Linear Programming
Part 1: Formulation and Modeling (Cont.)

Jeffrey Kantor

August 28, 2006
Announcements

- Enrollment and Registration - Last Day for Class Change is August 30th.
- Homework set 1 was handed out today. A copy is available on the course web page http://jkantor.blogspot.com
Today’s Agenda

1. Revisit Giapetto to add some detail.
2. Demonstrate solution using Excel Solver.
3. Model and solve in Xpress
4. What do we seek from a mathematical analysis?
5. If time permits, introduce a simple process application.
Giapetto’s Workshop produces two types of wooden toys:

- **Soldiers** - Each sells for $27, and requires
  - $10 of raw materials, and $14 of labor and overhead
  - 2 hours of finishing labor, and 1 hour of carpentry labor

- **Train** - Each sells for $21, and requires
  - $9 of raw materials, and $10 of labor and overhead
  - 1 hour of finishing labor, and 1 hour of carpentry labor

- **Constraints**
  - 100 finishing hours, 80 carpentry hours available weekly.
  - At most 40 toy soldiers will be sold each week.

What is the maximum achievable weekly profit?

---

Example from Winston, *Operations Research*
### Decision Variables

What are the decision variables?

<table>
<thead>
<tr>
<th>Var</th>
<th>Description</th>
<th>Units</th>
<th>Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>Production of toy soldiers</td>
<td>[units/week]</td>
<td>$\geq 0$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>Production of toy trains</td>
<td>[units/week]</td>
<td>$\geq 0$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>Raw materials used</td>
<td>[$/week]</td>
<td>$\geq 0$</td>
</tr>
<tr>
<td>$x_4$</td>
<td>Finishing labor used</td>
<td>[hours/week]</td>
<td>$\geq 0$</td>
</tr>
<tr>
<td>$x_5$</td>
<td>Carpentry labor used</td>
<td>[hours/week]</td>
<td>$\geq 0$</td>
</tr>
<tr>
<td>$x_6$</td>
<td>Revenue</td>
<td>[$/week]</td>
<td></td>
</tr>
<tr>
<td>$x_7$</td>
<td>Expense</td>
<td>[$/week]</td>
<td></td>
</tr>
<tr>
<td>$x_8$</td>
<td>Profit</td>
<td>[$/week]</td>
<td></td>
</tr>
</tbody>
</table>
Model Formulation

Objective: \( \text{max } x_8 \)

Model:

\[
\begin{align*}
x_3 &= 10x_1 + 9x_2 \\
x_4 &= 2x_1 + x_2 \\
x_5 &= x_1 + x_2 \\
x_6 &= 27x_1 + 21x_2 \\
x_7 &= x_3 + 4x_4 + 6x_5 \\
x_8 &= x_6 - x_7
\end{align*}
\]
How are decisions constrained?

In addition to non-negativity and model constraints, decisions are subject to:

Demand Constraint:

\[ x_1 \leq 40 \]

Resource Constraints:

\[ x_4 \leq 100 \]
\[ x_5 \leq 80 \]
<table>
<thead>
<tr>
<th><strong>Objective Function</strong></th>
<th>Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit</td>
<td>180</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Decision Variables</strong></th>
<th>Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soldiers Produced</td>
<td>20</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>Trains Produced</td>
<td>60</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Raw Materials</td>
<td>740</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Finishing Labor</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Carpentry Labor</td>
<td>80</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>Revenue</td>
<td>1800</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Expense</td>
<td>1620</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Profit</td>
<td>180</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Equality Constraints</strong></th>
<th>Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Material Cost</td>
<td>740</td>
<td>740</td>
<td></td>
</tr>
<tr>
<td>Finishing Labor Hours</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Carpentry Labor Hours</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Revenue</td>
<td>1800</td>
<td>1800</td>
<td></td>
</tr>
<tr>
<td>Expense</td>
<td>1620</td>
<td>1620</td>
<td></td>
</tr>
<tr>
<td>Profit</td>
<td>180</td>
<td>180</td>
<td></td>
</tr>
</tbody>
</table>
Microsoft includes a sophisticated solver for LP's in modern releases of Excel, however it is not typically not installed by default. Thus you may need to carry out the following one-time tasks if "Solver" doesn't appear under the Tools menu:

1. In Excel, select menu option Tools ¿ Add-Ins...
2. From the dialog box, select the Solver Add-In

The "Giapetto's Workshop" example is available for download from the course web site.
Structure of a Simple Mosel Program

model "Giapetto’s Workshop" uses "mmxprs";

declarations
    x1, x2: mpvar ! Declare Decision Vars
end-declarations

x1 >= 0; x2 >= 0 ! Non-negativity
x1 <= 40 ! Demand
2*x1 + x2 <= 100 ! Finishing Labor
x1 + x2 <= 80 ! Carpentry Labor

maximize(3*x1 + 2*x2)

writeln(" Profit: ", getobjval)
writeln(" Soldiers: ", getsol(x1))
writeln(" Trains: ", getsol(x2))
end-model
The complete Giapetto’s Workshop example can be downloaded from the course web site. This was coded using very elementary Mosel constructions, more succinct constructions are possible, and desirable.
Demonstrations

- Unboundedness
- Infeasibility
- Sensitivity Analysis
Mathematical Formulation

Up to this point we have been emphasizing the formulation of simple linear models that involve

- A linear objective
- A set of decision variables
- Possible lower and upper bounds on the decision variables
- Linear equality constraints expressing the relationships among variables.
- Linear inequality constraints expressing resource or other operational bounds.

Let’s establish a ”Standard Form” for expressing a Linear Program
As we saw from the "Giapetto’s Workshop" example, a given problem can be written in many forms, with varying definition and number of decision variables, constraints, and even objective functions. Is there a "Standard Form" for linear programs?

Edgar, Himmelblau, and Lasdon:

\[ \max f = \sum_{j=1}^{n} c_j x_j \]

Subject to:

\[ \sum_{j=1}^{n} a_{ij} x_j = b_i; \; i = 1, 2, \ldots, m \]

\[ l_j \leq x_j \leq u_j; \; j = 1, 2, \ldots, n \]
Matrix form

Minimize:

\[ f = cx \]

Subject to:

\[ Ax = b \]
\[ l \leq x \leq u \]

where \( A \) is an \( m \times n \) matrix, \( b \) is an \( m \times 1 \) vector of constraints, \( c \) is a \( 1 \times n \) vector of objective coefficients, and \( l \) and \( u \) are \( n \times 1 \) vectors of bounds on the decision variables.
What did we learn today?

- Formulation of Linear Programs
- Computational solution of linear programs using spreadsheets and modeling languages.
- Standard form for a linear program.
Next time we will be begin to examine linear programming in earnest from an mathematical point of view. We wish to establish a sound understanding of the principals behind constrained optimization, the simplex methods, and algorithmic pitfalls.