The (Revised) Simplex Method

Key questions:

1. How do we find an initial BFS (extreme point)?
2. How do we know if the current BFS is optimal?
3. How do we move to a better BFS when one exists?

Let us begin with Question 2…

L.P. OPTIMALITY CONDITIONS

Min \( cx = z \)

\[ \begin{align*}
\text{st} \quad & Ax = b \quad x \geq 0; \\
\quad & \text{where } A \text{ is } m \times n \quad c, x \in \mathbb{R}^n \quad b \in \mathbb{R}^m
\end{align*} \]

Let \( B \) = index set of Basic variables

\( N \) = index set of Nonbasic variables = \( \{1,2,\ldots,n \setminus B\} \)

The original problem may be partitioned and rearranged to write our equations via

\[
(P) \begin{cases}
\quad c^B x^B + c^N x^N = z \\
\quad A^B x^B + A^N x^N = b \\
\quad x^B, x^N \geq 0
\end{cases}
\]

\[ b, c^B, x^B \in \mathbb{R}^m \]

\[ c^N, x^N \in \mathbb{R}^{m-n} \]

\[ A^B \equiv m \times m, A^N \equiv m \times (n - m), \]

Here \( A^B \) is the (non-singular) submatrix of \( A \) corr. to columns for basic variables while \( A^N \) is the submatrix of \( A \) corr. to columns for nonbasic variables, and \( e^B \) and \( e^N \) are similarly defined from \( c \).
Since B is a basis, \( A^B \) is a nonsingular matrix and \( (A^B)^{-1} \) exists. Therefore

\[
(A^B)^{-1}[A^B x^B + A^N x^N] = x^B + (A^B)^{-1}A^N x^N = (A^B)^{-1}b
\]

i.e.,

\[
x^B = (A^B)^{-1}b - (A^B)^{-1}A^N x^N = (A^B)^{-1}b - \sum_{j \in N} (A^B)^{-1}a^j x_j
\]

where \( a^j \) is the column in \( A \) for \( x_j \). That is,

\[
x^B = \bar{b} - \sum_{j \in N} y^j x_j
\]

where \( \bar{b} = (A^B)^{-1}b \ (\geq 0) \) and \( y^j = (A^B)^{-1}A^j \).

Substituting into the objective we can write \( z \) in term of the nonbasic variables:

\[
z = c^B x^B + c^N x^N = c^B \left[ (A^B)^{-1}b - \sum_{j \in N} (A^B)^{-1}A^j x_j \right] + \sum_{j \in N} c_j x_j
\]

Suppose that at the current basis the objective evaluated above (with \( x_j = 0 \) for \( j \in N \)) is equal to \( z_0 (= c^B (A^B)^{-1}b) \)

Let us define

\[
\pi^B = c^B (A^B)^{-1} \quad \text{and} \quad z_j = c^B (A^B)^{-1}A^j = \pi^B A^j = c^B y^j
\]

Then the general expression for \( z \) is given by

\[
z = z_0 - \sum_{j \in N} (z_j - c_j) x_j
\]
Therefore (P) is equivalent to

(P1) \[\begin{align*}
\text{Minimize } z_0 - \sum_{j \in N} (z_j - c_j) x_j
\end{align*}\]
\[st \sum_{j \in N} y^j x_j + x^B = \bar{b}, x_j \geq 0, j \in N, x^B \geq 0\]

It is clear that the basic variable \(x^B\) is just a vector of slack variables for the above system and we can therefore reduce this LP to the following one:

(P2) \[\begin{align*}
\text{Minimize } z_0 - \sum_{j \in N} (z_j - c_j) x_j
\end{align*}\]
\[st \sum_{j \in N} y^j x_j \leq \bar{b}, x_j \geq 0, j \in N\]

Note that this is an LP in \(n-m\) variables in the space of the nonbasic variables.

The coefficients \((z_j - c_j)\) for the nonbasic variable \(x_j\) is called its REDUCED COST COEFFICIENT, and the vector of these coefficients for all nonbasic variables is called the REDUCED COST VECTOR and given by

\[\tilde{c} = c^B \left( A^B \right)^{-1} A^N - c^N = \pi^B A^N - c^N\]

The key result that follows is that if \((z_j - c_j) \leq 0\) for every \(j \in N\), then (since \(\bar{b} \geq 0\)) the optimal solution to this LP has all such \(x_j = 0\), and thus the current BFS to the original problem is the optimal one and the optimal value of the LP is given by \(z_0\).
We thus have the following OPTIMALITY CONDITIONS

**Minimization:** If the reduced cost vector \( \hat{c} \) is wholly nonpositive (\( z_j - c_j \leq 0 \) for all \( j \in N \)), then \( B \) is an optimal basis.

**Maximization:** If the reduced cost vector \( \hat{c} \) is wholly nonnegative (\( z_j - c_j \geq 0 \) for all \( j \in N \)), then \( B \) is an optimal basis.

These conditions form the foundation for the REVISED SIMPLEX method.

The row vector \( \pi^B = c^B (A^B)^{-1} \) is referred to as the vector of SIMPLEX MULTIPLIERS relative to Basis \( B \).

Note that \( \pi^B \in R^m \), i.e., it has \( m \) elements, one corresponding to each constraint. Each element of this vector is called the simplex multiplier for the corresponding constraint.

**IMPROVEMENT**

Now suppose that the current basis is not optimal, i.e., there is at least one \( x_j, j \in N \) for which \( (z_j - c_j) > 0 \) (assuming minimization). Thus increasing this \( x_j \) from its current value of 0 while holding all the other nonbasic variables at zero is guaranteed to improve the objective from its current value of \( z_0 \).

In particular, if we pick \( x_k, k \in N \) to increase while retaining \( x_j = 0 \) for all the other \( j \in N - \{k\} \) at zero then the objective of (P1) or (P2) indicates that the new value is

\[
z = z_0 - (z_k - c_k)x_k
\]
EDGE DIRECTIONS AND MAINTAINING FEASIBILITY

We also have to simultaneously maintain feasibility. Looking at the constraints from (P1) this means that we want \( y^k x_k + x^B = \bar{b} \)

More clearly, we want

\[
\begin{bmatrix}
  x_1^B \\
  x_2^B \\
  \vdots \\
  x_r^B \\
  \vdots \\
  x_m^B
\end{bmatrix}
= \begin{bmatrix}
  \bar{b}_1 \\
  \bar{b}_2 \\
  \vdots \\
  \bar{b}_r \\
  \vdots \\
  \bar{b}_m
\end{bmatrix}
- \begin{bmatrix}
  y_1^k \\
  y_2^k \\
  \vdots \\
  y_r^k \\
  \vdots \\
  y_m^k
\end{bmatrix} x_k
\]

Assuming that the current BFS is nondegenerate (\( \bar{b} \succ 0 \)) consider the direction

\[
d = \begin{bmatrix}
  -y_k \\
  e_k
\end{bmatrix}
\]

where \( e_k \) is a vector in \( \mathbb{R}^n \) with zeros everywhere except for a 1 in position \( k \).

This direction \( d \) is a feasible direction (we can move a nonzero distance \( \Delta \) along this direction from the current solution \( x = [x^B] = [\bar{b}] \) and maintain feasibility).

To see this consider \( x + \Delta d = [\bar{b}] + \Delta [ -y_k ] \) = \( [\bar{b} - \Delta y_k] = [ \bar{b} - \Delta e_k ] \) for some \( \Delta > 0 \)

First, we can always maintain nonnegativity for sufficiently small \( \Delta \).

Second, \( A(x+\Delta d) = (A^B + A^N)(x+\Delta d) = A^B(\bar{b} - \Delta y_k) + \Delta A^k = A^B \bar{b} - \Delta A^B y_k + \Delta A^k = b - \Delta A^k + \Delta A^k = b \).

(recall that \( y^k = (A^B)^{-1} A^k \) and \( \bar{b} = (A^B)^{-1} b \))

So \( x+\Delta d \) is also feasible.

The direction \( d \) is called an edge direction since we move along an edge from the current vertex \( \bar{b} \).
Now, clearly if $y^k_r \leq 0$ then $x^B_r$ increases in value as $x_k$ increases and so $x^B_r$ continues to be nonnegative and stays feasible. However, if $y^k_r > 0$ then $x^B_r$ decreases in value as $x_k$ increases and so $x^B_r$ might eventually become zero. In order to maintain the nonnegativity of all $x^B_r$ we will increase $x_k$ until the first point at which some $x^B_r$ reaches zero. It is clear that this value for the increase is

$$v = \frac{\bar{b}_j}{y_j^k} \equiv \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_i^k} | y_i^k > 0 \right\}$$

As $x_k$ increases to this value $v$ we get a new basic feasible solution with $x_k$ replacing $x^B_j$ (which is now zero) in the basis and an improved objective value of

$$z_0 - (z_k - c_k)x_k = z_0 - (z_k - c_k)\frac{\bar{b}_j}{y_j^k}$$

Substituting $x_k = \frac{\bar{b}_j}{y_j^k}$ in (♣) the new solution has

$$x^B_i = \frac{\bar{b}_l}{y_j^k} - \frac{y_i^k}{y_j^k} \bar{b}_j, \quad i = 1, 2, ..., m, i \neq j; \quad x^B_j = x_k = v = \frac{\bar{b}_j}{y_j^k}$$

and all other $x_i = 0$.

Recall that if we replace vector $\mathbf{x}^i$ in a basis $[\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n]$ for $\mathbb{R}^n$ with a new vector (say $\mathbf{x}$), a necessary and sufficient condition to ensure that set of vectors $[\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^i, \mathbf{x}, \mathbf{x}^{i+1}, ..., \mathbf{x}^n]$ is linearly independent is to pick $\mathbf{x}^i$ such that $\lambda_j > 0$ in the expression $\mathbf{x} = \sum_{i=1}^n \lambda_i \mathbf{x}^i$. 
Now, recall that we defined \( y^k = (A^B)^{-1}A^k \), i.e.,

\[
A^k = A^B y^k = A^{B_1} y^k_{1} + \cdots + A^{B_r} y^k_{r} + \cdots + A^{B_m} y^k_m.
\]

So if we replace the set of basis vectors \([A^{B_1}, \ldots, A^{B_r}, \ldots, A^{B_m}]\) with the set \([A^{B_1}, \ldots, A^{B_{r-1}}, A^k, A^{B_{r+1}}, \ldots, A^{B_m}]\) then this set of vectors will be linearly independent as long as \( y^k_r > 0 \), and this is indeed the case here.

Therefore, the new set of vectors also correspond to a basic feasible solution.

In summary, each unit increase in the entering variable \( x_k \) will improve the objective function by \( \hat{c}_k = z_k - c_k = \pi^B A^k - c_k \) units, and the allowable increase in \( x_k \) is bounded by \( \min_{1 \leq i \leq m} \{ \frac{x_i^B}{y^k_i} | y^k_i > 0 \} \)

QUESTION: What if all \( y^k_i \) are \( \leq 0 \)?

From \((\jmath)\), it is clear that all the current basic variables increase in value (from \( \vec{b} \)) as \( x_k \) increases and thus there is no danger of ever violating feasibility with a basic variable reaching a value of zero. Concurrently the objective improves indefinitely, and thus there is no optimum solution and the optimal objective is unbounded in its value. Specifically, the direction \( d = \begin{bmatrix} -y^k_k \\ e_k \end{bmatrix} \) is a direction of unboundedness and the LP Is unbounded along the ray \( \begin{bmatrix} \vec{b} \\ 0 \end{bmatrix} + x_k \begin{bmatrix} -y^k_k \\ e_k \end{bmatrix}; x_k \geq 0 \)
### REVISED SIMPLEX METHOD

<table>
<thead>
<tr>
<th>STEP 0</th>
<th>Find an initial basis $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP 1</td>
<td>Extract $A^B$ from $A$ and $c^B$ from $c$. Then compute $(A^B)^{-1}$ and use this to compute $x^B = (A^B)^{-1}b$ and $z = c^B x^B$</td>
</tr>
<tr>
<td>STEP 2</td>
<td>Compute the simplex multiplier vector $\pi^B = c^B (A^B)^{-1}$</td>
</tr>
<tr>
<td>STEP 3</td>
<td>Price the nonbasic variables: Compute the reduced cost vector $\tilde{c} = \pi^B A^N - c^N$. For minimization stop if $\tilde{c} \leq 0$ (or if $\tilde{c} \geq 0$ for maximization…). Otherwise select a nonbasic variable to enter into the basis as any $x_k \in N</td>
</tr>
<tr>
<td>STEP 4</td>
<td>Compute the updated entering column via $y^k = (A^B)^{-1}A^k$. If $y^k_i \leq 0$ for all $1 \leq i \leq m$, STOP; the problem has an unbounded objective and no optimal solution. Otherwise, find the variable $x^B_j$ to remove from the basis via $j \in \text{argmin}_{1 \leq i \leq m} \left{ \frac{x^B_i}{y^k_i}</td>
</tr>
<tr>
<td>STEP 5</td>
<td>Return to Step 1 with the new basis $B$</td>
</tr>
</tbody>
</table>
AN EXAMPLE

Max \(4x_1 + 3x_2\)

St \(x_1 + 4x_2 \leq 52\)

\[14x_1 + 4x_2 \leq 156\]

\(x_1 \leq 10\) \(x_1, x_2 \geq 0\)

After adding slack variables the problem is

\[\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 4 & 3 & 0 & 0 & 0 \\ 14 & 4 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix}; \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \geq \begin{bmatrix} 0 \end{bmatrix}\]

STEP 0: The last 3 columns provide us with an initial feasible basis matrix, i.e.,

\(B = (3, 4, 5) \quad N = (1, 2)\)

**ITERATION 1**

STEP 1: Compute current solution

\(A^B = [A^3 \quad A^4 \quad A^5] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad A^N = [A^1 \quad A^2] = \begin{bmatrix} 1 & 4 \\ 14 & 4 \\ 1 & 0 \end{bmatrix}\)

\(c^B = [0 \quad 0 \quad 0]; \quad c^N = [4 \quad 3];\)
\[ x^B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = (A^B)^{-1} b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix} = \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix}; \quad x^N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

\[ z=c^B x^B = \begin{bmatrix} 0 & 0 \\ 0 & 156 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix} = 0 \]

**STEP 2:** Compute the simplex multiplier vector

\[ \pi^B = c^B (A^B)^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

**STEP 3:** Price out the nonbasic variables

\[ \hat{c} = \pi^B A^N - c^N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 14 & 4 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} -4 & -3 \end{bmatrix} \]

Both entries are negative so this cannot be optimal (remember we are maximizing…) and either NBV could enter the basis. Let us pick the first NBV \( x^N_1 = x_1 \) (i.e., \( k = 1 \)).

**STEP 4:** Update the entering column and find leaving variable

\[ y^1 = (A^B)^{-1} A^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 14 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 14 \\ 1 \end{bmatrix} \]

Find the column \( j \) to be replaced in \( B \):
\( j \in \arg\min_{1 \leq i \leq m} \left\{ \frac{x_i^B}{y_i} y_i^k \right\} > 0 \) = \arg\min_{1 \leq i \leq 3} \left\{ \frac{x_1^B}{y_1}, \frac{x_2^B}{y_2}, \frac{x_3^B}{y_3} \right\} = \arg\min_{1 \leq i \leq 3} \left\{ \frac{52}{1}, \frac{156}{14}, \frac{10}{1} \right\} = 3 \) with a minimum ratio value of 10. So we remove the third basic variable \((x_3^B = x_5)\)

STEP 5: Update the basis; we now have \(B = (3,4,1)\) \(N = (5,2)\)

**ITERATION 2**

STEP 1: Compute current solution

\[
A^B = [A^3 \ A^4 \ A^1] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 14 \\ 0 & 0 & 1 \end{bmatrix}; \quad A^N = [A^5 \ A^2] = \begin{bmatrix} 0 & 4 \\ 0 & 4 \\ 1 & 0 \end{bmatrix}
\]

\[
(A^B)^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 14 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -14 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
c^B = [0 \ 0 \ 4]; \quad c^N = [0 \ 3];
\]

\[
x^B = \begin{bmatrix} x_3^B \\ x_4^B \\ x_1^B \end{bmatrix} = (A^B)^{-1} b = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -14 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix} = \begin{bmatrix} 42 \\ 16 \\ 10 \end{bmatrix}; \quad x^N = \begin{bmatrix} x_5^B \\ x_2^B \end{bmatrix} = [0]
\]

\[
z = c^B x^B = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 42 \\ 16 \\ 10 \end{bmatrix} = 40
\]

STEP 2: Compute the simplex multiplier vector

\[
\pi^B = c^B (A^B)^{-1} = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -14 \\ 0 & 0 & 1 \end{bmatrix} = [0 \ 0 \ 4]
\]
STEP 3: Price out the nonbasic variables

\[ \hat{c} = \pi^B A^N - c^N = \begin{bmatrix} 0 & 4 \\ 0 & 4 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 3 \end{bmatrix} = \begin{bmatrix} 4 & -3 \end{bmatrix} \]

Since \( \hat{c}_2 \) is negative so this cannot be optimal and entering \( x_2^N \) into the basis can improve \( z \). So enter \( x_2^N = x_2 \) (i.e., \( k=2 \)).

STEP 4: Update the entering column and find leaving variable

\[ y^2 = (A^B)^{-1} A^2 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -14 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} \]

Find the column \( j \) to be replaced in \( B \):

\[ j \in \text{argmin}_{1 \leq i \leq m} \left\{ \frac{x_i^B}{y_i^k} \middle| y_i^k > 0 \right\} = \text{argmin} \left\{ \frac{x_i^B}{y_1^k}, \frac{x_i^B}{y_2^k}, - \right\} = \text{argmin} \left\{ \frac{42}{4}, \frac{16}{4}, - \right\} = 2 \text{ with a minimum ratio value of 4.} \]

That is, we remove the second basic variable \( (x_2^B = x_4) \)

STEP 5: Update the basis; we now have \( B = (3,2,1) \) \( N = (5,4) \)

ITERATION 3

STEP 1: Compute current solution

\[ A^B = [A^3 \; A^2 \; A^1] = \begin{bmatrix} 1 & 4 & 1 \\ 0 & 4 & 14 \\ 0 & 0 & 1 \end{bmatrix}; \quad A^N = [A^5 \; A^4] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \]
\[
(A^B)^{-1} = \begin{bmatrix}
1 & 4 & 1 \\
0 & 4 & 14 \\
0 & 0 & 1
\end{bmatrix}^{-1} = \begin{bmatrix}
1 & -1 & 13 \\
0 & 1/4 & -7/2 \\
0 & 0 & 1
\end{bmatrix}
\]

\[c^B = [0 \ 3 \ 4 \ ]; \ c^N = [0 \ 0];\]

\[x^B = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} = (A^B)^{-1} b = \begin{bmatrix} 1 & -1 & 13 \\ 0 & 1/4 & -7/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix} = \begin{bmatrix} 26 \\ 4 \\ 10 \end{bmatrix}; \quad x^N = \begin{bmatrix} x_5 \\ x_4 \end{bmatrix} = [0]
\]

\[z = c^B x^B = [0 \ 3 \ 4] \begin{bmatrix} 26 \\ 4 \\ 10 \end{bmatrix} = 52
\]

**STEP 2:** Compute the simplex multiplier vector

\[\pi^B = c^B (A^B)^{-1} = [0 \ 3 \ 4] \begin{bmatrix} 1 & -1 & 13 \\ 0 & 1/4 & -7/2 \\ 0 & 0 & 1 \end{bmatrix} = [0 \ 3/4 \ -13/2]
\]

**STEP 3:** Price out the nonbasic variables

\[\hat{c} = \pi^B A^N - c^N = [0 \ 3/4 \ -13/2] \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} - [0 \ 0] = [-13/2 \ 3/4]
\]

Since \(\hat{c}_1\) is negative so this cannot be optimal and entering \(x^N_1\) into the basis can improve \(z\). So enter \(x^N_1 = x_5\) (i.e., \(k=5\)).

**STEP 4:** Update the entering column and find leaving variable
\[ y^5 = (A^B)^{-1}A^5 = \begin{bmatrix} 1 & -1 & 13 \\ 0 & 1/4 & -7/2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 13 \\ -7/2 \\ 1 \end{bmatrix} \]

Find the column \( j \) to be replaced in \( B \):

\[ j \in \arg\min_{1 \leq i \leq m} \left\{ \frac{x^B_i}{y^k_i} \mid y^k_i > 0 \right\} = \arg\min_{1 \leq i \leq 3} \left\{ \frac{x^B_i}{y^k_1}, -\frac{x^B_3}{y^k_2} \right\} = \arg\min_{1 \leq i \leq 3} \left\{ \frac{26}{13}, -\frac{10}{1} \right\} = 1 \text{ with a minimum ratio value of } 2. \]

That is, we remove the first basic variable \((x^B_1 = x_3)\)

**STEP 5:** Update the basis; we now have \( B = (5,2,1) \quad N = (3,4) \)**

**ITERATION 4**

**STEP 1:** Compute current solution

\[
(A^B)^{-1} = [A^5 \quad A^2 \quad A^1] = \begin{bmatrix} 0 & 4 & 1 \\ 0 & 4 & 14 \\ 1 & 0 & 1 \end{bmatrix}; \quad A^N = [A^3 \quad A^4] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

\[
A^B = \begin{bmatrix} 0 & 4 & 1 \\ 0 & 4 & 14 \\ 1 & 0 & 1 \end{bmatrix}^{-1} = \left( \frac{1}{13} \right) \begin{bmatrix} 1 & -1 & 13 \\ 7/2 & -1/4 & 0 \\ -1 & 1 & 0 \end{bmatrix}
\]

\[
c^B = [0 \quad 3 \quad 4]; \quad c^N = [0 \quad 0];
\]

\[
x^B = \begin{bmatrix} x_5 \\ x_2 \\ x_1 \end{bmatrix} = (A^B)^{-1}b = \left( \frac{1}{13} \right) \begin{bmatrix} 1 & -1 & 13 \\ 7/2 & -1/4 & 0 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 52 \\ 156 \\ 10 \end{bmatrix} = \begin{bmatrix} 2 \\ 11 \\ 8 \end{bmatrix};
\]

\[
x^N = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
\[ z = c^B x^B = [0 \quad 3 \quad 4] \begin{bmatrix} 2 \\ 11 \\ 8 \end{bmatrix} = 65 \]

STEP 2: Compute the simplex multiplier vector

\[ \pi^B = c^B (A^B)^{-1} = [0 \quad 3 \quad 4] \begin{bmatrix} 1 & -1 & 13 \\ 7/2 & -1/4 & 0 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{13} \end{bmatrix} = \begin{bmatrix} 1/2 & 1/4 & 0 \end{bmatrix} \]

STEP 3: Price out the nonbasic variables

\[ \hat{c} = \pi^B A^N - c^N = \begin{bmatrix} 1/2 & 1/4 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/4 \end{bmatrix} \]

We now see that \( \hat{c}_1 \) has all positive elements. So no further improvement in \( z \) is possible. STOP!

The optimal solution is given by \( x^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \\ x_5^* \end{bmatrix} = \begin{bmatrix} 8 \\ 11 \\ 0 \\ 0 \\ 2 \end{bmatrix} \) with \( z^* = 65 \)

**NOTE:** The inverse of the basis matrix at any step need not be computed from scratch; rather we may use EROs on the previous basis inverse. For this we need \( y^k \) and the pivot element \( y_{j^k}^k \) (found in Step 4).
At Iteration 1, \((A^B)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \ y^1 = \begin{bmatrix} 1 \\ 14 \\ 1 \end{bmatrix}\) and \(y_j^1 = y_3^1 = 1\)

To get the new \((A^B)^{-1}\), we augment the old \((A^B)^{-1}\) with \(y^k\) and reduce this column to a unit vector with a 1 in row \(j=3\):

\[
\begin{bmatrix} 1 & 0 & 0 : 1 \\ 0 & 1 & 0 : 14 \\ 0 & 0 & 1 : 1 \end{bmatrix} \quad \begin{array}{l}
R1 \leftarrow R1 - R3 \\
R2 \leftarrow R2 - 14(R3)
\end{array} \Rightarrow \begin{bmatrix} 1 & 0 & -1 : 0 \\ 0 & 1 & 14 : 0 \\ 0 & 0 & 1 : 1 \end{bmatrix}
\]

At Iteration 2, \((A^B)^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 14 \\ 0 & 0 & 1 \end{bmatrix}; \ y^2 = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix}\) and \(y_j^2 = y_2^1 = 4\)

To get the new \((A^B)^{-1}\), we augment the old \((A^B)^{-1}\) with \(y^k\) and reduce this column to a unit vector with a 1 in row \(j=2\):

\[
\begin{bmatrix} 1 & 0 & -1 : 4 \\ 0 & 1 & 14 : 4 \\ 0 & 0 & 1 : 0 \end{bmatrix} \quad \begin{array}{l}
R2 \leftarrow R2 ÷ 4 \\
R1 \leftarrow R1 - 4R2
\end{array} \Rightarrow \begin{bmatrix} 1 & 0 & -1 : 4 \\ 0 & 1/4 & -7/2 : 1 \\ 0 & 0 & 1 : 0 \end{bmatrix}
\]

At Iteration 3, \((A^B)^{-1} = \begin{bmatrix} 1 & -1 & 13 \\ 0 & 1/4 & -7/2 \\ 0 & 0 & 1 \end{bmatrix}; \ y^5 = \begin{bmatrix} 13 \\ -7/2 \\ 1 \end{bmatrix}\) and \(y_j^5 = y_1^5 = 13\)

To get the new \((A^B)^{-1}\), we augment the old \((A^B)^{-1}\) with \(y^k\) and reduce this column to a unit vector with a 1 in row \(j=1\):
\[
\begin{bmatrix}
1 & -1 & 13 & : & 13 \\
0 & 1/4 & -7/2 & : & -7/2 \\
0 & 0 & 1 & : & 1
\end{bmatrix}
\quad R1 \leftarrow R1 \div 13 \\
\begin{bmatrix}
1/13 & -1/13 & 1 & : & 1 \\
7/26 & -1/52 & 0 & : & 0 \\
-1/13 & 1/13 & 0 & : & 0
\end{bmatrix}
= \left( \frac{1}{13} \right)
\begin{bmatrix}
1 & -1 & 13 & : & 13 \\
7/2 & -1/4 & 0 & : & 0 \\
-1 & 1 & 0 & : & 0
\end{bmatrix}
\]

In practice, we don’t ever invert matrices: finding \( x^B = (A^B)^{-1}b \) or \( \pi^B = c^B (A^B)^{-1} \) or \( y^k = (A^B)^{-1}A^k \) are all equivalent to solving the system of equations \( A^B x^B = b \) or \( \pi^B A^B = c^B \) or \( A^B y^k = A^k \), respectively.

There are schemes for this whereby the basis matrix \( A^B \) is factorized as a product of elementary matrices; we will see this later…

**MULTIPLE OPTIMA**

Consider STEP 3 (pricing) and suppose that \( \hat{c} \leq 0 \) (so that we are at an optimum solution – assuming minimization), but at least one \( \hat{c}_k = z_k - c_k = 0 \) corresponding to some NBV \( x_k \). As \( x_k \) increases (while all the other NBVs stay at zero) we get points that are clearly distinct from \( x^* \) but with the exact same optimum value, because \( z = z^* - (z_k - c_k)x_k \).

Assuming no degeneracy and that \( y^k \) has at least one positive element we can increase \( x_k \) until its increase is blocked by a basic variable (if at all) where we reach an adjacent extreme point. We get alternative optimal solutions along the way; this
process generates infinitely many optimal solutions. In fact, any convex combination of these two extreme points is also an optimal solution. There may be more than two such optimal points, in which case any convex combination of these is also optimal.

Conversely, if every $z_j - c_j$ is negative then the optimum must be unique (WHY?)

**Theorem of Finite Convergence:** In the absence of degeneracy, for a feasible LP the simplex method converges in a finite number of iterations, either with an optimal solution or with the conclusion that the objective is unbounded. (WHY?)

**DEGENERACY**

In Step 3 (pricing) we did not specify which specific nonbasic variable we choose to enter into the basis. In a degenerate problem it is possible that if we don’t do this carefully, we might cycle endlessly without any improvement in the objective and thus fail to converge to a solution! A degenerate problem is one where there is an extreme point that has more than $n$ defining hyperplanes passing through the point. Such points do not have a unique representation for the corresponding basic feasible solution because more than $(n-m)$ variables are equal to zero at the point, but only $(n-m)$ can be treated as being nonbasic. So at least one basic variable must also be equal to zero.
Consider for such a problem

$$
\begin{bmatrix}
x_1^B \\
x_2^B \\
\vdots \\
x_r^B \\
x_{m}^B
\end{bmatrix}
= 
\begin{bmatrix}
\bar{b}_1 \\
\bar{b}_2 \\
\vdots \\
\bar{b}_r \\
\bar{b}_{m}
\end{bmatrix}
- 
\begin{bmatrix}
y_1^k \\
y_2^k \\
\vdots \\
y_r^k \\
y_{m}^k
\end{bmatrix}
x_k
$$

and suppose that $\bar{b}_r = 0$ and that $y_r^k > 0$. Then $v = \frac{\bar{b}_i}{y_j^f} \equiv \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_i^k} \mid y_i^k > 0 \right\}$ will be zero corresponding to row $r$. So $x_k$ enters the basis at a value of 0 (and replaces $x_j$) and the improvement in $z$ will also be zero via

$$
z_0 - (z_k - c_k)x_k = z_0 - (z_k - c_k) \cdot 0 = z_0
$$

Many times we will have $y_r^k > 0$ for every $\bar{b}_r = 0$ so that $v \neq 0$ and we don’t have this issue, and even if we do, at the next iteration (or the next…) we will have $y_r^k > 0$ for every $\bar{b}_r = 0$, at which point we would continue to make progress after temporarily “getting stuck” at the same extreme point (this is referred to as stalling).

However, there are situations where we might never break out and cycle forever e.g., this might happen if we choose the most promising reduced cost value and break ties by selecting the first candidate row! To avoid this we must use an anti-cycling rule to select the entering variable.
ANTI-CYCLING RULES

Bland’s Rule: Suppose we choose the natural ordering for the variables: \( x_1, x_2, \ldots \)

This rule says that the entering variable is the \textbf{first} one in this list that has a negative reduced cost (for a minimization). Then, among all potential leaving variables (i.e., if there is a tie for the value of \( v \)) again choose the first one that appears on this list. Sometimes this is also called the “smallest subscript” rule.

An alternative (and slightly more complex...) rule is the following where the entering variable is not restricted (unlike Bland’s rule).

Lexicographic Rule: For the selected entering variable we find

\[
I_0 = \{ j \mid \frac{\bar{b}_j}{y_j^k} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_i^k} \mid y_i^k > 0 \right\} \}
\]

If \( I_0 \) has only one entry we are done with \( x_j^B \) leaving the basis. Otherwise define

\[
I_1 = \{ j \mid \frac{y_j^{B_1}}{y_j^k} = \min_{i \in I_0} \left\{ \frac{y_i^{B_1}}{y_i^k} \mid y_i^k > 0 \right\} \}
\]

If \( I_1 \) has only one entry we are done with \( x_j^B \) leaving the basis. Otherwise define \( I_2 \) using \( I_1 \) and repeat the process.

In general, \( I_r \) is formed from \( I_{r-1} \) via
Eventually for some $r < m$ the set $I_r$ will have only one member and we select the corresponding NBV to leave the basis. Note that what we are doing is to start with the current RHS for the numerators in the ratio test. If there is a tie we replace the RHS with the updated column for the first basic variable and use only rows where there is a tie. If the tie is still not broken we use the updated column for the second basic variable and rows with a tie, etc. etc.

**Theorem:** If we implement the simplex method using one of these two rules, then it is guaranteed to terminate.

**GETTING STARTED**

Lastly, we look at how to get an initial basis. What we need is a set of variables whose columns in $A$ are linearly independent. Sometimes this is obvious: e.g., if had $Ax \leq b$, $x, b \geq 0$ we could just pick the slack variables.

We get around this issue by using *artificial variables*. Suppose we have the constraints in standard form $Ax = b$, $x, b \geq 0$ and $A$ has no identity submatrix.
Let us add an artificial variable to each constraint so that we now have 

$$Ax + x^A = b, \ x^A \geq 0.$$ 

Note that the artificial variable corresponding to constraint \(i\) is \(x^A_i\). Note that this gives an immediate BFS with \(x^A_1, x^A_2, \ldots, x^A_m\) in the basis with values \(b_1, b_2, \ldots, b_m\) respectively.

For example

\[
\begin{align*}
3x_1 - 6x_2 + x_4 &= 30 \\
6x_1 + 12x_2 + 3x_3 &= 75; \quad x_1, x_2, x_3, x_4 \geq 0.
\end{align*}
\]

yields

\[
\begin{align*}
3x_1 - 6x_2 + x_4 + x^A_1 &= 30 \\
6x_1 + 12x_2 + 3x_3 + x^A_2 &= 75; \quad \text{all variables } \geq 0.
\end{align*}
\]

and the BFS \(x^A_1 = 30, x^A_2 = 40\).

However, we have fundamentally altered the problem! The feasible region for this new problem is the same as the one for the original one if, and only if, \(x^A = 0\). Thus we can use the artificial variables as a tool to get an initial BFS but eventually they must all be driven to zero (in which case we have a point that is within the feasible region).

The two common approaches are the \textit{Big-M} method and the \textit{Two-Phase} method – we will look only at the second.
Two-Phase Method

**Phase I:** We solve the following problem:

Minimize \( w = x^T A (\sum_{j=1}^{m} x_j^A) \)

subject to \( Ax + x^A = b, \quad x, x^A \geq 0 \)

If at optimality, we obtain \( x^A \neq 0 \), then stop; the original problem is infeasible; notice that this is true because we cannot reduce the current system to the original system by driving all the artificial variables to their minimal value of zero. Otherwise these variables are nonbasic and the optimal basis vector \( x^B \) has only the original variables and the remaining original variables constitute \( x^N \). Now proceed to Phase II.

**Phase II:** We now go to STEP 0 of the RSM to solve the original problem as usual, starting with the basis obtained in Phase I.

**Pathological Case:** What if we get an optimal value of zero for the Phase I problem, but an artificial variables is in the optimal basis at a value of zero?

**Answer:** If possible, we could forcibly replace this degenerate artificial variable in the basis with a “legitimate” nonbasic variable and proceed as usual (the Phase I objective would still be 0). If this is not possible \( (y_r^k = 0) \) it implies that the original constraint set had a linearly dependent (and hence redundant constraint) that must be removed; *(refer to Griva et al. or Bazaraa et al.)*
COMPLEXITY OF THE SIMPLEX METHOD

The objective of complexity analysis is to evaluate an algorithm for a worst-case type situation, so as to obtain a performance guarantee. Algorithmic complexity is usually measured by the relationship between the expected no. of elementary operations (multiplications, additions, comparisons, etc.) and the problem size.

Example: The total size of a linear program may be represented by the triplet 

\((m, n, L)\) where

\(m\) = no. of constraints

\(n\) = no. of variables

\(L\) = No. of binary bits required to record all problem data (depends on \(c, A\) and \(b\)).

Thus the complexity of an LP algorithm is a function of \(m, n\) and \(L\) - say \(f(m, n, L)\).

Suppose there exists \(\tau > 0\) such that the total number of elementary operations required by the algorithm for any instance of the problem is no more than \(\tau f(\bullet)\).

Then we say that the algorithm is of order of complexity \(O(f(\bullet))\).

When the complexity function \(f(\bullet)\) is a polynomial function of the problem size \((m, n\) and \(L\) in the case of LP), we say the algorithm is polynomially bounded or is of polynomial complexity. Otherwise the algorithm is said to be a nonpolynomial-time algorithm.
To illustrate this point, suppose the entire problem size can be captured by some single number $n$. A polynomial time algorithm's complexity may be bounded by $\tau n^2$, in which case it is $O(n^2)$. On the other hand a nonpolynomial time algorithm may have its complexity bounded by a function such as $\tau 2^n$, in which case it is $O(2^n)$ - specifically, in this case we would say the algorithm has exponential complexity ($n$ appears as an exponent).

Polynomial time algorithms are obviously better since the effort required increases much more slowly with increase in problem size $n$. The simplex method is an exponential time algorithm. Noting that there could be as many as $\binom{n}{m}$ vertices for the feasible region, in the WORST case the method could actually go through ALL of these to come to the optimal solution (indeed one such problem was devised by Klee and Minty).

Note that the total no. of extreme points is given by

\[
\binom{n}{m} = \frac{n!}{m! (n-m)!} = \frac{n}{m} \frac{n-1}{m-1} \cdots \frac{n-(m-1)}{m-(m-1)} > \left(\frac{n}{m}\right)^m
\]

Also note that $\left(\frac{n}{m}\right)^m$ is $\geq 2^m$ whenever $n \geq 2m$. Thus the total number of vertices grows exponentially with $m$; for instance if $n=2m$, we could have over $2^m$ (for
\( m = 50, \ 2^m = 1.125 \times 10^{15} \) vertices! In the WORST case an effort that grows exponentially could thus be required.

Fortunately...

In PRACTICE the Simplex method rarely takes more than \( 3m \) iterations.

Although estimates of this based on numerous studies have ranged from \( 2m \) to \( 10m \), it is nevertheless not exponential, and more along order \( O(m^2n) \). When sparsity is an issue a regression equation of the form \( Km^\alpha nd^0.33 \) usually provides a good fit for the actual complexity (\( K \) is a constant and \( \alpha \in (1.25, 2.5) \)). Here \( d \) is the density of the coefficient matrix \( A \). Thus, although the Simplex method is theoretically "bad" (in the worst case), it usually does pretty well in the average case.

Later we shall study POLYNOMIAL TIME algorithms for LP.