fickle, $e < \ell$ (ℓ is largest in $\mathcal{F}$), $c_e > 0$, and $a_{\ell e} > 0$.

Let D^* be a dictionary in the cycle with x_ℓ entering the basis.
Since each dictionary in the cycle identifies the same BFS, the objective value v stays constant throughout the cycle.

Proof

Since each dictionary in the cycle identifies the same BFS, the objective value v stays constant throughout the cycle.

Proof

Since each dictionary in the cycle identifies the same BFS, the objective value v stays constant throughout the cycle.

Write the objective row of D^* (x_ℓ entering) as

$$z = v + \sum_{j=1}^{m+n} c_j^* x_j,$$

where $c_j^* = 0$ if x_j is basic in D^*. Note, $c_j^* \leq 0 \ \ \forall\, j \in \mathcal{F} \setminus \{\ell\}$ and $c_\ell^* > 0$.

Proof

Since each dictionary in the cycle identifies the same BFS, the objective value v stays constant throughout the cycle.

Write the objective row of D^* (x_ℓ entering) as

$$z = v + \sum_{j=1}^{m+n} c_j^* x_j,$$

where $c_j^* = 0$ if x_j is basic in D^*. Note, $c_j^* \leq 0 \;\; \forall\, j \in \mathcal{F} \setminus \{\ell\}$ and $c_\ell^* > 0$. Since the solution sets to D and D^* coincide, the solution to D obtained by setting $x_e = t,\; x_j = 0 \; (j \notin B,\; j \neq e)$ giving

$$x_i = b_i - a_{ie} t \;\; (i \in B) \text{ and } z = v + c_e t$$

must satisfy D^*.

Proof

Since each dictionary in the cycle identifies the same BFS, the objective value v stays constant throughout the cycle.

Write the objective row of D^* (x_ℓ entering) as

$$z = v + \sum_{j=1}^{m+n} c_j^* x_j,$$

where $c_j^* = 0$ if x_j is basic in D^*. Note, $c_j^* \leq 0 \ \ \forall\, j \in \mathcal{F} \setminus \{\ell\}$ and $c_\ell^* > 0$. Since the solution sets to D and D^* coincide, the solution to D obtained by setting $x_e = t,\ x_j = 0\ (j \notin B,\ j \neq e)$ giving

$$x_i = b_i - a_{ie} t\ (i \in B) \text{ and } z = v + c_e t$$

must satisfy D^*.

Hence $\quad v + c_e t = v + c_e^* t + \sum_{i \in B} c_i^* (b_i - a_{ie} t) \qquad$ for all t.

Proof

$$v + c_e t = v + c_e^* t + \sum_{i \in B} c_i^* (b_i - a_{ie} t) \qquad \text{for all } t.$$

Proof

$$v + c_e t = v + c_e^* t + \sum_{i \in B} c_i^* (b_i - a_{ie} t) \qquad \text{for all } t.$$

Grouping terms gives

$$\left(c_e - c_e^* + \sum_{i \in B} c_i^* a_{ie} \right) t = \sum_{i \in B} c_i^* b_i \qquad \text{for all } t.$$

Proof

$$v + c_e t = v + c_e^* t + \sum_{i \in B} c_i^* (b_i - a_{ie} t) \qquad \text{for all } t.$$

Grouping terms gives

$$\left(c_e - c_e^* + \sum_{i \in B} c_i^* a_{ie} \right) t = \sum_{i \in B} c_i^* b_i \qquad \text{for all } t.$$

Since the right hand side is constant, it must be 0 as is the coefficient on the left:

$$c_e - c_e^* = - \sum_{i \in B} c_i^* a_{ie} \ .$$

Proof

$$c_e - c_e^* = -\sum_{i \in B} c_i^* a_{ie}$$

Proof

$$c_e - c_e^* = -\sum_{i \in B} c_i^* a_{ie}$$

$$x_e \text{ enters in } D \;\Rightarrow\; c_e > 0$$

Proof

$$c_e - c_e^* = -\sum_{i \in B} c_i^* a_{ie}$$

$$x_e \text{ enters in } D \;\Rightarrow\; c_e > 0$$

$$x_\ell \text{ enters in } D^* \text{ and } e < \ell \;\Rightarrow\; c_e^* \leq 0 \text{ by the } \textbf{SSR}$$

Proof

$$c_e - c_e^* = -\sum_{i \in B} c_i^* a_{ie}$$

$$x_e \text{ enters in } D \;\Rightarrow\; c_e > 0$$

$$x_\ell \text{ enters in } D^* \text{ and } e < \ell \;\Rightarrow\; c_e^* \leq 0 \text{ by the } \textbf{SSR}$$

Therefore, $c_e - c_e^* > 0$.

Proof

$$c_e - c_e^* = -\sum_{i \in B} c_i^* a_{ie}$$

$$x_e \text{ enters in } D \;\Rightarrow\; c_e > 0$$

$$x_\ell \text{ enters in } D^* \text{ and } e < \ell \;\Rightarrow\; c_e^* \leq 0 \text{ by the } \textbf{SSR}$$

Therefore, $c_e - c_e^* > 0$.

Consequently, $\sum_{i \in B} c_i^* a_{ie} < 0$, so for some $s \in B$,

$$c_s^* a_{se} < 0 \ .$$

Proof

$$\text{for some } s \in B, \quad c_s^* a_{se} < 0.$$

Proof

$$\text{for some } s \in B, \quad c_s^* a_{se} < 0.$$

Since $s \in B$, x_s is basic in D, and since $c_s^* \neq 0$, x_s is nonbasic in D^*, so $x_s \in \mathcal{F}$ which implies that $s \leq \ell$ (since ℓ is largest).

Proof

$$\text{for some } s \in B, \quad c_s^* a_{se} < 0.$$

Since $s \in B$, x_s is basic in D, and since $c_s^* \neq 0$, x_s is nonbasic in D^*, so $x_s \in \mathcal{F}$ which implies that $s \leq \ell$ (since ℓ is largest).

We claim that $s < \ell$.

Proof

$$s \in B \quad c_s^* a_{se} < 0 \quad x_s \text{ not basic in } D^*$$

Show $s < \ell$.

Proof

$$s \in B \quad c_s^* a_{se} < 0 \quad x_s \text{ not basic in } D^*$$

Show $s < \ell$.

x_ℓ leaves in D with x_e entering, so $a_{\ell e} > 0$.

Proof

$$s \in B \quad c^*_s a_{se} < 0 \quad x_s \text{ not basic in } D^*$$

Show $s < \ell$.

x_ℓ leaves in D with x_e entering, so $a_{\ell e} > 0$.

x_ℓ enters in D^*, so $c^*_\ell > 0$.

Proof

$$s \in B \quad c^*_s a_{se} < 0 \quad x_s \text{ not basic in } D^*$$

Show $s < \ell$.

x_ℓ leaves in D with x_e entering, so $a_{\ell e} > 0$.

x_ℓ enters in D^*, so $c^*_\ell > 0$.

Consequently, $c^*_\ell a_{\ell e} > 0$, so $s < \ell$.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

But then $a_{se} > 0$, since $c_s^* a_{se} < 0$.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

But then $a_{se} > 0$, since $c_s^* a_{se} < 0$.

$x_s \in \mathcal{F}$ and $s \in B$, so the value of x_s in the BFS is zero, $(b_s = 0)$.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

But then $a_{se} > 0$, since $c_s^* a_{se} < 0$.

$x_s \in \mathcal{F}$ and $s \in B$, so the value of x_s in the BFS is zero, $(b_s = 0)$.

Since $b_s = 0$ and $a_{se} > 0$, we have $\frac{b_s}{a_{se}} = 0$ which is the minimum ration in D. That is, x_s is a candidate for leaving in D.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

But then $a_{se} > 0$, since $c_s^* a_{se} < 0$.

$x_s \in \mathcal{F}$ and $s \in B$, so the value of x_s in the BFS is zero, $(b_s = 0)$.

Since $b_s = 0$ and $a_{se} > 0$, we have $\frac{b_s}{a_{se}} = 0$ which is the minimum ration in D. That is, x_s is a candidate for leaving in D.

But x_ℓ leaves in D with $s < \ell$.

Proof

Since $s < \ell$, x_s cannot be a candidate to enter the basis in D^* by **SSR**, that is, $c_s^* < 0$.

But then $a_{se} > 0$, since $c_s^* a_{se} < 0$.

$x_s \in \mathcal{F}$ and $s \in B$, so the value of x_s in the BFS is zero, $(b_s = 0)$.

Since $b_s = 0$ and $a_{se} > 0$, we have $\frac{b_s}{a_{se}} = 0$ which is the minimum ration in D. That is, x_s is a candidate for leaving in D.

But x_ℓ leaves in D with $s < \ell$.

This contradicts the **SSR**, so no cycle can exist.