Chapter 4
The Simplex Algorithm and Goal Programming

to accompany
Introduction to Mathematical Programming: Operations Research, Volume 1
4th edition, by Wayne L. Winston and Munirpallam Venkataramanan

Presentation: H. Sarper

4.1 – How to Convert an LP to Standard Form

Before the simplex algorithm can be used to solve an LP, the LP must be converted into a problem where all the constraints are equations and all variables are nonnegative. An LP in this form is said to be in standard form.

Consider the following example.

Leather Limited manufactures two types of leather belts: the deluxe model and the regular model. Each type requires 1 square yard of leather. A regular belt requires 1 hour of skilled labor and a deluxe belt requires 2 hours of skilled labor. Each week, 40 square yards of leather and 60 hours of skilled labor are available. Each regular belt contributes $3 profit and each deluxe belt $4. Write an LP to maximize profit.
4.1 – How to Convert an LP to Standard Form

The decision variables are:
- \( x_1 \) = number of deluxe belts produced weekly
- \( x_2 \) = number of regular belts produced weekly

The appropriate LP is:

\[
\begin{align*}
\text{max } z &= 4x_1 + 3x_2 \\
\text{s.t. } &x_1 + x_2 \leq 40 \quad \text{(leather constraint)} \\
&2x_1 + x_2 \leq 60 \quad \text{(labor constraint)} \\
&x_1, x_2 \geq 0
\end{align*}
\]

To convert the leather and labor (\( \leq \)) constraints to equalities, we define for each constraint a slack variable \( s_i \) (\( s_i \) = slack variable for the \( i \) th constraint). A slack variable is the amount of the resource unused in the \( i \) th constraint.

The LP not in standard form is:

\[
\begin{align*}
\text{max } z &= 4x_1 + 3x_2 \\
\text{s.t. } &x_1 + x_2 \leq 40 \\
&2x_1 + x_2 \leq 60 \\
&x_1, x_2 \geq 0
\end{align*}
\]

The same LP in standard form is:

\[
\begin{align*}
\text{max } z &= 4x_1 + 3x_2 \\
\text{s.t. } &x_1 + x_2 + s_1 = 40 \\
&2x_1 + x_2 + s_2 = 60 \\
&x_1, x_2, s_1, s_2 \geq 0
\end{align*}
\]

In summary, if a constraint \( i \) of an LP is a \( \leq \) constraint, convert it to an equality constraint by adding a slack variable \( s_i \) to the \( i \) th constraint and adding the sign restriction \( s_i \geq 0 \).
4.1 – How to Convert an LP to Standard Form

A ≥ constraint can be converted to an equality constraint. Consider the formulation below:

\[
\begin{align*}
\text{min } z &= 50x_1 + 20x_2 + 30x_3 + 80x_4 \\
\text{s.t.} & \quad 400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500 \\
& \quad 3x_1 + 2x_2 \geq 6 \\
& \quad 2x_1 + 2x_2 + 4x_3 + 4x_4 \geq 10 \\
& \quad 2x_1 + 4x_2 + x_3 + 5x_4 \geq 8 \\
& \quad x_1, x_2, x_3, x_4 \geq 0
\end{align*}
\]

To convert the i th ≥ constraint to an equality constraint, define an excess variable (sometimes called a surplus variable) \(e_i\) (\(e_i\) will always be the excess variable for the i th ≥ constraint. We define \(e_i\) to be the amount by which i th constraint is over satisfied.

Transforming the LP on the previous slide to standard form yields:

\[
\begin{align*}
\text{min } z &= 50x_1 + 20x_2 + 30x_3 + 80x_4 \\
\text{s.t.} & \quad 400x_1 + 200x_2 + 150x_3 + 500x_4 - e_1 = 500 \\
& \quad 3x_1 + 2x_2 - e_2 = 6 \\
& \quad 2x_1 + 2x_2 + 4x_3 + 4x_4 - e_3 = 10 \\
& \quad 2x_1 + 4x_2 + x_3 + 5x_4 - e_4 = 8 \\
& \quad x_i, e_i > 0 \ (i = 1,2,3,4)
\end{align*}
\]

In summary, if the i th constraint of an LP is a ≥ constraint, it can be converted to an equality constraint by subtracting the excess variable \(e_i\) from the i th constraint and adding the sign restriction \(e_i \geq 0\).
4.1 – How to Convert an LP to Standard Form

If an LP has both ≤ and ≥ constraints, apply the previous procedures to the individual constraints. Consider the example below.

<table>
<thead>
<tr>
<th>Nonstandard Form</th>
<th>Standard Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>max z = 20x₁ + 15x₂</td>
<td>max z = 20x₁ + 15x₂</td>
</tr>
<tr>
<td>s.t. x₁ ≤ 100</td>
<td>s.t. x₁ + s₁ = 100</td>
</tr>
<tr>
<td>x₂ ≤ 100</td>
<td>x₂ + s₂ = 100</td>
</tr>
<tr>
<td>50x₁ + 35x₂ ≤ 6000</td>
<td>50x₁ + 35x₂ + s₃ = 6000</td>
</tr>
<tr>
<td>20x₁ + 15x₂ ≥ 2000</td>
<td>20x₁ + 15x₂ - e₄ = 2000</td>
</tr>
<tr>
<td>x₁, x₂ &gt; 0</td>
<td>x₁, x₂, s₁, s₂, s₃, e₄ &gt; 0</td>
</tr>
</tbody>
</table>

4.2 – Preview of the Simplex Algorithm

Suppose an LP with m constraints and n variables has been converted into standard form. The form of such an LP is:

max (or min) z = c₁x₁ + c₂x₂ + … + cₙxₙ

s.t. a₁₁x₁ + a₁₂x₂ + … + a₁ₙxₙ = b₁
     a₂₁x₁ + a₂₂x₂ + … + a₂ₙxₙ = b₂
     .                      .                      .
     .                      .                      .
     aₘ₁x₁ + aₘ₂x₂ + … + aₘₙxₙ = bₘ

xᵢ ≥ 0 (i = 1, 2, …, n)
4.2 – Preview of the Simplex Algorithm

If we define: 

\[
\begin{pmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_m
\end{pmatrix}
\]

The constraints may be written as a system of equations \( Ax = b \).

Consider a system \( Ax = b \) of \( m \) linear equations in \( n \) variables (where \( n \geq m \)).

A basic solution to \( Ax = b \) is obtained by setting \( n - m \) variables equal to 0 and solving for the remaining \( m \) variables. This assumes that setting the \( n - m \) variables equal to 0 yields a unique value for the remaining \( m \) variables, or equivalently, the columns for the remaining \( m \) variables are linearly independent.

To find a basic solution to \( Ax = b \), we choose a set of \( n - m \) variables (the nonbasic variables, or NBV) and set each of these variables equal to 0. Then we solve for the values of the \( n - (n - m) = m \) variables (the basic variables, or BV) that satisfy \( Ax = b \).

Different choices of nonbasic variables will lead to different basic solutions. Consider the basic solutions to the system of 2 equations shown to the right:

\[
\begin{align*}
x_1 + x_2 &= 3 \\
-x_2 + x_3 &= -1
\end{align*}
\]

The number of nonbasic variables is \( 3 - 2 = 1 \). Setting, for example, \( NBV = \{x_3\} \) (as shown to the right), then \( BV = \{x_1, x_2\} \). We can obtain the values for these basic variables by setting \( x_3 = 0 \). Solving we find \( x_1 = 2, x_2 = 1 \). Thus, \( x_1 = 2, x_2 = 1, \) and \( x_3 = 0 \) is a basic solution.
4.2 – Preview of the Simplex Algorithm

If NBV = \{x_1\} and BV = \{x_2, x_3\} are chosen instead, the basic solution becomes \( x_1 = 0, x_2 = 3, \) and \( x_3 = 2. \)

If NBV = \{x_2\} and BV = \{x_1, x_3\} are chosen instead, the basic solution becomes \( x_1 = 3, x_2 = 0, \) and \( x_3 = -1. \)

Some sets of \( m \) variables do not yield a basic solution. Consider the linear system shown to the right:

\[
\begin{align*}
  x_1 + 2x_2 + x_3 &= 1 \\
  2x_1 + 4x_2 + x_3 &= 3 \\
  x_1 + 2x_2 &= 1 \\
  2x_1 + 4x_2 &= 3
\end{align*}
\]

If NBV = \{x_3\} and BV = \{x_1, x_2\} the corresponding basic solution would be:

\[
\begin{align*}
  x_1 + 2x_2 &= 1 \\
  2x_1 + 4x_2 &= 3
\end{align*}
\]

Since this system has no solution, there is no basic solution corresponding to BV = \{x_1, x_2\}.

Any basic solution in which all variables are nonnegative is called a **basic feasible solution** (or **bfs**). For the basic solutions on the previous slides, \( x_1 = 2, x_2 = 1, x_3 = 0 \) and \( x_1 = 0, x_2 = 3, x_3 = 2 \) are basic feasible solutions, but the basic solution \( x_1 = 3, x_2 = 0, x_3 = -1 \) fails to be a bfs (because \( x_3 < 0 \)).

The following two theorems explain why the concept of a basic feasible solution is of great importance in linear programming:

**Theorem 1** The feasible region for any linear programming problem is a convex set. Also, if an LP has an optimal solution, there must be an extreme point of the feasible region that is optimal.

**Theorem 2** For any LP, there is a unique extreme point of the LP’s feasible region corresponding to each basic feasible solution. Also, there is at least one bfs corresponding to each extreme point in the feasible region.
4.2 – Preview of the Simplex Algorithm

The relationship between extreme points and basic feasible solutions outlined in Theorem 2, is seen in the Leather Limited problem. The LP (with slack variables) was:

max \( z = 4x_1 + 3x_2 \)

s.t. \( x_1 + x_2 + s_1 = 40 \)

\( 2x_1 + x_2 + s_2 = 60 \)

\( x_1, x_2, s_1, s_2 \geq 0 \)

Both inequalities are satisfied in the shaded area. The extreme points are of the feasible region are B, C, E, and F.

<table>
<thead>
<tr>
<th>Basic Variables</th>
<th>Nonbasic Variables</th>
<th>Basic Feasible Solution</th>
<th>Corresponds to Corner Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1, x_2 )</td>
<td>( s_1, s_2 )</td>
<td>( s_1 = s_2 = 0, x_1 = x_2 = 20 )</td>
<td>E</td>
</tr>
<tr>
<td>( x_1, s_1 )</td>
<td>( x_2, s_2 )</td>
<td>( x_2 = s_2 = 0, x_1 = 30, s_1 = 10 )</td>
<td>C</td>
</tr>
<tr>
<td>( x_1, s_2 )</td>
<td>( x_2, s_1 )</td>
<td>( x_2 = s_1 = 0, x_1 = 40, s_2 = -20 )</td>
<td>Not a bfs since ( s_2 &lt; 0 )</td>
</tr>
<tr>
<td>( x_2, s_1 )</td>
<td>( x_1, s_2 )</td>
<td>( x_1 = s_2 = 0, s_1 = -20 x_2 = 60 )</td>
<td>Not a bfs since ( s_1 &lt; 0 )</td>
</tr>
<tr>
<td>( x_1, s_2 )</td>
<td>( s_1, s_2 )</td>
<td>( x_1 = s_1 = 0, x_2 = 40, s_2 = 20 )</td>
<td>B</td>
</tr>
<tr>
<td>( s_1, s_2 )</td>
<td>( x_1, x_2 )</td>
<td>( x_1 = x_2 = 0, s_1 = 40, s_2 = 60 )</td>
<td>F</td>
</tr>
</tbody>
</table>

The table above shows the correspondence between the basic feasible solutions to the LP and the extreme points of the feasible region. The basic feasible solutions to the standard form of the LP correspond in a natural fashion to the LP’s extreme points.
4.2 – Preview of the Simplex Algorithm

Adjacent Basic Feasible Solutions

For any LP with m constraints, two basic feasible solutions are said to be adjacent if their sets of basic variables have m – 1 basic variables in common.

For example in the Leather Limited LP on the previous slide, the bfs corresponding to point E is adjacent to the bfs corresponding to point C. These points share (m – 1 = 2 - 1 = 1) one basic variable, x₁. Points E (BV = {x₁, x₂}) and F (BV = {s₁, s₂}) are not adjacent since they share no basic variables.

Intuitively, two basic feasible solutions are adjacent if they both lie on the same edge of the boundary of the feasible region.

4.2 – Preview of the Simplex Algorithm

General description of the simplex algorithm solving an LP in a maximization problem:

Step 1 Find a bfs to the LP. We will call this bfs the initial bfs. In general, the most recent bfs will be called the current bfs, so at the beginning of the problem, the initial bfs is the current bfs.

Step 2 Determine if the current bfs is an optimal solution to the LP. If it is not, find an adjacent bfs that has a larger z-value.

Step 3 Return to Step 2, using the new bfs as the current bfs.
4.3 – The Simplex Algorithm (max LPs)

The Simplex Algorithm Procedure for maximization LPs

Step 1  Convert the LP to standard form
Step 2  Obtain a bfs (if possible) from the standard form
Step 3  Determine whether the current bfs is optimal
Step 4  If the current bfs is not optimal, determine which nonbasic variable should be come a basic variable and which basic variable should become a nonbasic variable to find a bfs with a better objective function value.
Step 5  Use ero’s to find a new bfs with a better objective function value. Go back to Step 3.

In performing the simplex algorithm, write the objective function in the form: \( Z - c_1x_1 - c_2x_2 - \ldots - c_nx_n = 0 \)

We call this format the row 0 version of the objective function (row 0 for short).

4.3 – The Simplex Algorithm (max LPs)

Consider the simplex algorithm applied to the maximization below:

The Dakota Furniture company manufactures desks, tables, and chairs. The manufacturer of each type of furniture requires lumber and two types of skilled labor: finishing and carpentry. The amount of each resource needed to make each type of furniture is given in the table below.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Desk</th>
<th>Table</th>
<th>Chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lumber</td>
<td>8 board ft</td>
<td>6 board ft</td>
<td>1 board ft</td>
</tr>
<tr>
<td>Finishing hours</td>
<td>4 hours</td>
<td>2 hours</td>
<td>1.5 hours</td>
</tr>
<tr>
<td>Carpentry hours</td>
<td>2 hours</td>
<td>1.5 hours</td>
<td>0.5 hours</td>
</tr>
</tbody>
</table>
4.3 – The Simplex Algorithm (max LPs)

At present, 48 board feet of lumber, 20 finishing hours, 8 carpentry hours are available. A desk sells for $60, a table for $30, and a chair for $20.

Dakota believes that demand for desks and chairs is unlimited, but at most 5 tables can be sold.

Since the available resources have already been purchased, Dakota wants to maximize total revenue.

Define:

- $x_1$ = number of desks produced
- $x_2$ = number of tables produced
- $x_3$ = number of chairs produced.

The LP is:

$$ \text{max } z = 60x_1 + 30x_2 + 20x_3 $$

s.t.

- $8x_1 + 6x_2 + x_3 \leq 48$ (lumber constraint)
- $4x_1 + 2x_2 + 1.5x_3 \leq 20$ (finishing constraint)
- $2x_1 + 1.5x_2 + 0.5x_3 \leq 8$ (carpentry constraint)
- $x_2 \leq 5$ (table demand constraint)
- $x_1, x_2, x_3 \geq 0$
4.3 – The Simplex Algorithm (max LPs)

Step 1 - Convert the LP to Standard Form

Canonical Form 0

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Canonical Form 0</th>
<th>Basic Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z – 60x₁ – 30x₂ – 20x₃ = 0</td>
<td>z = 0</td>
</tr>
<tr>
<td>Row 1</td>
<td>8x₁ + 6x₂ + x₃ + s₁ = 48</td>
<td>s₁ = 48</td>
</tr>
<tr>
<td>Row 2</td>
<td>4x₁ + 2x₂ + 1.5x₃ + s₂ = 20</td>
<td>s₂ = 20</td>
</tr>
<tr>
<td>Row 3</td>
<td>2x₁ + 1.5x₂ + 0.5x₃ + s₃ = 8</td>
<td>s₃ = 6</td>
</tr>
<tr>
<td>Row 4</td>
<td>x₂ + s₄ = 5</td>
<td>s₄ = 5</td>
</tr>
</tbody>
</table>

If we set x₁ = x₂ = x₃ = 0, we can solve for the values s₁, s₂, s₃, s₄. Thus, BV = {s₁, s₂, s₃, s₄} and NBV = {x₁, x₂, x₃}. Since each constraint is then in canonical form (BVs have a coefficient = 1 in one row and zeros in all other rows) with a nonnegative rhs, a bfs can be obtained by inspection.

Step 2 – Obtain a Basic Feasible Solution

To perform the simplex algorithm, we need a basic (although not necessarily nonnegative) variable for row 0. Since z appears in row 0 with a coefficient of 1, and z does not appear in any other row, we use z as the basic variable. With this convention, the basic feasible solution for our initial canonical form has:

BV = {z, s₁, s₂, s₃, s₄} and NBV = {x₁, x₂, x₃}.

For this initial bfs, z = 0, s₁ = 48, s₂ = 20, s₃ = 8, s₄ = 5, x₁ = x₂ = x₃ = 0.

As this example indicates, a slack variable can be used as a basic variable if the rhs of the constraint is nonnegative.
4.3 – The Simplex Algorithm (max LPs)

Step 3 – Determine if the Current BFS is Optimal

Once we have obtained a bfs, we need to determine whether it is optimal. To do this, we try to determine if there is any way $z$ can be increased by increasing some nonbasic variable from its current value of zero while holding all other nonbasic variables at their current values of zero. Solving for $z$ in row 0 yields:

$$Z = 60x_1 + 30x_2 + 20x_3$$

For each nonbasic variable, we can use the equation above to determine if increasing a nonbasic variable (while holding all other nonbasic variables to zero) will increase $z$. Increasing any of the nonbasic variables will cause an increase in $z$. However increasing $x_1$ causes the greatest rate of increase in $z$. If $x_1$ increases from its current value of zero, it will have to become a basic variable. For this reason, $x_1$ is called the **entering variable**. Observe $x_1$ has the most negative coefficient in row 0.

4.3 – The Simplex Algorithm (max LPs)

Step 4 - We choose the **entering variable** (in a max problem) to the nonbasic variable with the most negative coefficient in row 0 (ties broken arbitrarily). We desire to make $x_1$ as large as possible but as we do, the current basic variables ($s_1$, $s_2$, $s_3$, $s_4$) will change value. Thus, increasing $x_1$ may cause a basic variable to become negative.

**RATIO**

From row 1 we see that $s_1 = 48 - 8x_1$. Since $s_1 \geq 0$, $x_1 \leq 48 / 8 = 6$

From row 2, we see that $s_2 = 20 - 4x_1$. Since $s_2 \geq 0$, $x_1 \leq 20 / 4 = 5$

From row 3, we see that $s_3 = 8 - 2x_1$. Since $s_3 \geq 0$, $x_1 \leq 8 / 2 = 4$

From row 4, we see that $s_4 = 5$. For any $x_1$, $s_4$ will always be $\geq 0$

This means to keep all the basic variables nonnegative, the largest we can make $x_1$ is min $(6, 5, 4) = 4$. 

Copyright (c) 2003 Brooks/Cole, a division of Thomson Learning, Inc.
4.3 – The Simplex Algorithm (max LPs)

The Ratio Test

When entering a variable into the basis, compute the ratio:

\[ \text{rhs of row} / \text{coefficient of entering variable in row} \]

For every constraint in which the entering variable has a positive coefficient. The constraint with the smallest ratio is called the winner of the ratio test. The smallest ratio is the largest value of the entering variable that will keep all the current basic variables nonnegative.

Make the entering variable \( x_1 \) a basic variable in row 3 since this row (constraint) was the winner of the ratio test (8/2 =4).

To make \( x_1 \) a basic variable in row 3, we use elementary row operations (ero’s) to make \( x_1 \) have a coefficient of 1 in row 3 and a coefficient of 0 in all other rows. This procedure is called pivoting on row 3; and row 3 is called the pivot row. The final result is that \( x_1 \) replaces \( s_3 \) as the basic variable for row 3. The term in the pivot row that involves the entering basic variable is called the pivot term.

Step 5 - The Gauss-Jordan method using ero’s and simplex tableaus shown on the next slide makes \( x_1 \) a basic variable.
4.3 – The Simplex Algorithm (max LPs)

The result is:

Canonical Form 1

<table>
<thead>
<tr>
<th>Row</th>
<th>Basic Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>z = 240</td>
</tr>
<tr>
<td>1</td>
<td>s1 = 16</td>
</tr>
<tr>
<td>2</td>
<td>s2 = 4</td>
</tr>
<tr>
<td>3</td>
<td>x1 = 4</td>
</tr>
<tr>
<td>4</td>
<td>s4 = 5</td>
</tr>
</tbody>
</table>

In canonical form 1, BV = \{z, s1, s2, x1, s4\} and NBV = \{s3, x2, x3\}.

yielding the bfs \( z = 240, s1 = 16, s2 = 4, x1 = 4, s4 = 5, s3 = x2 = x3 = 0 \).

The procedure from going to one bfs to a better adjacent bfs is called an iteration (or sometime pivot) of the simplex algorithm.
4.3 – The Simplex Algorithm (max LPs)

Returning to Step 3, we attempt to determine if the current bfs is optimal.

Rearranging row 0 from Canonical Form 1, and solving for \( a \) yields:

\[
z = 240 - 15x_2 + 5x_3 - 30s_3
\]

The current bfs is NOT optimal because increasing \( x_3 \) to 1 (while holding the other nonbasic variable to zero) will increase the value of \( z \). Making either \( x_2 \) or \( s_3 \) basic will caused the value of \( z \) to decrease.

Step 4 - Recall the rule for determining the entering variable is the row 0 coefficient with the greatest negative value. Since \( x_3 \) is the only variable with a negative coefficient, \( x_3 \) should be entered into the basis.

Begin iteration (pivot) 2

Performing the ratio test using \( x_3 \) as the entering variable yields the following results (holding other NBVs to zero):

- From row 1, \( s_1 \geq 0 \) for all values of \( x_3 \) since \( s_1 = 16 + x_3 \)
- From row 2, \( s_2 \geq 0 \) if \( x_3 > 4 / 0.5 = 8 \)
- From row 3, \( x_1 \geq 0 \) if \( x_3 > 4 / 0.25 = 16 \)
- From row 4, \( s_4 \geq 0 \) for all values of \( x_3 \) since \( s_4 = 5 \)

This means to keep all the basic variables nonnegative, the largest we can make \( x_1 \) is \( \min \{8,16\} = 8 \). So, row 2 becomes the pivot row.

Step 5 - Now use ero’s, to make \( x_3 \) a basic variable in row 2.
### 4.3 – The Simplex Algorithm (max LPs)

#### Canonical Form 2

<table>
<thead>
<tr>
<th>Row</th>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>s4</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-4</td>
<td>1</td>
<td>-4</td>
<td>1</td>
<td>-4</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>1</td>
<td>2</td>
<td>-4</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.75</td>
<td>0.25</td>
<td>0.5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

**Row 0 for z**

\[ z = 15z + 15x_1 + 15x_2 + 15x_3 = 240 \]

**Row 1 for s1**

\[ s_1 = 1x_1 - 4x_2 + 1s_1 + 2s_2 - 8s_3 = 24 \]

**Row 2 for x3**

\[ x_3 = 2x_2 + 2s_2 - 4s_3 = 8 \]

**Row 3 for x1**

\[ x_1 = 1.25x_2 + 0.25s_2 + 0.5s_3 = 2 \]

**Row 4 for s4**

\[ s_4 = 5x_2 + s_4 = 5 \]

#### The result is:

\[
\begin{align*}
\text{Canonical Form 2} & \\
\text{Basic Variable} & \\
\text{Row 0} & z + 5x_2 + 10s_2 + 10s_3 = 280 & z = 280 \\
\text{Row 1} & -2x_2 + s_1 + 2s_2 - 8s_3 = 24 & s_1 = 24 \\
\text{Row 2} & -2x_2 + x_3 + 2s_2 - 4s_3 = 8 & x_3 = 8 \\
\text{Row 3} & x_1 + 1.25x_2 - 0.5s_2 + 1.5s_3 = 2 & x_1 = 2 \\
\text{Row 4} & x_2 + s_4 = 5 & s_4 = 5
\end{align*}
\]

In Canonical Form 2, BV = \{z, s_1, x_3, x_1, s_4\} and NBV = \{s_3, s_2, x_2\}, yielding the bfs \( z = 280, s_1 = 24, x_3 = 8, x_1 = 2, s_4 = 5, s_2 = s_3 = x_2 = 0 \).
4.3 – The Simplex Algorithm (max LPs)

Solving for z in row 0 yields: \[ Z = 280 - 5x_2 - 10s_2 - 10s_3 \]

We can see that increasing \( x_2, s_2, \) or \( s_3 \) (while holding the other NBVs to zero) will not cause the value of \( z \) to decrease. The solution at the end of iteration 2 is therefore optimal. The following rule can be applied to determine whether a canonical form’s bfs is optimal:

A canonical form is optimal (for a max problem) if each nonbasic variable has a nonnegative coefficient in the canonical form’s row 0.

---

4.4 – The Simplex Algorithm (min LPs)

Two different ways the simplex method can be used to solve minimization problems. Consider the LP shown to the right:

\[
\begin{align*}
\text{min } z &= 2x_1 - 3x_2 \\
\text{s.t. } &x_1 + x_2 \leq 4 \\
&x_1 - x_2 \leq 6 \\
&x_1, x_2 \geq 0
\end{align*}
\]

**Method 1**

The optimal solution is the point \((x_1, x_2)\) that makes \( z = 2x_1 - 3x_2 \) the smallest. Equivalently, this point makes \( \text{max } -z = -2x_1 + 3x_2 \) the largest. This means we can find the optimal solution to the LP by solving the LP shown to the right.
In solving this modified LP, use \(-z\) as the basic variable in row 0.

After adding slack variables \(s_1\) and \(s_2\) to the constraints we obtain the initial tableau.

**Initial Tableau**

<table>
<thead>
<tr>
<th>Row</th>
<th>(-z)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(s_1)</th>
<th>(s_2)</th>
<th>rhs</th>
<th>BVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-z = 0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>(s_1 = 4)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>(s_2 = 6)</td>
</tr>
</tbody>
</table>

**ero1**

<table>
<thead>
<tr>
<th>Row</th>
<th>(-z)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(s_1)</th>
<th>(s_2)</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

**ero2**

<table>
<thead>
<tr>
<th>Row</th>
<th>(-z)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(s_1)</th>
<th>(s_2)</th>
<th>rhs</th>
<th>BVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>12</td>
<td>-z = 12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>(x_2 = 4)</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>(s_2 = 10)</td>
</tr>
</tbody>
</table>

The optimal solution (to the max problem) is \(-z = 12, x_2 = 4, s_2 = 10, x_1 = s_1 = 0\). Then the optimal solution to the min problem is \(z = -12, x_2 = 4, s_2 = 10, x_1 = s_1 = 0\). The min LP objective function confirms this: \(z = 2x_1 - 3x_2 = 2(0) - 3(4) = -12\).

In summary, multiply the objective function for the min problem by \(-1\) and solve the problem as a maximization problem with the objective function \(-z\). The optimal solution to the max problem will give you the optimal solution for the min problem. Remember that (optimal \(z\)-value for the min problem) = - (optimal \(z\)-value for the max problem).
4.4 – The Simplex Algorithm (min LPs)

**Method 2**

A simple modification of the simplex algorithm can be used to solve min problems directly. Modify Step 3 of the simplex algorithm as follows: If all nonbasic variables (NBV) in row 0 have nonpositive coefficients, the current bfs is optimal. If any nonbasic variable has a positive coefficient, choose the variable with the "most positive" coefficient in row 0 as the entering variable.

This modification of the simplex algorithm works because increasing a nonbasic variable (NBV) with a positive coefficient in row 0 will decrease $z$.

4.5 – Alternate Optimal solutions

For some LPs, more than one extreme point is optimal. If an LP has more than one optimal solution, it has multiple optimal solutions.

If there is no nonbasic variable (NBV) with a zero coefficient in row 0 of the optimal tableau, the LP has a unique optimal solution. Even if there is a nonbasic variable with a zero coefficient in row 0 of the optimal tableau, it is possible that the LP may not have alternative optimal solutions.
4.6 – Unbounded LPs

For some LPs, there exist points in the feasible region for which \( z \) assumes arbitrarily large (in max problems) or arbitrarily small (in min problems) values. When this occurs, we say the LP is unbounded.

An unbounded LP occurs in a max problem if there is a nonbasic variable with a negative coefficient in row 0 and there is no constraint that limits how large we can make this NBV.

Specifically, an unbounded LP for a max problem occurs when a variable with a negative coefficient in row 0 has a non positive coefficient in each constraint.

4.6 – The LINDO Computer Package

LINDO (Linear Interactive and Discrete Optimizer) is a user friendly computer package that can be used to solve linear, integer, and quadratic programming problems.

The Dakota Furniture LP will solved using LINDO as an example. LINDO permits the user to name the variables so they are defined:

- DESKS = number of desk produced
- TABLES = number of tables produced
- CHAIRS = number of chairs produced

LINDO assumes all variables are nonnegative, so nonnegative constraints are not necessary. To be consistent with LINDO, the objective function row is labeled row 1 and constraints rows 2-5.

View the LINDO Help file for syntax questions.
To enter this problem into LINDO make sure the screen contains a blank window (work area) with the word “untitled” at the top of the work area. If necessary, a new window can be opened by selecting New from the file menu or by clicking on the File button.

The first statement in a LINDO model is always the objective function. MAX or MIN directs LINDO to solve a maximization or minimization problem.

Enter the constraints by typing SUBJECT TO (or st) on the next line. Then enter the constraints. LINDO assumes the < symbol means ≤ and the > symbol means ≥. There is no need to insert the asterisk symbol between coefficients and variables to indicate multiplication. The Dakota input is shown on the next slide.
4.6 – The LINDO Computer Package

To save the file (LP formulation) for future use, select SAVE from the File menu and when asked, replace the * symbol with the name of your choice. The file will be saved with the name you select with the suffix .LTX. DO NOT type over the .LTX suffix. Save using any path available.

To solve the model, from the Solve menu select the SOLVE command or click the red bulls eye button.

When asked if you want to do a range (sensitivity) analysis, choose no. Range or sensitivity analysis will be discussed in Chapter 6, when the solution is complete, a display showing the status of the Solve command will be present. View the displayed information and select CLOSE.

The data input window will now be overlaying the Reports Window. Click anywhere in the Reports Window to bring it to the foreground. View the next slide to see this Reports Window.
4.6 – The LINDO Computer Package

The LINDO output shows:
- LINDO found the optimum solution after 2 iterations (pivots)
- The optimal z-value = 280
- VALUE gives the value of the variable in the optimal LP solution. Thus the optimal solution calls for production of 2 desks, 0 tables, and 8 chairs.
- SLACK OR SURPLUS gives (by constraint row) the value of slack or excess in the optimal solution.
- REDUCED COST gives the coefficient in row 0 of the optimal tableau (in a max problem). The reduced cost of each basic variables must be 0. For NBVs, the reduced cost is the amount by which the optimal z-value will be reduced if the NBV is increased by 1 unit (and all other NBVs remain 0).
- The other LINDO outputs will be discussed in Chapters 5 and 6.

4.10 – The Big M Method

The simplex method algorithm requires a starting bfs. Previous problems have found starting bfs by using the slack variables as our basic variables. If an LP have \( \geq \) or = constraints, however, a starting bfs may not be readily apparent. In such a case, the Big M method may be used to solve the problem. Consider the problem below.

Bevco manufactures an orange-flavored soft drink called Oranj by combining orange soda and orange juice. Each orange soda contains 0.5 oz of sugar and 1 mg of vitamin C. Each ounce of orange juice contains 0.25 oz of sugar and 3 mg of vitamin C. It costs Bevco 2¢ to produce an ounce of orange soda and 3¢ to produce an ounce of orange juice. Bevco’s marketing department has decided that each 10-oz bottle of Oranj must contain at least 30 mg of vitamin C and at most 4 oz of sugar. Use linear programming to determine how Bevco can meet the marketing department’s requirements at minimum cost.
4.10 – The Big M Method

Letting \( x_1 = \) number of ounces of orange soda in a bottle of Oranj \\
\( x_2 = \) number of ounces of orange juice in a bottle of Oranj 

The LP is:

\[
\begin{align*}
\text{min } z &= 2x_1 + 3x_2 \\
\text{st } &0.5x_1 + 0.25x_2 \leq 4 \quad \text{(sugar constraint)} \\
&x_1 + 3x_2 \geq 20 \quad \text{(Vitamin C constraint)} \\
&x_1 + x_2 = 10 \quad \text{(10 oz in 1 bottle of Oranj)} \\
&x_1, x_2 > 0 
\end{align*}
\]

The LP in standard form is shown on the next slide.

4.10 – The Big M Method

The LP in standard form has \( z \) and \( s_1 \) which could be used for BVs but row 2 would violate sign restrictions and row 3 no readily apparent basic variable.

In order to use the simplex method, a bfs is needed. To remedy the predicament, artificial variables are created. The variables will be labeled according to the row in which they are used as seen below.

\[
\begin{align*}
\text{Row 1: } z &= -2x_1 - 3x_2 = 0 \\
\text{Row 2: } 0.5x_1 + 0.25x_2 + s_1 &= 4 \\
\text{Row 3: } x_1 + 3x_2 - e_2 &= 20 \\
\text{Row 4: } x_1 + x_2 + a_3 &= 10 
\end{align*}
\]
4.10 – The Big M Method

In the optimal solution, all artificial variables must be set equal to zero. To accomplish this, in a min LP, a term \( Ma_i \) is added to the objective function for each artificial variable \( a_i \). For a max LP, the term \(-Ma_i\) is added to the objective function for each \( a_i \). \( M \) represents some very large number. The modified Bevco LP in standard form then becomes:

\[
\begin{align*}
\text{Row 1: } & z - 2x_1 - 3x_2 - Ma_2 - Ma_3 = 0 \\
\text{Row 2: } & 0.5x_1 + 0.25x_2 + s_1 = 4 \\
\text{Row 3: } & x_1 + 3x_2 - e_2 + a_2 = 20 \\
\text{Row 4: } & x_1 + x_2 + a_3 = 10
\end{align*}
\]

Modifying the objective function this way makes it extremely costly for an artificial variable to be positive. The optimal solution should force \( a_2 = a_3 = 0 \).

4.10 – The Big M Method

Description of the Big M Method

1. Modify the constraints so that the rhs of each constraint is nonnegative. Identify each constraint that is now an \( = \) or \( \geq \) constraint.
2. Convert each inequality constraint to standard form (add a slack variable for \( \leq \) constraints, add an excess variable for \( \geq \) constraints).
3. For each \( \geq \) or \( = \) constraint, add artificial variables. Add sign restriction \( a_i \geq 0 \).
4. Let \( M \) denote a very large positive number. Add (for each artificial variable) \( Ma_i \) to min problem objective functions or \(-Ma_i\) to max problem objective functions.
5. Since each artificial variable will be in the starting basis, all artificial variables must be eliminated from row 0 before beginning the simplex. Remembering \( M \) represents a very large number, solve the transformed problem by the simplex.
If all artificial variables in the optimal solution equal zero, the solution is optimal. If any artificial variables are positive in the optimal solution, the problem is infeasible.

The Bevco example continued:

Initial Tableau

<table>
<thead>
<tr>
<th>Row</th>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>-2.00</td>
<td>-3.00</td>
<td>-M</td>
<td>-M</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.25</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>3.00</td>
<td>-1.00</td>
<td>1.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
<td>10.00</td>
</tr>
</tbody>
</table>

Pivot 1

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th>ratio</th>
<th>ero</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2m - 2</td>
<td>-M</td>
<td>-M</td>
<td>3M</td>
<td></td>
<td>30M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.25</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.00</td>
<td></td>
</tr>
</tbody>
</table>

Row 2 divided by 3

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2M-3</td>
<td>-M</td>
<td>-M</td>
<td></td>
<td></td>
<td></td>
<td>60+10M</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.00</td>
<td></td>
</tr>
</tbody>
</table>

Row 3 - Row 2

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2M-3</td>
<td>-M</td>
<td>-M</td>
<td></td>
<td></td>
<td></td>
<td>60+10M</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.42</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.67</td>
<td>0.33</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.33</td>
<td></td>
</tr>
</tbody>
</table>

Pivot 1

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th>ratio</th>
<th>ero</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2m - 2</td>
<td>-M</td>
<td>-M</td>
<td>3M</td>
<td></td>
<td>30M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.25</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.00</td>
<td></td>
</tr>
</tbody>
</table>

Row 2 divided by 3

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2M-3</td>
<td>-M</td>
<td>-M</td>
<td></td>
<td></td>
<td></td>
<td>60+10M</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.50</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.00</td>
<td></td>
</tr>
</tbody>
</table>

Row 1 - 0.25*Row 2

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2M-3</td>
<td>-M</td>
<td>-M</td>
<td></td>
<td></td>
<td></td>
<td>60+10M</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.33</td>
<td>0.67</td>
<td>-33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.67</td>
<td>0.33</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.33</td>
<td></td>
</tr>
</tbody>
</table>

Row 3 - 1.00*Row 2

<table>
<thead>
<tr>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>rhs</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>2M-3</td>
<td>-M</td>
<td>-M</td>
<td></td>
<td></td>
<td></td>
<td>60+10M</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.42</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.67</td>
<td>0.33</td>
<td>-0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.33</td>
<td></td>
</tr>
</tbody>
</table>
### 4.10 – The Big M Method

<table>
<thead>
<tr>
<th>Pivot</th>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>RHS</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>-1/3</td>
<td>-1/3</td>
<td>-2/3</td>
<td>-1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3/3</td>
</tr>
<tr>
<td>1</td>
<td>0.42</td>
<td>1.00</td>
<td>0.08</td>
<td>-0.08</td>
<td>2.33</td>
<td>5.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td>-0.33</td>
<td>0.33</td>
<td>6.67</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pivot</th>
<th>z</th>
<th>x1</th>
<th>x2</th>
<th>s1</th>
<th>e2</th>
<th>a2</th>
<th>a3</th>
<th>RHS</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>-1/3</td>
<td>-1/3</td>
<td>-2/3</td>
<td>-1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3/3</td>
</tr>
<tr>
<td>1</td>
<td>0.42</td>
<td>1.00</td>
<td>0.08</td>
<td>-0.08</td>
<td>2.33</td>
<td>5.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>1</td>
<td>-0.33</td>
<td>0.33</td>
<td>6.67</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimal Solution

### 4.14 – Goal Programming (GP)

The concept of GP can be applied to any modeling activity, linear or otherwise. The GP concept helps find a compromise solution when no solution is possible and/or no meaningful objective exists. Some special software exist to solve GPs, but LINDO is sufficient as shown in example 9.
4.14 – Goal Programming (GP)

TWO MAJOR FORMS OF GP

1. Regular GP: if the decision maker can actually put a value or price on each unit deviation from the goal.

2. Pre-emptive GP: if the above is not possible. Then, we have a case in which higher priority goal is so much more important than the next goal in priority list that the lower level goal can not be met unless the higher level one is met.

WHAT TO MINIMIZE and WHEN?

- Minimize the negative deviation if the constraint is of GE (greater than or equal) kind or it sounds like we want to exceed the RHS value.
- Minimize the positive deviation if the constraint is of LE (less than or equal) kind or it sounds like we do not want to exceed the RHS value.
- Minimize both deviations if the constraint is of equality kind.