DUAL SIMPLEX METHOD

In dual simplex method, the LP starts with an optimum (or better) objective function value which is infeasible. Iterations are designed to move toward feasibility without violating optimality. At the iteration when feasibility is restored, the algorithm ends.

In order to maintain optimality and move toward feasibility at each iteration, the following two conditions are employed.

**Dual Feasibility:** The leaving variable \( x_r \) is the basic variable having the most negative value. If all the basic variables are nonnegative, the algorithm ends.

**Dual Optimality:** The entering variable is determined from among the nonbasic variables as the one corresponding to

\[
\min_{x_j} \left\{ \frac{z_j - c_j}{\alpha_{rj}} \right\} \quad \text{where} \quad \alpha_{rj} < 0
\]

where \( \alpha_{rj} \) is the constraint coefficient of the tableau associated with the row of leaving variable \( x_r \) and the column of the entering variable \( x_j \).

**Example:**

**Primal:**

\[
\begin{align*}
\text{max} & \quad 3x_1 + 6x_2 + 3x_3 \\
\text{s.t.} & \quad 3x_1 + 4x_2 + x_3 \leq 3 \\
& \quad x_1 + 3x_2 + x_3 \leq 2 \\
& \quad x_1, x_2 \geq 0 \\
& \quad x_3 \leq 0
\end{align*}
\]

**Dual:**

\[
\begin{align*}
\text{min} & \quad 3y_1 + 2y_2 \\
\text{s.t.} & \quad 3y_1 + y_2 \geq 3 \\
& \quad 4y_1 + 3y_2 \geq 6 \\
& \quad y_1 + y_2 \leq 3 \\
& \quad y_1, y_2 \geq 0
\end{align*}
\]

\[
\begin{align*}
\min & \quad W \\
W - 3y_1 - 2y_2 & = 0 \\
3y_1 + y_2 - y_3 & = 3 \\
4y_1 + 3y_2 & = 6 \\
y_1 + y_2 & + y_5 = 3 \\
y_1, y_2, y_3, y_4, y_5 & \geq 0
\end{align*}
\]
Initial Dual Simplex Tableau:

<table>
<thead>
<tr>
<th></th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(y_4)</th>
<th>(y_5)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_3)</td>
<td>-3</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(y_4)</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>(y_5)</td>
<td>-4</td>
<td>-3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-6</td>
</tr>
<tr>
<td>(y_5)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Leaving variable: \(y_5\)

Variable

<table>
<thead>
<tr>
<th>(z_j - c_j)</th>
<th>(\alpha_{4j})</th>
</tr>
</thead>
<tbody>
<tr>
<td>row 0</td>
<td>-3</td>
</tr>
<tr>
<td>row 2</td>
<td>-4</td>
</tr>
<tr>
<td>ratio (\frac{z_j - c_j}{\alpha_{4j}})</td>
<td>3/4</td>
</tr>
</tbody>
</table>

Entering variable: \(y_2\)

Iteration 1)

<table>
<thead>
<tr>
<th></th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(y_4)</th>
<th>(y_5)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_3)</td>
<td>-1/3</td>
<td>0</td>
<td>0</td>
<td>-2/3</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>(y_4)</td>
<td>-5/3</td>
<td>0</td>
<td>1</td>
<td>-1/3</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>(y_5)</td>
<td>4/3</td>
<td>1</td>
<td>0</td>
<td>-1/3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(y_5)</td>
<td>-1/3</td>
<td>0</td>
<td>0</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Ratio | 1/5 | - | - | 2 | - |

Leaving variable: \(y_3\)

Iteration 2)

<table>
<thead>
<tr>
<th></th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(y_3)</th>
<th>(y_4)</th>
<th>(y_5)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_1)</td>
<td>0</td>
<td>0</td>
<td>-1/5</td>
<td>-3/5</td>
<td>0</td>
<td>21/5</td>
</tr>
<tr>
<td>(y_1)</td>
<td>1</td>
<td>0</td>
<td>-3/5</td>
<td>1/5</td>
<td>0</td>
<td>3/5</td>
</tr>
<tr>
<td>(y_1)</td>
<td>0</td>
<td>1</td>
<td>4/5</td>
<td>-3/5</td>
<td>0</td>
<td>6/5</td>
</tr>
<tr>
<td>(y_5)</td>
<td>0</td>
<td>0</td>
<td>-1/5</td>
<td>2/5</td>
<td>1</td>
<td>6/5</td>
</tr>
</tbody>
</table>

=> \(x_1 = 1/5, x_2 = 3/5\)
The Simplex Method with Bounds

It is common in linear programming problems to have bounds on some of the variables. These bounds can be either lower bounds, upper bounds or both. A variable with bounds is represented as follows:

\[ x_j^L \leq x_j \leq x_j^U \]

It is possible to treat these bounds as constraints and obtain standard notation by defining slack, excess and artificial variables before applying the traditional simplex method. However, this approach is not very efficient since it would lead to a dramatic increase in the total number of variables and constraints. Therefore, it is necessary to modify the simplex algorithm to handle the bounds on variables more effectively.

**Lower Bounds:**
The lower bounds can be handled with a minor modification in the model by change of variables.

Since \( x_j \geq x_j^L \), replace \( x_j \) by \( x_j' = x_j + x_j^L \geq 0 \).

Therefore, the simplex method can directly be applied to the linear programming problem without any modification after this new definition.

**Upper Bounds:**
The upper bounds can be replaced by,

\[ x_j + y_j = x_j^U \]

when \( x_j = 0 \), then \( y_j = x_j^U \). Here, \( y_j \) is referred to as the complementary variable. At any point in the iterations of the simplex method, it is possible to,

1. use \( x_j \) when \( 0 \leq x_j \leq x_j^U \) or
2. replace \( x_j \) by \( x_j^U - y_j \) when \( 0 \leq y_j \leq x_j^U \)

This choice is arbitrary, since the variables \( x_j \) and \( y_j \) are complementary to each other (i.e., when the value for \( x_j \) is known, it is possible to determine the value of \( y_j \) and vice versa).

The upper bound technique uses the following rule to make his choice:

**Rule:** Begin with choice 1 (i.e., use \( x_j \)). Whenever \( x_j = 0 \), use \( x_j \) as the nonbasic variable.

Whenever \( x_j = x_j^U \), use \( y_j \) as the nonbasic variable.

**Example 1:**
max \( z = 2x_1 + x_2 + 2x_3 \)
s.t. \[
4x_1 + x_2 = 12 \\
-2x_1 + x_3 = 4 \\
0 \leq x_1 \leq 4, 0 \leq x_2 \leq 14, 0 \leq x_3 \leq 6
\]

\( x_1^U = 4, x_2^U = 14, x_3^U = 6 \)

The simplex tableau is constructed as follows:
The optimal solution is:
\[ z^* = 22, \ x_1 = 1, \ x_2 = 8, \ x_3 = 6 \]