Course Objectives

- Survey of optimization models and formulations, with focus on modeling, not on algorithms
- Include a variety of applications, such as, industrial, mechanical, civil and electrical engineering, financial optimization models, health care systems, environmental ecology, and forestry
- Include many types of optimization models, such as, linear programming, integer programming, quadratic assignment problem, nonlinear convex problems and black-box models
- Include many common formulations, such as, facility location, vehicle routing, job shop scheduling, flow shop scheduling, production scheduling (min make span, min max lateness), knapsack/multi-knapsack, traveling salesman, capacitated assignment problem, set covering/packing, network flow, shortest path, and max flow.
Each topic is an introduction to what could be a complete course:

1. basic linear models (LP) with sensitivity analysis
2. integer models (IP), such as the assignment problem, knapsack problem and the traveling salesman problem
3. mixed integer formulations
4. quadratic assignment problems
5. include uncertainty with chance-constraints, stochastic programming scenario-based formulations, and robust optimization
6. multi-objective formulations
7. nonlinear formulations, as often found in engineering design
8. brief introduction to constraint logic programming
9. brief introduction to dynamic programming
- Catalyst Tools (https://catalyst.uw.edu/)
- AIMMS - optimization software (http://www.aimms.com/)
  Ming Fang - AIMMS software consultant
Mathematical programming refers to “programming” as a “planning” activity: as in

- linear programming (LP)
- integer programming (IP)
- mixed integer linear programming (MILP)
- non-linear programming (NLP)

“Optimization” is becoming more common, e.g. the Mathematical Programming Society (MPS) just changed their name to the Mathematical Optimization Society - but are keeping the name of their flagship journal *Mathematical Programming*
Optimization models are typically categorized by the mathematical structure of the functions, which is necessary for algorithm development.

There is a connection between the models and the algorithms:
- a nonlinear integer model may be very accurate, but impractical to solve in reasonable time
- be aware of the assumptions and implications of the model

The process of creating and exploring an optimization model is often more useful than a single solution.
The standard optimization formulation consists of:

- decision variables
- objective function
- constraints

Formulation

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_j(x) \leq b_j, \quad j = 1, \ldots, m_1 \\
& \quad h_j(x) = 0, \quad j = 1, \ldots, m_2 \\
& \quad x_i \geq 0, \quad i = 1, \ldots, n \\
& \quad x_1, x_2 \in \{0, 1\} \quad (\text{binary variables})
\end{align*}
\]
Terminology

1. **Feasible solution**: a vector \( x \in \mathbb{R}^n \) that satisfies all constraints
2. **Feasible region**: the set of all feasible solutions
3. **Optimal solution**: a feasible solution \( x^* \) such that 
   \[ f(x^*) \leq f(x) \quad \text{for all feasible solutions} \quad x \]
4. **Optimal value**: \( f^* = f(x^*) \).

Existence:

- Does a feasible solution always exist? NO
- If the feasible region is non-empty, does an optimal solution always exist? NO
Existence of an optimal solution

Example (Unbounded feasible region)
Maximize $x$, for $x \geq 0$

Example (Open feasible region)
Minimize $1/x$, for $x \in (0,1)$

Theorem (Weierstrass Theorem)
A continuous function $f$ defined on a compact (closed and bounded) feasible region $S$ HAS a minimum point $x^* \in S$ such that $f(x) \geq f(x^*)$ for all $x \in S$. 
Terminology (Linear Program)

A linear program (LP) is a linear optimization model: minimize a linear objective function subject to linear equality and linear inequality constraints.

Example

minimize $2x_1 - x_2 + 4x_3$

subject to $x_1 + x_2 + x_4 \leq 2$
$3x_2 - x_3 = 5$
$x_3 + x_4 \geq 3$
$x_1 \geq 0$
$x_3 \leq 0$
A Standard Linear Program

Formulation

minimize \( cx \)
subject to \( Ax = b \)
\( x \geq 0 \)

where \( x \in R^n \), \( A \) is an \( m \times n \) matrix, \( c \) is \( 1 \times n \), and \( b \) is \( m \times 1 \) for some integers \( m \) and \( n \).

Remark

Every linear program can be converted into an equivalent problem with equality constraints and non-negativity constraints.

What does “equivalent problem” mean? Problem P is equivalent to Problem Q, if, for every feasible \( x \in P \), there exists a feasible \( y \in Q \), and vice versa, and for every optimal \( x^* \in P \), there exists an optimal \( y \in Q \), and vice versa.
1. **Minimize or Maximize - that is the question** Minimizing $cx$ is equivalent to maximizing $-cx$. Is this true for $f(x)$? YES, minimizing $f(x)$ is equivalent to maximizing $-f(x)$.

2. **Eliminate Free Variables** For variables $x_j$ that are not restricted to be non-negative or non-positive, write $x_j = x_j^+ - x_j^-$, $x_j^+ \geq 0$, $x_j^- \geq 0$. Does this trick work for nonlinear problems? ALMOST, need to add $(x_j^+) \cdot (x_j^-) = 0$.

3. **Eliminate Inequality Constraints**
   - Modify every $\geq$ inequality constraint by introducing a surplus variable $s_i \geq 0$ to write $a'_i x - s_i = b_i$.
   - Modify every $\leq$ inequality constraint by introducing a slack variable $s_i \geq 0$ to write $a'_i x + s_i = b_i$. 
Two Equivalent Linear Programs

Example

minimize $2x_1 + 4x_2$

$\begin{align*}
x_1 + x_2 & \geq 3 \\
3x_1 + 2x_2 & = 14 \\
x_1 & \geq 0
\end{align*}$

minimize $2x_1 + 4x_2^+ - 4x_2^-$

$\begin{align*}
x_1 + x_2^+ - x_2^- - s_1 & = 3 \\
3x_1 + 2x_2^+ - 2x_2^- & = 14 \\
x_1, x_2^+, x_2^-, s_1 & \geq 0
\end{align*}$
minimize $-x_1 - x_2$

subject to

$x_1 + 2x_2 \leq 3$

$2x_1 + x_2 \leq 3$

$x_1, x_2 \geq 0$
Example (Product Mix Problem)

An engineering factory can produce five types of product (PROD1, PROD2, . . . , PROD5) by using two processes: grinding and drilling. We are given data on:

- How much each unit of each product contributes to profit
- How much time each unit of each product requires on each process (grinding and drilling)
- How much employee’s time is needed for final assembly of each unit of each product
- Availability of grinding machines, drilling machines, and employees

The problem is to find how much to make of each product so as to maximize the total profit contribution.
Example

<table>
<thead>
<tr>
<th></th>
<th>PROD1</th>
<th>PROD2</th>
<th>PROD3</th>
<th>PROD4</th>
<th>PROD5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit</td>
<td>550</td>
<td>600</td>
<td>350</td>
<td>400</td>
<td>200</td>
</tr>
<tr>
<td>Grinding</td>
<td>12</td>
<td>20</td>
<td>–</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>Drilling</td>
<td>10</td>
<td>8</td>
<td>16</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Each unit of each product uses 20 hours of an employee’s time in the final assembly. The factory has three grinding machines and two drilling machines and works a six-day week with two shifts of 8 hours on each day. Eight workers are employed in assembly, each working one shift a day.

Decision variables:

\( x_i \) is units of product \( i \) to be produced every week, \( i = 1, \ldots, 5 \).

Objective function:

Profit per week = \( 550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5 \)
Example: Product Mix LP Model

Formulation

minimize \[ 550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5 \]

subject to

\[ 12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 288 \]
\[ 10x_1 + 8x_2 + 16x_3 \leq 192 \]
\[ 20x_1 + 20x_2 + 20x_3 + 20x_4 + 20x_5 \leq 384 \]

\[ x_1, x_2, x_3, x_4, x_5 \geq 0 \]
Example: Product Mix LP in AIMMS Software

Example

- Indices: Types of product \( i \), \( i = 1, \ldots, 5 \)
- Variables: Number of product of type \( i \) per week
- Parameters:
  - \( P_i \) Profit contribution of product \( i \)
  - \( G_i \) Grinding time for product \( i \)
  - \( D_i \) Drilling time for product \( i \)
  - \( M \) Manpower time for product \( i \)

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{5} P_i x_i \\
\text{subject to} & \quad \sum_{i=1}^{5} G_i x_i \leq 288 \\
& \quad \sum_{i=1}^{5} D_i x_i \leq 192 \\
& \quad \sum_{i=1}^{5} M x_i \leq 384 \\
& \quad x \geq 0
\end{align*}
\]
Notes 2: Blending and other Linear Models

IND E 599

October 4, 2010
A blending problem is another common LP found in:

- chemical processing industry
- oil refinery
- animal feed
- pharmaceuticals

A characteristic of a blending problem is using proportions with a type of balance constraint.
Example (Blending Problem)
A food is manufactured by refining raw oils and blending them together. The raw oils come in two categories:

- Vegetable oils: VEG1 and VEG2
- Non-vegetable oils: OIL1, OIL2 and OIL3

In any month it is not possible to refine more than 200 tons of vegetable oils and more than 250 tons of non-vegetable oils. **ASSUME:** there is no loss of weight in the refining process.

There is a technological restriction on hardness. In the units in which hardness is measured, this must lie between 3 and 6. **ASSUME:** hardness blends linearly.
Example

The final product sells for $150 per ton.

<table>
<thead>
<tr>
<th></th>
<th>VEG1</th>
<th>VEG2</th>
<th>OIL1</th>
<th>OIL2</th>
<th>OIL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost (per ton)</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>110</td>
<td>115</td>
</tr>
<tr>
<td>Hardness</td>
<td>8.8</td>
<td>6.1</td>
<td>2.0</td>
<td>4.2</td>
<td>5.0</td>
</tr>
</tbody>
</table>

How should the food manufacturer make their product in order to maximize their net profit?

Decision variables:

\( x_i \) is the tons of oil \( i \) (\( i = 1, \ldots, 5 \) for VEG1, VEG2, OIL1, OIL2, OIL3) bought, refined and blended per month.

\( y \) is the total quantity (tons) of product made per month.

Equality (Balance) Constraint:

\[
x_1 + x_2 + x_3 + x_4 + x_5 = y
\]
Objective Function
Max Profit \(-110x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5 + 150y\)

Capacity Constraints
Refining capacity on vegetable oil (200 tons/month)
\(x_1 + x_2 \leq 200\)
Refining capacity on non-vegetable oil (250 tons/month)
\(x_3 + x_4 + x_5 \leq 250\)

Hardness Constraints
\(8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 \leq 6y\)
\(8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 \geq 3y\)

Non-negativity Constraints
\(x_1, x_2, x_3, x_4, x_5 \geq 0\quad y \geq 0\)

The non-negativity constraint on \(y\) is redundant, and in fact, \(y\) could be removed by substituting the Balance equality.
Example (Use slack variable for upper and lower bound)

8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 \leq 6y
8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 \geq 3y

An equivalent way:

8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 + s = 6y
s \geq 0 \text{ and } s \leq 3y

Now, substitute balance equation to eliminate \( y \):

8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_5 + s = 6(x_1 + x_2 + x_3 + x_4 + x_5)
2.8x_1 + 0.1x_2 - 4x_3 - 1.8x_4 - 1x_5 + s = 0
s \geq 0 \text{ and } s \leq 3y

Example (A better example (variables are the same))

7 \leq 5x_1 + 3x_2 \leq 10
5x_1 + 3x_2 + s = 10, \quad s \geq 0, s \leq 3
Example

There are $t$ batches of chemicals available. The following data is available:

- $w_k =$ weight (tons) of material in the $k$th batch;
- $a_{k1}, a_{k2}, \ldots, a_{km}$ (with $\sum_{i=1}^{m} a_{ki} = 1$) is the fraction of material (by weight) in the $k$th batch in the various particle sizes $i$;
- $b_1, b_2, \ldots, b_m$ (with $\sum_{i=1}^{m} b_i = 1$) is the desired fraction of blend (by weight) in the various particle sizes $i$;
- $W =$ weight in tons of the blend to be produced.

It is desired to blend material from the various batches, so as to produce a blend with fractional size-weight distribution as close to the desired as possible.
Example (Fraction by Weight of Various Particle Sizes ($a_{ki}$))

<table>
<thead>
<tr>
<th>Batch ($k$)</th>
<th>Size1</th>
<th>Size2</th>
<th>Size3</th>
<th>Size4</th>
<th>Tons Avail ($w_k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10</td>
<td>0.40</td>
<td>0.30</td>
<td>0.20</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>0.10</td>
<td>0.50</td>
<td>0.20</td>
<td>250</td>
</tr>
<tr>
<td>3</td>
<td>0.30</td>
<td>0.58</td>
<td>0.02</td>
<td>0.10</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>0.40</td>
<td>0.15</td>
<td>0.30</td>
<td>0.15</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>0.60</td>
<td>0.17</td>
<td>0.18</td>
<td>0.05</td>
<td>200</td>
</tr>
<tr>
<td>Desired fraction in blend ($b_i$)</td>
<td>0.45</td>
<td>0.30</td>
<td>0.15</td>
<td>0.10</td>
<td>800 tons required ($W$)</td>
</tr>
</tbody>
</table>

Decision variables:

$x_k$ is the tons of batch $k$, $k = 1, \ldots, 5$ used in the final blend.
Constraints

Desired Blending Constraints:

\[
\begin{align*}
0.10x_1 + 0.20x_2 + 0.30x_3 + 0.40x_4 + 0.60x_5 + b_-^1 - b_+^1 &= 0.45 \\
0.40x_1 + 0.10x_2 + 0.58x_3 + 0.15x_4 + 0.17x_5 + b_-^2 - b_+^2 &= 0.30 \\
0.30x_1 + 0.50x_2 + 0.02x_3 + 0.30x_4 + 0.18x_5 + b_-^3 - b_+^3 &= 0.15 \\
0.20x_1 + 0.20x_2 + 0.10x_3 + 0.15x_4 + 0.05x_5 + b_-^4 - b_+^4 &= 0.10
\end{align*}
\]

Available material:

\[x_1 \leq 100, \; x_2 \leq 250, \; x_3 \leq 50, \; x_4 \leq 500, \; x_5 \leq 200\]

Required tons:

\[x_1 + x_2 + x_3 + x_4 + x_5 = 800\]

Non-negativity constraints:

\[x_k \geq 0, \; k = 1, \ldots, 5, \; b_i^+, b_i^- \geq 0, \; i = 1, \ldots, 4\]

and \( (b_i^+) \cdot (b_i^-) = 0 \)
Objective

Produce the blend with fractional size-weight distribution as close to the desired as possible:

\[ \min \sum_{i=1}^{4} b_i^+ + b_i^- \]

What about, minimize the maximum deviation from the desired fraction in the blend over all the sizes:

\[ \min \left( \max \{ b_1^+, b_1^-, \ldots, b_4^+, b_4^- \} \right) \]
Trick to Minimize the Maximum

We want to minimize $f(x)$ where

$$f(x) = \max\{b_1^+, b_1^-, \ldots, b_4^+, b_4^-\}$$

Introduce a new variable $z$, and force $z$ to be bigger than any of the terms in the maximum:

$$z \geq b_1^+, \quad z \geq b_1^-, \quad \ldots, \quad z \geq b_4^+, \quad z \geq b_4^-,$$

and minimize $z$.

Mini-max

This trick works for any expression in the maximum! It is commonly used to minimize makespan in production scheduling problems. Also in conservative financial investments, to minimize the maximum risk. My research used minimize maximum margin-of-safety in design of optimal composite structures.
We want to solve:

\[
\begin{align*}
\text{Minimize} & \quad \max_i \left\{ \sum_j a_{ij}x_j \right\} \\
\text{subject to} & \quad \text{conventional linear constraints}
\end{align*}
\]

Introduce a new variable \( z \):

\[
\begin{align*}
\text{Minimize} & \quad z \\
\text{subject to} & \quad \text{conventional linear constraints} \\
& \quad \sum_j a_{ij}x_j - z \leq 0 \quad \text{for all}\ i \\
& \quad z \geq 0
\end{align*}
\]
Other situations that can be converted to LPs

- Ratio of linear functions
- Piecewise linear functions
- Absolute value functions
Common Objectives

- Minimize cost
- Maximize profit
- Maximize utility
- Maximize return on investment
- Maximize net present value
- Maximize customer satisfaction
- Maximize probability of achieving a goal
- Maximize reliability
- Minimize the change in employment (hiring/firing)
- Maximize robustness of a strategy or plan
- Minimize makespan
- others?
Common Constraints

- Capacity constraints
- Demand constraints
- Balance constraints
- Flow constraints
- Bounding constraints (upper & lower bounds, also mini-max)
- Hard and soft constraints (use $b_i^+$ and $b_i^-$ to “soften” a constraint and penalize deviations - also goal programming)
- Use fuzzy set membership instead of soft constraints - and convert to LP using mini-max objective
- Chance constraints $P(\sum a_i x_i \leq B) \geq \beta$
- Either-or constraints (use binary variables)
- Logic constraints (Produce product 1 if product 2 is produced but neither products 3 or 4 are produced)
How to Build a Good Model (Section 3.4)

1. Ease of understanding the model
2. Ease of detecting errors in the model
3. Ease of computing the solution
Ease of understanding the model

- state assumptions
- clearly document the model
- clearly define the decision variables
- clearly state the objectives and constraints
- document variations on model
- include details such as units, sources of data
Ease of detecting errors in the model

- clerical errors and formulation errors
- build modular formulations
- if using substitutions and compact formulations, test equivalency in small problems
- develop test cases that are scalable
- verify that trends follow intuitive explanations
- check for reasonable ranges on feasibility, boundedness
- compare to historical solutions
- get feedback from decision makers on reasonableness
Ease of computing the solution

- consider tradeoffs between accuracy of model and computational time
- test different solution techniques and compare
- be alert to numerical error
- debug a model as you would a computer program
Interpreting and using an LP (Chapter 6)

- Infeasible models
- Unbounded models
- Solvable models
Example

\[ x_1 - x_2 \geq 1 \]
\[ x_2 - x_3 \geq 1 \]
\[ x_1 + x_3 \geq 1 \]

There is a mutual incompatibility of all three constraints, yet, it may be wrong to identify any single one of the three as being incorrect.
Example

Maximize $x_1 + x_2$
subject to $2x_1 + x_2 \geq 1$

This is unbounded because there is no limit to how large we can make $x_1 + x_2$. Consider missing physical constraints, or missing balance constraints.
Solvable Models

Does the solution make sense?
Check trends:
If capacity is increased, does the solution make sense?
If capacity is decreased, does the solution make sense?
If costs are modified, does the solution make sense?
Compare to practical solutions

Example
In testing optimal design of composite structures, started with a simple flat plate, then extended to large panel. Physically tested a 30 ft by 10 ft panel that was built according to optimal design. The panel failed in a mode that was not predicted, implying the analysis for the constraints was insufficient to accurately predict performance. The analysis was extended and a new model was developed.
Notes 3: Economic Interpretations

IND E 599

October 6, 2010
Common terms relating to economic interpretations: shadow prices, sensitivity analysis, duality

Easiest interpretation when in the following form:

\[
\begin{align*}
\text{minimize} & \quad cx \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\]

where \(x \in \mathbb{R}^n\), \(A\) is an \(m \times n\) matrix, \(c\) is \(1 \times n\), and \(b\) is \(m \times 1\) for some integers \(m\) and \(n\).
Example (Product Mix Example)

$x_i$ is units of product $i$ to be produced every week, $i = 1, \ldots, 5$

Maximize profit, subject to constraints on grinding, drilling and manpower time

$$\text{maximize} \quad 550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5$$

$$\text{subject to} \quad 12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 288$$
$$10x_1 + 8x_2 + 16x_3 \leq 192$$
$$20x_1 + 20x_2 + 20x_3 + 20x_4 + 20x_5 \leq 384$$

$x_1, x_2, x_3, x_4, x_5 \geq 0$
The optimal solution to the example product mix problem is

\[ x_1^* = 12, \quad x_2^* = 7.2, \quad x_3^* = x_4^* = x_5^* = 0 \]

with optimal value

\[ Profit^* = 10,920 \]
When investigating the optimal solution, the grinding and manpower constraints are binding (also called active, or holding at equality, or have no slack), whereas there is slack in the drilling capacity

\[ x_1^* = 12, \quad x_2^* = 7.2, \quad x_3^* = x_4^* = x_5^* = 0 \]

\[
\begin{align*}
12x_1^* &+ 20x_2^* + 0 + 0 = 288 \\
10x_1^* &+ 8x_2^* + 0 = 177.6 \leq 192 \\
20x_1^* &+ 20x_2^* + 0 + 0 + 0 = 384
\end{align*}
\]
Why are only 2 products being made? Are products 3, 4, and 5 underpriced? What should their price be in order to make it worthwhile to manufacture them?

Why is there extra drilling capacity? Should I pay to increase grinding and manpower capacity to increase profit?

What is the marginal value of these capacities?
Extra Economic Information

- **Reduced costs**: Each variable has an associated reduced cost, or opportunity cost, which is the amount by which an objective function coefficient would have to improve (so increase for maximization problem, decrease for minimization problem) before it would be possible for its corresponding variable to assume a positive value in the optimal solution.

- **Shadow price**: Each constraint has an associated shadow price which is the change in the objective value of the optimal solution obtained by relaxing the constraint by one unit – it is the marginal utility of modifying the constraint.

- More formally, the shadow price is the value of the Lagrange multiplier at the optimal solution, which means that it is the infinitesimal change in the objective function arising from an infinitesimal change in the constraint.

- The extra economic information arises from the dual model.
Shadow prices:
▶ Grinding Constraint: 6.25
▶ Drilling Constraint: 0
▶ Manpower Constraint: 23.75

The shadow price for Drilling is zero, because it is not binding. So there is no value to increasing this capacity.

The shadow price for Manpower is much larger than for Grinding, so it is more advantageous to add 1 unit of Manpower than 1 unit of Grinding (if they have equal cost), because 1 unit of Manpower would increase profit by 23.75, whereas Grinding would increase profit by 6.25.

The shadow prices are for small changes, performed one at a time. How large a change can be made? What if more than one change is made?
When Interpreting Shadow Prices

- How large a change in resource can be made without losing the shadow price interpretation?
  Right-hand side (rhs) Ranging
- What if more than one change is made at a time?
  Sensitivity Analysis and Parametric Programming
### RHS Ranging for the Product Mix Example

<table>
<thead>
<tr>
<th></th>
<th>Shadow Price</th>
<th>Current Value</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grinding</td>
<td>6.25</td>
<td>288</td>
<td>230.4</td>
<td>384</td>
</tr>
<tr>
<td>Drilling</td>
<td>0</td>
<td>192</td>
<td>177.6</td>
<td>infinity</td>
</tr>
<tr>
<td>Manpower</td>
<td>23.75</td>
<td>384</td>
<td>288</td>
<td>406.1</td>
</tr>
</tbody>
</table>

- If we added 1 hour of grinding time, we would increase profit by 6.25. We could add as much as 96 (=384-288) hours, and gain $96 \times 6.25 = 600$.

- If we added 1 hour of manpower time, we would increase profit by 23.75. We could add as much as 22.1 (=406.1-384) hours, and gain $22.1 \times 23.75 = 524.875$.

- In either situation, the amount produced will change.
To visualize right-hand side ranging and how the optimal solution and objective value change as the rhs changes, consider a two-variable problem:

**Example**

\[
\begin{align*}
\text{maximize} \quad & 3x_1 + 2x_2 \\
\text{subject to} \quad & x_1 + x_2 \leq 4 \\
& 2x_1 + x_2 \leq 5 \\
& -1x_1 + 4x_2 \geq 2 \\
& x_1, x_2 \geq 0
\end{align*}
\]

(Draw Figure 6.3 on board.)
## RHS Ranging and Solutions for the Product Mix Example

<table>
<thead>
<tr>
<th>Current RHS</th>
<th>Optimal Solution</th>
<th>Optimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grinding 288</td>
<td>$x_1^* = 12$</td>
<td>10,920</td>
</tr>
<tr>
<td>Drilling 192</td>
<td>$x_2^* = 7.2$</td>
<td></td>
</tr>
<tr>
<td>Manpower 384</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Grinding RHS</th>
<th>Optimal Solution</th>
<th>Optimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grinding 384</td>
<td>$x_2^* = 19.2$</td>
<td>11,520</td>
</tr>
<tr>
<td>Drilling 192</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manpower 384</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Max Manpower RHS</th>
<th>Optimal Solution</th>
<th>Optimal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grinding 288</td>
<td>$x_1^* = 14.8$</td>
<td>11,446</td>
</tr>
<tr>
<td>Drilling 192</td>
<td>$x_2^* = 5.5$</td>
<td></td>
</tr>
<tr>
<td>Manpower 406.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notice the extreme solutions, at most two products are made. Is it ever optimal to make three different products? More than three?
Sensitivity Analysis

Change capacity constraint for Manpower from 384 to 410

Example (Change rhs for Manpower)

\[
\begin{align*}
\text{maximize} & \quad 550x_1 + 600x_2 + 350x_3 + 400x_4 + 200x_5 \\
\text{subject to} & \quad 12x_1 + 20x_2 + 25x_4 + 15x_5 \leq 288 \\
& \quad 10x_1 + 8x_2 + 16x_3 \leq 192 \\
& \quad 20x_1 + 20x_2 + 20x_3 + 20x_4 + 20x_5 \leq 410 \\
& \quad x_1, x_2, x_3, x_4, x_5 \geq 0
\end{align*}
\]

The optimal solution to this version of the product mix problem is

\[x_1^* = 15.48, \quad x_2^* = 4.65, \quad x_3^* = 0, \quad x_4^* = 0.37, \quad x_5^* = 0\]

with optimal value

\[\text{Profit}^* = 11,451.85\]
It is possible for three products to be in the optimal solution. However, there will never be more than three non-zero products because there are only three constraints.

In a linear model with $m$ constraints, there will never be more than $m$ non-zero variables in an optimal basic (extreme point) solution.

In a nonlinear model, the optimal solution is not always an extreme point solution (may be interior).
Reduced Costs and Objective Function Ranging

For the product mix example:

<table>
<thead>
<tr>
<th>Current Obj Function $c_i$</th>
<th>Reduced Cost</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1 = 550$</td>
<td></td>
<td>500</td>
<td>600</td>
</tr>
<tr>
<td>$c_2 = 600$</td>
<td></td>
<td>550</td>
<td>683.3</td>
</tr>
<tr>
<td>$c_3 = 350$</td>
<td>125</td>
<td>-infinity</td>
<td>475</td>
</tr>
<tr>
<td>$c_4 = 400$</td>
<td>231.25</td>
<td>-infinity</td>
<td>631.25</td>
</tr>
<tr>
<td>$c_5 = 200$</td>
<td>368.75</td>
<td>-infinity</td>
<td>568.75</td>
</tr>
</tbody>
</table>

Interpretation for Product 3: We would have to increase its unit price by 125 (from $c_3 = 350$ to 475), to make it (just) worth making.
<table>
<thead>
<tr>
<th>Obj. Value $c_3$</th>
<th>Optimal Solution $x_1^* = 12$</th>
<th>10,920</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_3 = 350$</td>
<td>$x_2^* = 7.2$</td>
<td></td>
</tr>
<tr>
<td>$c_3 = 475$</td>
<td>$x_2^* = 14.4$</td>
<td>10,920</td>
</tr>
<tr>
<td></td>
<td>$x_3^* = 4.8$</td>
<td></td>
</tr>
</tbody>
</table>
To visualize objective function values ranging and how it can impact multiple optima, consider a two-variable problem:

**Example**

$$\text{maximize} \quad 3x_1 + 1.5x_2$$

subject to

$$x_1 + x_2 \leq 4$$
$$2x_1 + x_2 \leq 5$$
$$-1x_1 + 4x_2 \geq 2$$

$$x_1, x_2 \geq 0$$

(Draw Figure 6.1 on board.)
To visualize a degenerate solution to a linear program, consider a two-variable problem:

**Example**

\[
\begin{align*}
\text{maximize} & \quad 3x_1 + 2x_2 \\
\text{subject to} & \quad x_1 + x_2 \leq 3 \\
& \quad 2x_1 + x_2 \leq 4 \\
& \quad 4x_1 + 3x_2 \leq 10 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]

(Draw Figure 6.2 on board.)

A degenerate solution in the dual implies multiple optima in the primal, and vice versa.
The variables $y_1, y_2, y_3$ represent the valuations for each hour of each of the capacities of grinding, drilling and manpower – from an accounting perspective.

For example, the profit contribution of 550 for PROD1 is accounted for by the “value” of the 12 hours of grinding, 10 hours of drilling and 20 hours of manpower used in making one unit.

Example

\[
\begin{align*}
\text{minimize} & \quad 288y_1 + 192y_2 + 384y_3 \\
\text{subject to} & \quad 12y_1 + 10y_2 + 20y_3 \geq 550 \\
& \quad 20y_1 + 8y_2 + 20y_3 \geq 600 \\
& \quad 16y_2 + 20y_3 \geq 350 \\
& \quad 25y_1 + 20y_3 \geq 400 \\
& \quad 15y_1 + 20y_3 \geq 200 \\
\end{align*}
\]

$y_1, y_2, y_3 \geq 0$
Duality theory provides the economic interpretation of shadow prices. It also provides the straightforward calculation of rhs ranging, obj. function coefficients ranging, and sensitivity analysis.

Duality theory exists for convex programming, not just linear programming, where the dual variables are called Lagrange multipliers.
Parametric Programming

Allow several changes to occur, as a function of a single parameter.

For example, suppose we are considering hiring an extra worker for each new grinding machine bought.

Let $\theta$ be the number of grinding machines bought. Then new rhs is:

$$
\begin{pmatrix}
288 \\
192 \\
384
\end{pmatrix}
+ \theta
\begin{pmatrix}
96 \\
0 \\
48
\end{pmatrix}
$$

because each grinding machine has 96 hours per week, and each additional worker has 48 hours per week.
It is also possible to use parametric programming to explore tradeoffs in objective function coefficients.

For example, suppose we reduce the price of PROD4, but increase the price of PROD 1. Then, the objective function would be:

\[(550 + \theta)x_1 + 600x_2 + 350x_3 + (400 - \theta)x_4 + 200x_5\]
Notes 4: Integer Programming

IND E 599

October 11, 2010
Many practical problems can be modeled using integer variables and linear constraints:

- The variables can only take on discrete values, such as number of trucks, airplanes, employees
- Binary variables 0–1 represent yes–no decisions, or other logical connections between decisions and constraints
Example (Integral Units)

$x_i$ is number of units of product $i$, integer valued, $i = 1, 2$

\[
\begin{align*}
\text{maximize} & \quad x_1 + x_2 \\
\text{subject to} & \quad -2x_1 + 2x_2 \geq 1 \\
& \quad -8x_1 + 10x_2 \leq 13 \\
& \quad x_1, x_2 \geq 0 \text{ and integer valued}
\end{align*}
\]

Optimal solution to the LP-relaxation (drop integer constraint) is $(4, 4.5)$ with objective $= 8.5$

Optimal solution to the IP is $(1, 2)$ with objective $= 3$
Graph the Example – a Thin Polytope

\[
\begin{align*}
\text{max } & \quad x_1 + x_2 \\
\text{s.t. } & \quad -2x_1 + 2x_2 \geq 1 \\
& \quad -8x_1 + 10x_2 \leq 13 \\
& \quad x_1, x_2 \geq 0
\end{align*}
\]
Many integer programs are NP-hard, or NP-complete, whereas a linear program can be solved in “weakly polynomial” time with interior-point methods (not the simplex method).

Some integer programs (most notably network type problems) can be solved in polynomial time and faster than a general LP because of the specific structure of the problem.
Common Types of Integer Programs

Common integer models, discussed later in Chapter 9.5:

- Set covering problems
- Set packing problems
- Set partitioning problems
- Knapsack problem
- Traveling salesman problem
- Quadratic assignment problem

and network integer models, discussed in Chapters 5.3 and 10:

- Transportation problem
- Assignment problem
- Minimum cost network flow problem
Integer Programming Applications

- Set-up costs or set-up times, also called fixed-charge, where thresholds exist, such as pricing in “bundles”, machine time comes in “packets”

- Sequencing problems, such as job-shop scheduling
  Also Traveling Salesman Problem is a sequencing problem

- Allocation problems, such as project selection, capital budget allocation, or product mix with variations (number of products)

- Assembly line-balancing, aircrew scheduling

- Facility location, called depot location in text

- Graph theory, such as the four-color problem
Active research on algorithms for solving IPs quickly. Some methods include:

- Branch and Bound
- Cutting Planes
- Enumeration (only practical for small problems)
- Pseudo-Boolean Methods
- Randomized (Monte Carlo) Algorithms
- Approximation Techniques
- Heuristics and Meta-heuristics
**Integer Notation**

Discrete (integer) quantities: let \( x \in \{\ldots, -1, 0, 1, \ldots\} \)

Finite sets: let \( x \in \{0, 2, 5, 10\} \)

Let \( x = \begin{cases} 
0 & \text{if no depot should be built} \\
1 & \text{if a depot of type A should be built} \\
2 & \text{if a depot of type B should be built} 
\end{cases} \)

Binary indicator variable coupled with a real-valued variable:

- \( x \) is the quantity of an ingredient to be blended
- \( \delta \) indicates whether \( x > 0 \)

Constraints include, \( x \geq 0, \delta \in \{0, 1\} \) and

\[ x - M\delta \leq 0 \]

for a large positive value \( M \)

if \( x > 0 \), then \( \delta = 1 \)

if \( x = 0 \), then \( \delta \) may be 0 or 1
The Fixed-Charge Problem

Let $x$ be the quantity of a product to be manufactured, and $C_1$ is the marginal cost per unit, and $C_2$ is the set-up cost if any product is to be manufactured.

Let $\delta = 1$ if any product is manufactured, i.e., if $x > 0$. (Draw Figure 9.1 on board.)

Total cost = $C_1x + C_2\delta$

Add constraints: $x \geq 0$, $\delta \in \{0, 1\}$ and

$x - M\delta \leq 0$ for a large positive value $M$

If $x > 0$, then $\delta = 1$, and the total cost is well-represented. If $x = 0$ and $\delta = 1$, the constraints are satisfied, however, if $x = 0$ and $\delta = 0$, the constraints are also satisfied at a lower cost.
How to choose $M$?

To ensure that the constraints:
- $x \geq 0$, $\delta \in \{0, 1\}$ and
- $x - M\delta \leq 0$ for a large positive value $M$

imply that, if $x > 0$, then $\delta = 1$,

we need $M$ to exceed any possible value of $x$. 
Suppose we would like:

- if $x > 0$, then $\delta = 1$, and
- if $x = 0$, then $\delta = 0$

then we can impose the condition

$$\delta = 1 \text{ if and only if } x > 0$$

Or, more realistically,

$$\delta = 1 \text{ if and only if } x > m, \text{ a threshold value}$$

Add constraints:

- $x \geq 0$, $\delta \in \{0, 1\}$ and
- $x - M\delta \leq 0$ for a large positive value $M$, and
- $x - m\delta \geq 0$ for the threshold value $m$

Check that both inequalities are needed.
Suppose $x_A$ represents the proportion of ingredient A to be included in a blend, and $X_B$ represents the proportion of ingredient B to be included in the same blend. Also, suppose

If A is included in the blend, then B must be included also.

Add constraints: $\delta \in \{0, 1\}$ and

$$x_A - M\delta \leq 0 \quad \text{and} \quad M = 1 \text{ is large enough because } x_A \text{ is a proportion, and}$$

$$x_B - m\delta \geq 0 \text{ with } m = 0.01 \text{ to detect presence of B.}$$
Notes 5: More on Integer Programming

IND E 599

October 18, 2010
Binary variables are very useful to capture common types of requirements, as seen with the fixed-charge model.

Another similar and common requirement is the All-or-Nothing requirement:

Example (All-or-Nothing requirement)

\[
    x_j = \begin{cases} 
    0 & \text{or} \\
    5 & \text{or} \\
    \end{cases}
\]

\[
    x_j = 5\delta \quad \text{where} \quad \delta \in \{0, 1\}
\]

can be modeled by substituting

\[
    x_j = 5\delta
\]
Examples of All-or-Nothing Situations

- Should we invest 0 or 5 million dollar?
- Should capacity remain at 0 or jump to 5 units?
- Should we start a job immediately (time 0) or wait a day?
- Should a certain site be selected for a new facility?
Example (Indicate whether an inequality holds or not)

We would like

\[ \delta = 1 \implies \sum_j a_j x_j \leq b \]

which can be modeled by

\[ \sum_j a_j x_j + M\delta \leq M + b \]

where \( \delta \in \{0, 1\} \) and \( M \) is an upper bound on \( \sum_j a_j x_j - b \)

Check: if \( \delta = 0 \), the inequality may or may not hold, but if \( \delta = 1 \), the inequality must hold.
Example (The reverse: An inequality implies $\delta = 1$)

We would like

$$\sum_j a_j x_j \leq b \implies \delta = 1$$

or equivalently

$$\delta = 0 \implies \sum_j a_j x_j \not\geq b$$

however, $\sum_j a_j x_j \not\geq b$ is the same as $\sum_j a_j x_j > b$, but we need to introduce a threshold $\epsilon$ for numerical issues, $\sum_j a_j x_j \geq b + \epsilon$, which can be modeled by

$$\sum_j a_j x_j - (m - \epsilon)\delta \geq b + \epsilon$$

where $\delta \in \{0, 1\}$ and $m$ is a lower bound on $\sum_j a_j x_j - b$
Example

Use a binary variable $\delta$ to indicate whether or not the following constraint is satisfied

$$2x_1 + 3x_2 \leq 1, \quad \text{for } 0 \leq x_1, x_2 \leq 1$$

Taking $M \geq \sum_j a_j x_j - b$, e.g. $M = 4$ and taking $m \leq \sum_j a_j x_j - b$, e.g. $m = -1$, and $\epsilon = 0.1$

this can be modeled by

$$2x_1 + 3x_2 + 4\delta \leq 5$$

$$2x_1 + 3x_2 + 1.1\delta \geq 1.1$$

Check: if $\delta = 1$, then $2x_1 + 3x_2 \leq 1$ must hold. If $\delta = 0$, then $2x_1 + 3x_2 \leq 1$ does not hold, and in fact, $2x_1 + 3x_2 \geq 1.1$
Examples of common logical conditions

- If we manufacture product A, then we must also manufacture product B or at least one of products C and D.
- If this station is closed, then both branch lines terminating at the station must also be closed.
- No more than five of the ingredients in this class may be included in the blend at any one time.
- Either task A must be finished before task B starts, or vice versa.
- If the library’s subscription to this journal is cancelled, then we must retain at least one subscription to another journal in this class.
- Choose at most 10 different investments, where each one has a minimum amount.
Notation for connectives from Boolean algebra

► ∨ means *or* (this is inclusive, i.e. A or B or both)
► · means *and*
► ~ means *not*
► ➞ or → means *implies* (e.g. If ... then)
► ⇔ or ↔ means *if and only if* (iff)

**Example**

Let $X_i$ stand for the proposition: “*Ingredient i is in the blend*” for $i = A, B, C$.

The expression $X_A \implies (X_B \lor X_C)$ means, if $X_A$ is in the blend, then ingredient B or C (or both) must also be in the blend.
If either of products A or B (or both) are manufactured, then at least one of products C, D, or E must also be manufactured.

Let \( X_i \) stand for the proposition: “Product i is manufactured” and let \( \delta_i \) be a binary variable that equals 1 iff product \( i \) is manufactured, for \( i = A, \ldots, E \).

\[
(X_A \lor X_B) \implies (X_C \lor X_D \lor X_E)
\]

Model “\( X_A \lor X_B \) holds” with the following inequality

\[
\delta_A + \delta_B \geq 1
\]

and model “\( X_C \lor X_D \lor X_E \) holds” with the following inequality

\[
\delta_C + \delta_D + \delta_E \geq 1
\]
Finally, we wish:

- $\delta_A + \delta_B \geq 1 \implies \hat{\delta} = 1$ and
- $\hat{\delta} = 1 \implies \delta_C + \delta_D + \delta_E \geq 1$

We model that with the following additional constraints:

- $\delta_A + \delta_B - 2\hat{\delta} \leq 0$
- $-\delta_C - \delta_D - \delta_E + \hat{\delta} \leq 0$
Another way to model this logical condition

\[(X_A \lor X_B) \implies (X_C \lor X_D \lor X_E)\]

is equivalent to

\[(X_A \implies (X_C \lor X_D \lor X_E)) \cdot (X_B \implies (X_C \lor X_D \lor X_E))\]

which can be modeled with

\[\delta_A - \tilde{\delta} \leq 0\]
\[\delta_B - \tilde{\delta} \leq 0\]
An example of mutually exclusive choices: Choose A or B, but not both.

A and B are mutually exclusive

Let $\delta_A$ and $\delta_B$ be binary variables representing choosing A or B, respectively. The mutually exclusiveness condition can be modeled as

$$\delta_A + \delta_B = 1$$
An example: Either $3x_1 + 2x_2 \leq 18$
or $2x_1 + 4x_2 \leq 16$

**Either A or B**

Let $\delta$ be a binary variable, and $M$ be a large (enough) positive number. Either A ($3x_1 + 2x_2 \leq 18$) or B ($2x_1 + 4x_2 \leq 16$) can be modeled as

- $3x_1 + 2x_2 \leq 18 + M\delta$
- $2x_1 + 4x_2 \leq 16 + M(1 - \delta)$
At least $K$ out of $N$ possible constraints ($K < N$) must hold

Given $N$ possible constraints and let $\delta_i \in \{0, 1\}$ for $i = 1, \ldots, N$:

\[
\begin{align*}
  f_1(x) & \leq d_1 \\
  \vdots \\
  f_N(x) & \leq d_N
\end{align*}
\]

Some "$K$ of these constraints hold" can be modeled as

\[
\begin{align*}
  f_1(x) & \leq d_1 + M\delta_1 \\
  \vdots \\
  f_N(x) & \leq d_N + M\delta_N \\
  \sum_{i=1}^{N} \delta_i & = N - K
\end{align*}
\]

(This generalizes “either-or” constraints with $K = 1$ and $N = 2$.)
Set Covering Problems

Given a set of objects $S$, and a class $T$ of subsets of $S$ with a cost associated with each subset. Find the minimum cost cover of $S$.

Examples of set covering problems:

- Find the locations of fire stations and neighborhoods they serve so the city is covered
- Find the sales representations and the stores they serve so all the stores are covered
- Find the cell tower locations and the areas they serve so the region has cell coverage
Example of a Set Covering Problem

Suppose \( S = \{1, 2, \ldots, m\} \) and 
\[
T = \{(1, 2), (1, 3, 5), (2, 4, 5), (3), (1), (4, 5)\}
\]
and all members of 
\( T \) have a cost of 1.

A cover for \( S \) would be (1,2), (1,3,5), and (4,5) with a cost of 3. Another cover for \( S \) would be (1,3,5), and (2,4,5) with a cost of 2.

Let \( \delta_i = \begin{cases} 
1 & \text{if member } i \text{ of } T \text{ is in the cover} \\
0 & \text{otherwise} 
\end{cases} \)

minimize 
\[
\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_6 
\]
subject to 
\[
\begin{align*} 
\delta_1 + \delta_2 + \delta_5 & \geq 1 \\
\delta_1 + \delta_3 & \geq 1 \\
\delta_2 + \delta_4 & \geq 1 \\
+ \delta_3 + \delta_6 & \geq 1 \\
\delta_2 + \delta_3 + \delta_6 & \geq 1 
\end{align*}
\]
Properties of a Set Covering Problem

1. The problem is a minimization one with binary variables and all constraints are $\geq$
2. All RHS coefficients are 1
3. All other matrix coefficients are 0 or 1

If Property 2 is relaxed (RHS coefficients may be 1 or greater), it is a weighted set covering problem.
If Property 2 is relaxed, and Property 3 is relaxed to allow matrix coefficients of 0, $+1$, or $-1$, it is a generalized set covering problem.

Airline crew scheduling is a good example where $S$ are “flight legs”, and $T$ are possible “rosters” or “rotations” of crews.

Special algorithms exist for set covering problems.
Set Packing Problems

Given a set of objects $S$, and a class $T$ of subsets of $S$ with a benefit value associated with each subset. Find the maximum total value associated with packing as many of the members of $T$ into $S$ as possible with no overlap.

This is similar to the set covering problem, but in set covering, we want to include at least one member of $S$, and in set packing, we want to include at most one member of $S$ in the solution.
Simple Example

Suppose you are at a convention of foreign ambassadors, each of which speaks English and also various other languages. You want to make an announcement to a group of them, but because you don’t trust them, you don’t want them to be able to speak among themselves without you being able to understand them. To ensure this, you will choose a group such that no two ambassadors speak the same language, other than English. However, you also want to give your announcement to as many ambassadors as possible.

In this case, the elements of the set $S$ are languages other than English, and the subsets in $T$ are the sets of languages spoken by a particular ambassador. If two sets are disjoint, those two ambassadors share no languages other than English. A maximum set packing will choose the largest possible number of ambassadors under the desired constraint.

(I found this simple example on Wikipedia.)
Example of a Set Packing Problem

Suppose $S = \{1, 2, \ldots, m\}$ and $T = \{(1, 2, 5), (1, 3), (2, 4), (3, 6), (2, 3, 6)\}$ and all members of $T$ have a benefit of 1.

- A pack for $S$ would be $(1,2,5)$, and $(3,6)$
- Note, we cannot include $(1,2,5)$ AND $(2,4)$ in the pack, because the member 2 would be included twice.

Let $\delta_i = \begin{cases} 
1 & \text{if member } i \text{ of } T \text{ is in the pack} \\
0 & \text{otherwise}
\end{cases}$

\[
\begin{align*}
\text{maximize} & \quad \delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 \\
\text{subject to} & \quad \delta_1 + \delta_2 \leq 1 \\
& \quad \delta_1 + \delta_3 + \delta_5 \leq 1 \\
& \quad \delta_2 + \delta_4 + \delta_5 \leq 1 \\
& \quad \delta_3 \leq 1 \\
& \quad \delta_1 \leq 1 \\
& \quad \delta_4 + \delta_5 \leq 1
\end{align*}
\]

Note: some constraints are redundant, but are included to demonstrate one constraint for each member of set $S$.
Properties of a Set Packing Problem

1. The problem is a maximization one with binary variables and all constraints are $\leq$
2. All RHS coefficients are 1
3. All other matrix coefficients are 0 or 1

If Property 2 is relaxed (RHS coefficients may be 1 or greater), it is a weighted set packing problem.

A special sort of packing problem is the matching problem, where $S$ contains nodes in a graph, and $T$ contains pairs of nodes representing an arc, and matching (or pairing)

The relaxed LP for the set packing problem with obj. coeff.s of 1 is the dual of the relaxed LP for the set covering problem with obj. coeff.s of 1

Sensitivity analysis for LP does NOT apply to Integer Programming (think of graph)
Set Partitioning Problems

Given a set of objects $S$, and a class $T$ of subsets of $S$. Now, use members of $T$ to cover all the members of $S$, and also have no overlap.

This is a combination of the set covering problem and the set packing problem. The set covering problem is computationally more difficult to solve than the set packing and set partitioning problems.

Example applications include:

- Airline crew scheduling (with no “deadheading”)
- Political districting
- Emergency management systems (fire, police, ambulance)
Example of a Set Partitioning Problem

Suppose \( S = \{1, 2, \ldots, m\} \) and 
\( T = \{(1, 2), (1, 3, 5), (2, 4, 5), (3), (1), (4, 5)\} \). Ignore objective function for the example – there are many possibilities.

A cover for \( S \) would be \((1, 2), (1, 3, 5), \) and \((4, 5)\) with a cost of 3. Another cover for \( S \) would be \((1, 3, 5), \) and \((2, 4, 5)\) with a cost of 2.

Let \( \delta_i = \begin{cases} 1 & \text{if member } i \text{ of } T \text{ is in the partition} \\ 0 & \text{otherwise} \end{cases} \)

subject to \( \delta_1 + \delta_2 + \delta_5 = 1 \)
\( \delta_1 + \delta_3 = 1 \)
\( \delta_2 + \delta_4 = 1 \)
\( + \delta_3 + \delta_6 = 1 \)
\( \delta_2 + \delta_3 + \delta_6 = 1 \)
The defining constraints of set covering, packing, and partitioning models deal with subcollections of problem objects. Covering constraints demand that at least one member of each subcollection belongs to a solution, packing constraints allow at most one member, and partitioning constraints require exactly one member.
Using binary variables $\delta_i$ for $i \in T$;

Set covering constraints requiring that at least one member of subcollection $T$ belongs to a solution are expressed by $\sum_{i \in T} \delta_i \geq 1$

Set packing constraints requiring that at most one member of subcollection $T$ belongs to a solution are expressed by $\sum_{i \in T} \delta_i \leq 1$

Set partitioning constraints requiring that exactly one member of subcollection $T$ belongs to a solution are expressed by $\sum_{i \in T} \delta_i = 1$
More to come in next set of notes:

- Knapsack problem
- Traveling salesman problem
- Quadratic assignment problem
- Facility location problem
- Scheduling and sequencing problems (e.g. job shop)
Notes 6: Job Shop Scheduling Models

IND E 599

October 20, 2010
Job shop scheduling problems seek an optimal schedule for a given collection of jobs, each of which requires a known sequence of processors that can accommodate only one job at a time.

Example (Job Shop)

Suppose there are 3 jobs waiting to be scheduled. Job 1 requires work on a sequence of 5 workstations: 1 (forging), 2 (machining), 3 (grinding), 4 (polishing) and 6 (electric discharge cutting). Job 2 requires work on a sequence of 4 workstations: 7 (heating), 1 (forging), 2 (machining) and 3 (grinding). Job 3 requires work on a sequence of 5 workstations: 2 (machining), 5 (drilling), 6 (electric discharge cutting) and 4 (polishing). Suppose the process times are given as

\[ p_{j,k} \]

the processing time (in minutes) of job \( j \) on processor \( k \).
The typical decision variables for job shop scheduling problems are:

\[ x_{j,k} \]

the start time of job \( j \) on processor \( k \)

A common objective function is to minimize makespan (minimize the maximum completion time of all jobs)

**Example (Job Shop - minimize makespan)**

\[
\min \max\{x_{1,6} + p_{1,6}, x_{2,3} + p_{2,3}, x_{3,4} + p_{3,4}\}
\]

Remember how to model \( \min \max \)?

Introduce new variable, \( z \), and minimize \( z \) with

\[ z \geq x_{1,6} + p_{1,6}, \ z \geq x_{2,3} + p_{2,3}, \ \text{and} \ z \geq x_{3,4} + p_{3,4} \]
To ensure that a job is scheduled in the pre-specified sequence, start times are subject to *precedence constraints*.

The precedence requirement that job $j$ must complete processing on processor $k$ before starting on processor $\tilde{k}$ can be expressed as

$$x_{j,k} + p_{j,k} \leq x_{j,\tilde{k}}$$

**Example (Precedence constraints for Job 2 (processors 7-1-2-3))**

$$x_{2,7} + p_{2,7} \leq x_{2,1}$$

$$x_{2,1} + p_{2,1} \leq x_{2,2}$$

$$x_{2,2} + p_{2,2} \leq x_{2,3}$$
To ensure that jobs are not scheduled simultaneously on the same processor, we may model *conflicts* by introducing binary variables:

\[ y_{j,j',k} = \begin{cases} 
1 & \text{if job } j \text{ is scheduled before job } j' \text{ on processor } k \\
0 & \text{otherwise} 
\end{cases} \]

For each \( j \) and \( j' \) that both require any processor \( k \), include the constraint pair

\[
\begin{align*}
    x_{j,k} + p_{j,k} & \leq x_{j',k} + M(1 - y_{j,j',k}) \\
    x_{j',k} + p_{j',k} & \leq x_{j,k} + My_{j,j',k}
\end{align*}
\]

for a large positive constant \( M \).
Example (Conflict constraints for Jobs 1, 2 and 3 on Processor 2)

\begin{align*}
    x_{1,2} + p_{1,2} & \leq x_{2,2} + M(1 - y_{1,2,2}) \\
    x_{2,2} + p_{2,2} & \leq x_{1,2} + My_{1,2,2} \\
    x_{1,2} + p_{1,2} & \leq x_{3,2} + M(1 - y_{1,3,2}) \\
    x_{3,2} + p_{3,2} & \leq x_{1,2} + My_{1,3,2} \\
    x_{2,2} + p_{2,2} & \leq x_{3,2} + M(1 - y_{2,3,2}) \\
    x_{3,2} + p_{3,2} & \leq x_{2,2} + My_{2,3,2}
\end{align*}

Check:

- if Job 1 uses Processor 2 first, then $y_{1,2,2} = 1 = y_{1,3,2}$, and constraints (1) and (3) are enforced.
- if Job 2 uses Processor 2 first, then $y_{1,2,2} = 0$ and $y_{2,3,2} = 1$, and constraints (2) and (5) are enforced.
Due Dates

For job $j$, let $d_j$ represent the due date. Due date constraints may be added for the last processor $T$ as

$$x_j, T + p_j, T \leq d_j$$

however, there may be no feasible schedule. Due dates are more commonly handled as goals, rather than as explicit constraints. Then,

$$x_j, T + p_j, T + s_j^- - s_j^+ = d_j$$

and

- minimize maximum tardiness, $\min \max_j \{s_j^+\}$ or
- minimize mean tardiness, $\min \frac{1}{n} \sum_{j=1}^{n} \{s_j^+\}$
Notes 7: Knapsack and Traveling Salesman Problems

IND E 599

October 26, 2010
An integer program with a single constraint. A hiker is trying to fill a knapsack to maximize value, with a volume or weight constraint. Practical applications include project selection, capital budgeting allocation, stocking a warehouse, and the cutting stock problem.

Example (Knapsack Problem)

maximize \[ p_1 \gamma_1 + p_2 \gamma_2 + \cdots + p_n \gamma_n \]
\[ a_1 \gamma_1 + a_2 \gamma_2 + \cdots + a_n \gamma_n \leq b \]
\[ \gamma_1, \gamma_2, \ldots, \gamma_n \geq 0 \text{ and integer – valued} \]

Knapsack problems can be solved using column-generation methods or dynamic programming fairly efficiently, although they are NP-complete.
Multi-dimensional Knapsack Problem

May have multiple resources being consumed, or budget limits over time periods, yielding several constraints.

Example (Multiple Constraints)

A department store is considering 4 possible expansions in a shopping mall. The following table shows how much (in millions of dollars) each expansion would cost in the next 2 years, and the required floor space (in thousands of square feet).

<table>
<thead>
<tr>
<th>Expansion</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 1</td>
<td>1.5</td>
<td>5.0</td>
<td>7.3</td>
<td>1.9</td>
</tr>
<tr>
<td>Year 2</td>
<td>3.5</td>
<td>1.8</td>
<td>6.0</td>
<td>4.2</td>
</tr>
<tr>
<td>Space</td>
<td>2.2</td>
<td>9.1</td>
<td>5.3</td>
<td>8.6</td>
</tr>
</tbody>
</table>

Suppose 10 million dollars are available in each of 2 years, and the expansion cannot exceed 17 thousand square feet.
Example

Let \( x_j = \begin{cases} 
1 & \text{if expansion } j \text{ is selected} \\
0 & \text{otherwise}
\end{cases} 
\) for \( j = 1, \ldots, 4 \)

Model three constraints:

\[
\begin{align*}
1.5x_1 + 5.0x_2 + 7.3x_3 + 1.9x_4 & \leq 10 & \text{Year 1 Budget} \\
3.5x_1 + 1.8x_2 + 6.0x_3 + 4.2x_4 & \leq 10 & \text{Year 2 Budget} \\
2.2x_1 + 9.1x_2 + 5.3x_3 + 8.6x_4 & \leq 17 & \text{Floor Space}
\end{align*}
\]
Suppose we have large rolls of paper of width $W$, and demand for $b_i$ rolls of width $w_i$, for $i = 1, \ldots, m$.

The are a LOT of possible patterns.

For pattern $j$, let $A_j$ describe the pattern, where $a_{ij}$ is the number of repetitions of width $w_i$ in the roll.

Pattern $j$ is feasible if $\sum_{i=1}^{m} a_{ij} w_i \leq W$. The difference is the scrap associated with the pattern.

Pattern 1, $A_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $|| - - w_1 - - | - - - w_2 - - - | scrap ||$

Pattern 2, $A_2 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$, $|| - - - w_2 - - - | - w_3 | - w_3 | scrap ||$
Cutting Stock Problem

Let \( x_j \) be the number of rolls of paper cut with pattern \( j \).

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j = b_i \quad \text{for } i = 1, \ldots, m \\
& \quad x_j \geq 0 \quad \text{and integer-valued}
\end{align*}
\]

The total number of possible patterns is huge, even for relatively small \( m \).

The delayed column-generation method only generates the columns \( A_j \) when needed.
A salesman wants to visit all of his customers in the minimum distance travelled (or time) without visiting any one customer twice. Practical applications include vehicle routing, job sequencing, and circuit design.

Symmetric TSP if distance from city $i$ to city $j$ equals the distance from city $j$ to city $i$, that is, $d_{ij} = d_{ji}$, otherwise it is an asymmetric TSP.

There are many ways to formulate the TSP as an integer program. Any solution is called a tour.

Eliminating subtours is a difficulty.
TSP as an Integer Program

Let \( \delta_{ij} = \begin{cases} 
1 & \text{if the tour goes from } i \text{ to } j \\
0 & \text{otherwise}
\end{cases} \)
for \( i, j = 1, \ldots, n \)

minimize \( \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \delta_{ij} \) distance traveled (or time, or cost)
subject to

- exactly one city must be visited after city \( i \)
  \( \sum_{j=0, j \neq i}^{n} \delta_{ij} = 1 \) for \( i = 1, \ldots, n \)
- exactly one city must be visited before city \( j \)
  \( \sum_{i=0, i \neq j}^{n} \delta_{ij} = 1 \) for \( j = 1, \ldots, n \)

However, these constraints are not enough to eliminate subtours. Draw Figure 9.5 on board.
Subtours

It is easy to understand subtours, but difficult to create constraints to prevent them.

One form of **subtour elimination constraints** is obtained from any $S$ - a proper subset of the cities. Every tour must cross between points in $S$ and points outside of $S$ at least twice. This leads to constraints of the form:

- number of legs between points in $S$ and points not in $S$ is $\geq 2$
  $$\sum_{i \in S} \sum_{j \notin S} \delta_{ij} + \sum_{i \notin S} \sum_{j \in S} \delta_{ij} \geq 2$$

- There is one constraint for every proper subset of $S$ of at least 3 cities.

- Potentially, there are an exponential number of such subtour elimination constraints. They are not stated explicitly, but added as needed.
Notes 8: Back to Chapter 5

IND E 599

November 1, 2010
Linear, Integer, and Non-linear Models

Many types of applications:

▶ Economic models, sometimes called input-output or Leontief models (section 5.2)
▶ Network models, including transportation and assignment problems (section 5.3)
▶ Theory of games (we will skip)
Types of applications

- **Petroleum** industry, uses distribution, resource allocation, blending and marketing models
- **Chemical** industry, blending and resource allocation models
- **Manufacturing**, such as product mix, resource allocation, multi-period and blending models
- **Transport and distribution**, including transportation, transshipment, networks, scheduling
- **Finance**, such as portfolio selection, price setting, revenue management
- **Agriculture**, such as farm management including product mix, multi-period, blending, distribution, pricing
- **Health**, including resource allocation, scheduling, policy planning, and treatment
- **Mining**, such as blending, resource allocation, multi-period
- **Manpower (personnel) planning**, including recruitment, retention, promotion, training
More applications

- **Food**, blending, distribution, resource allocation
- **Energy**, resource allocation, distribution, near real-time
- **Pulp and paper**, including blending, cutting-stock problem, resource allocation
- **Advertising**, scheduling, pricing
- **Defense**, resource allocation, distribution, design
- **Forestry management**, resource allocation, investment, multi-period, distribution
- **Data envelopment analysis**

Consider **multi-objective** and **uncertainty** too.
Input-output models, also referred to as Leontief models: identify inputs to an industry, and resulting outputs

This can be helpful in identifying decision variables, and measurable outputs for constraints and objectives

Consider the coke production problem (Homework 3, Problem 5)

Static vs dynamic input-output models: extend to multi-period models, but consider steady state assumptions versus finite horizon implications
Use of network models are widespread — transportation networks, communication networks, electrical networks, social networks, financial networks ....

Several optimization problems in production, distribution, project planning, facility location, resource management, financial planning etc. have a “network representation” which provides a powerful visual and conceptual tool to tackle these problems.

Some special types of network optimization models include: transportation problems, assignment problems, shortest-path problems, minimum spanning tree problems, maximum flow problems, minimum cost flow problems.

There are many specialized algorithms for network optimization models.
Example of a Transportation Problem

A non-profit organization manages three warehouses (W1, W2, W3) and four healthcare centers (HC1, HC2, HC3, HC4). The organization has estimated the requirements for a specific vaccine at each healthcare center in number of boxes of vials. The organization knows the number of boxes of vials available at each warehouse. They want to decide how many boxes of vials to ship from the warehouses to the healthcare centers so as to meet the demand for the vaccine at minimum total shipping cost. The shipping costs per box for W-HC pairs are as follows:

<table>
<thead>
<tr>
<th></th>
<th>HC1</th>
<th>HC2</th>
<th>HC3</th>
<th>HC4</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>464</td>
<td>513</td>
<td>654</td>
<td>867</td>
<td>75</td>
</tr>
<tr>
<td>W2</td>
<td>352</td>
<td>416</td>
<td>690</td>
<td>791</td>
<td>125</td>
</tr>
<tr>
<td>W3</td>
<td>995</td>
<td>682</td>
<td>388</td>
<td>685</td>
<td>100</td>
</tr>
<tr>
<td>REQUIREMENT</td>
<td>80</td>
<td>65</td>
<td>70</td>
<td>85</td>
<td>300</td>
</tr>
</tbody>
</table>
Let $x_{ij}$ be the number of boxes shipped from warehouse $i$ to healthcare center $j$.

$$\min \quad Z = 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} +$$
$$352x_{21} + 416x_{22} + 690x_{23} + 791x_{24} +$$
$$995x_{31} + 682x_{32} + 388x_{33} + 685x_{34}$$

$$x_{11} + x_{12} + x_{13} + x_{14} = 75$$
$$x_{21} + x_{22} + x_{23} + x_{24} = 125$$
$$x_{31} + x_{32} + x_{33} + x_{34} = 100$$
$$x_{11} + x_{21} + x_{31} = 80$$
$$x_{12} + x_{22} + x_{32} = 65$$
$$x_{13} + x_{23} + x_{33} = 70$$
$$x_{14} + x_{24} + x_{34} = 85$$

$x_{ij} \geq 0, \; (i = 1, 2, 3; \; j = 1, 2, 3, 4)$.
Transportation Problems

- **m suppliers** (or sources) and **n consumers** (or destinations).
- **ith supplier** provides **sᵢ** units and **jth consumer** demands **dⱼ** units. We assume the network is balanced, that is,
  \[
  \sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j.
  \]
  If not, add dummy nodes.
- **Transport of one unit from supplier i to consumer j costs** **cᵢⱼ**.
- **Goal**: transport the goods from the suppliers to the consumers at minimum cost.
- **xᵢⱼ** = the amount of goods transported from the **i**th supplier to the **j**th consumer.
## Parameter Table for a Transportation Problem

<table>
<thead>
<tr>
<th>Source</th>
<th>Demand</th>
<th>Cost per unit distributed</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>c_{11}</td>
<td>c_{12}</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>c_{21}</td>
<td>c_{22}</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>m</td>
<td></td>
<td>c_{m1}</td>
<td>c_{m2}</td>
</tr>
<tr>
<td>Demand</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>d_1</td>
<td>d_2</td>
<td>...</td>
</tr>
</tbody>
</table>
Network Representation of a Transportation Problem
LP Formulation of a Transportation Problem

Let $x_{ij}$ be the number of units to be distributed from source $i$ to destination $j$, for $i = 1, 2 \ldots, m$ and $j = 1, \ldots, n$.

$$\min Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = s_i, \quad \text{for } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad \text{for } j = 1, \ldots, n,$$

$x_{ij} \geq 0$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$.

Note the special structure of the problem.
If $s_i$ for $i = 1, \ldots, m$ and $d_j$ for $j = 1, \ldots, n$ are integer valued, then the optimal solution $x^*_{ij}$ is guaranteed to be integer valued!

This is due to the special structure of the problem - the $A$ matrix satisfies a property called total unimodularity.
Suppose patients are assigned to physicians. To accomplish this, each patient submits a number for each physician, often called the “mismatch index.” A high mismatch index implies the patient thinks the physician would not be a good fit, whereas a low index number implies the physician is preferred. Thus, the task is to assign patients to physicians to minimize the total mismatch. Here is a small example:

<table>
<thead>
<tr>
<th></th>
<th>Patient A</th>
<th>Patient B</th>
<th>Patient C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. 1</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Dr. 2</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Dr. 3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Assignment Problems

A special case of the transportation problem.

- Each supplier has unit supply \((s_i = 1)\) and each consumer has unit demand \((d_j = 1)\).
- The number of suppliers is equal to the number of consumers (or dummy nodes can be added).
- It turns out that one can always find an optimal solution to the assignment problem where every \(x_{ij}\) is either zero or one.
- This implies that for every \(i\) there will be a unique and distinct \(j\) for which \(x_{ij} = 1\), and we can say that supplier \(i\) is assigned to consumer \(j\).
Assignment problem: There are $n$ people available to carry out $n$ jobs. Each person must be assigned to carry out exactly one job. The cost of assigning person $i$ to job $j$ is $c_{ij}$. The problem is to find a minimum cost assignment.

We use $x_{ij} = 1$ if person $i$ is assigned to job $j$, and 0 otherwise.

\[
\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}x_{ij}
\]

\[
\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i
\]

\[
\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j
\]

$x_{ij} \in \{0, 1\}, \forall i,j.$

The binary constraint is guaranteed by the special structure of the problem!
Notes 9: Finish Chapter 5

IND E 599

November 3, 2010
Example (Ex. 2, Production Planning)

A company produces a commodity in two shifts (regular and overtime) to meet demand over the next 4 months. Given production capacities, demand forecast, costs of production (regular and overtime), and storage costs, determine how much to produce each month to satisfy demand.
### Data for Production Planning Problem

<table>
<thead>
<tr>
<th></th>
<th>Jan.</th>
<th>Feb.</th>
<th>March</th>
<th>April</th>
<th>CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan. regular</td>
<td>1.0</td>
<td>1.3</td>
<td>1.6</td>
<td>1.9</td>
<td>100</td>
</tr>
<tr>
<td>Jan. overtime</td>
<td>1.5</td>
<td>1.8</td>
<td>2.1</td>
<td>2.4</td>
<td>50</td>
</tr>
<tr>
<td>Feb. regular</td>
<td>–</td>
<td>1.0</td>
<td>1.3</td>
<td>1.6</td>
<td>150</td>
</tr>
<tr>
<td>Feb. overtime</td>
<td>–</td>
<td>1.5</td>
<td>1.8</td>
<td>2.1</td>
<td>75</td>
</tr>
<tr>
<td>Mar. regular</td>
<td>–</td>
<td>–</td>
<td>1.0</td>
<td>1.3</td>
<td>140</td>
</tr>
<tr>
<td>Mar. overtime</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>1.8</td>
<td>70</td>
</tr>
<tr>
<td>April. regular</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1.0</td>
<td>160</td>
</tr>
<tr>
<td>April. overtime</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>80</td>
</tr>
<tr>
<td>DEMAND</td>
<td>80</td>
<td>200</td>
<td>300</td>
<td>200</td>
<td>–</td>
</tr>
</tbody>
</table>

Note: This is a transportation problem with 8 sources and 4 sinks. However, it is not balanced because the total demand is 780 and the total capacity is 825. Add a sink 5 for “surplus demand” of 45.
Let $x_{ij}$ be the units to be produced during period $i$ and delivered in month $j$, for $i = \{\text{Jan-reg, Jan-ot, Feb-reg, Feb-ot, Mar-reg, Mar-ot, Apr-reg, Apr-ot}\}$ and for $j = \{\text{Jan, Feb, March, April, Surplus}\}$

Notice that the optimal solution will be integer-valued!

There are very efficient algorithms for the transportation problem.
The transshipment problem is an extension of the transportation problem, where shipments can go from any source to any destination through intermediate nodes.

See Figure 5.2 and Table 5.5 (page 80)

The transshipment problem is a special case of the minimum cost flow problem.
Example (Ex. 3, Min Cost Flow)

Consider the network in Figure 5.3 (draw on board), with availabilities of 10 and 15 at the two sources, requirements of 9, 10 and 6 at the three sinks, and a cost (per unit flow) on each arc.

Find the flow through the network that satisfies the requirements at minimum cost.
Minimum Cost Flow Formulation

Consider a graph \( G = (N, A) \) with node set \( N \) and arc set \( A \). Suppose the cost (per unit flow) along arc \((i, j)\) is denoted \( c_{ij} \).

Let \( x_{ij} \) be the flow from node \( i \) to node \( j \), along arc \((i, j)\).

\[
\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}
\]

\[
\sum_{j: (i, j) \in A} x_{ij} - \sum_{j: (j, i) \in A} x_{ji} = \begin{cases} 
  s_i & \text{for source nodes } i \\
  0 & \text{for intermediate nodes } i \\
  -r_i & \text{for sink nodes } i 
\end{cases}
\]

We may also have an upper bound \( u_{ij} \) on flow along arc \((i, j)\), then add constraints:

\[
0 \leq x_{ij} \leq u_{ij}, \; \forall (i, j) \in A
\]
Min Cost Flow Problems

The material balance constraints are very common, and ensure that flow-in equals flow-out.

The matrix of coefficients in the balance constraints is called the node-arc incidence matrix, or just incidence matrix.

The special structure of the node-arc incidence matrix has the property that ensures integer solutions, if the rhs is integer-valued.
Network Flow Problems

- A minimum cost multi-commodity network flow problem does not have the integrality property.
- A generalized network flow problem, or a network flow with gains, may alter the flow between two nodes. For example, a multiplier along an arc can alter flow to represent evaporation, wastage, or application of interest rates.
Example

The network below shows cost incurred by a truck company to drive a truck from one city to another. Find the cheapest route from New York to Los Angeles.
The shortest path problem is a special case of the min cost flow problem with 1 unit of supply at the source node, and 1 unit required at the sink node.

There are very efficient algorithms, most notably, Dijkstra’s algorithm (1959).
Maximum Flow Problem

Maximize flow through a network from a source node to a sink node, when arcs have capacity limitations.

At the source, all arcs point away from the node. At the sink, all arcs point into the node. The remaining nodes are “transshipment nodes”. The numbers written next to directed arcs in maximum flow problems are arc capacities and not costs.

Example

Find the maximum flow that can be sent through the network below from source $O$ to sink $T$. 
Maximum Flow Applications

- Maximize the flow through a company’s distribution network from its factories to its customers.
- Maximize the flow through a company’s supply network from its vendors to its factories.
- Maximize the flow of oil through a system of pipelines.
- Maximize the flow of water through a system of aqueducts.
- Maximize the flow of vehicles through a transportation network.
Maximum Flow Formulation

As in the min cost flow problem, consider a graph $G = (N, A)$ with node set $N$ and arc set $A$, and an upper bound $u_{ij}$ on flow along each arc $(i, j)$. Now, let $s$ be the source node and $t$ be the terminal (or sink) node. Find the largest possible amount of flow from $s$ to $t$ on the network.

Let $x_{ij}$ be the flow from node $i$ to node $j$, along arc $(i, j)$.

$$
\begin{align*}
\max & \quad \sum_{i:(i,t)\in A} x_{it} \\
\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} & = 0 \quad \forall i \in N/\{s, t\}, \\
0 & \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A.
\end{align*}
$$
This is a common network formulation to help in construction projects, also called critical path method (CPM). Each arc represents a task, or activity, and each node is the possible start time for the task. The network represents precedence relationships between tasks. Draw Figure 5.8 on board.
Decision variables:

- $t_0$ is start time for activities 0-1, 0-3, and 0-2
- $t_1$ is start time for activity 1-3
- $t_2$ is start time for activity 2-5
- $t_3$ is start time for activity 3-4
- $t_4$ is start time for activities 4-2 and 4-5
- $t_5$ is start time for activity 5-6
- $z$ is finish time for the project
Critical Path Formulation for Example in Figure 5.8

minimize \( z \)
subject to
\[-t_0 + t_1 \geq 4\]
\[-t_0 + t_2 \geq 12\]
\[-t_0 + t_3 \geq 7\]
\[-t_1 + t_3 \geq 2\]
\[-t_3 + t_4 \geq 10\]
\[t_2 - t_4 \geq 0\]
\[-t_2 + t_5 \geq 5\]
\[-t_4 + t_5 \geq 3\]
\[-t_5 + z \geq 4\]
Optimal Solution for Example in Figure 5.8

Project completion time, $z^* = 26$ days
$t_0 = 0$ is start time for activities 0-1, 0-3, and 0-2
$t_1 = 4$ is start time for activity 1-3
$t_2 = 17$ is start time for activity 2-5
$t_3 = 7$ is start time for activity 3-4
$t_4 = 17$ is start time for activities 4-2 and 4-5
$t_5 = 22$ is start time for activity 5-6
Critical path is 0-3-4-2-5-6
Review for Take-home Midterm

Text: Model Building in Mathematical Programming

► Chapter 1: Introduction, Concept of an Optimization Model, Linear Program Examples (Product Mix, Blending)
► Chapter 2: Solving Mathematical Programming Models (Use AIMMS in this Class)
► Chapter 3: Building Linear Programming Models
► Chapter 4: Structured Linear Programming Models (SKIP)
► Chapter 5: Applications and Special Types of Mathematical Programming Model - many applications and network models
► Chapter 6: Interpreting and Using the Solution of a Linear Programming Model - shadow prices, ranging and sensitivity analysis
► Chapter 7: Non-linear Models (SKIP)
► Chapter 8: Integer Programming
► Chapter 9: Building Integer Programming Models I
There is a lot of research on this topic!

There are a lot of applications, primarily in the financial world, but also in traditional industrial engineering like supply chain and inventory.

We will discuss:

- chance-constrained programming (CCP)
- scenario-based stochastic programming with recourse
- robust optimization
Consider Demand Constraints with Uncertainty

minimize production costs subject to meeting demand
where $\sum_j a_j x_j$ represents the amount produced

Example (Deterministic Demand Constraint)

$$\sum_j a_j x_j \geq d$$

Suppose demand $d$ is uncertain. We could model demand with a random variable $D$ and a cumulative probability distribution $F_D(d) = P(D \leq d) = \epsilon$

(Draw a density and cumulative distribution on board with $\epsilon = 0.95$)
Chance Constraint

Example (Probabilistic Demand Constraint)

\[ P \left( \sum_j a_j x_j \geq D \right) \geq \epsilon \]

\[ P \left( D \leq \sum_j a_j x_j \right) = F_D \left( \sum_j a_j x_j \right) \geq \epsilon \]

\[ \sum_j a_j x_j \geq F_D^{-1}(\epsilon) \]

Note that the constraint is still linear. The interpretation of the sensitivity analysis (right-hand side ranging, shadow price) is in terms of probability, not actual demand. Use the probability distribution (may be estimated with historical data) to relate probabilities to demand values.
Suppose demand $D$ has a discrete probability distribution, estimated with 3 scenarios:

$$P(D = d) = \begin{cases} 
0.1 & \text{if } d = d_1 \quad \text{(scenario 1)} \\
0.7 & \text{if } d = d_2 \quad \text{(scenario 2)} \\
0.2 & \text{if } d = d_3 \quad \text{(scenario 3)} 
\end{cases}$$

We could easily write $\sum_j a_j x_j \geq d_2$ to ensure a probability of 0.80 of meeting demand, but suppose our costs include penalties for being above or below demand.
Example (Scenario-based Demand)

Minimize \( \sum_j c_j x_j + \sum_s p_s \left( e y^{(s)} + f z^{(s)} \right) \)

\( \sum_j a_j x_j - y^{(s)} + z^{(s)} = d_s \quad \text{for } s = 1, 2, 3 \)

where \( \sum_j c_j x_j \) represents the production costs, \( p_1 = 0.1 \), \( p_2 = 0.7 \), \( p_3 = 0.2 \) are the scenario probabilities, \( y^{(s)} \) is excess production under scenario \( s \), \( z^{(s)} \) is shortfall under scenario \( s \), \( e \) penalizes being over demand, and \( f \) penalizes being under demand.

This is a scenario-based stochastic program (SP) with simple recourse.
Suppose after the initial production decision $x$, the demand scenario is observed $d_s$ and another decision can be made $\tilde{x}_s$.

**Example (Two-stage Stochastic Program with Recourse)**

Minimize $c^T x + \sum_s p_s (Q_s(x, d_s))$

subject to $Ax \leq b$, $x \geq 0$

where

$Q_s(x, d_s) = \min \left\{ q_s^T \tilde{x}_s \mid W_s \tilde{x}_s \leq d_s + T_s x, \ \tilde{x}_s \geq 0 \right\}$
Examples of 2-stage SP with Recourse

- monthly decisions as 1st stage, daily modifications as 2nd stage
- facility location as 1st stage, distribution as 2nd stage
- route planning as 1st stage, modifications as 2nd stage
- emergency planning (preparedness) as 1st stage, response as 2nd stage
- financial investments, long-term as 1st stage, and with adaptations
A straightforward approach is to convert the SP problem into its deterministic equivalent. This usually has a block structure that may be able to be solved efficiently.

Many specialized algorithms have been (and are being) developed making use of the block structure and Bender’s decomposition or Dantzig-Wolfe decomposition.
Literature on Stochastic Programming

Committee on Stochastic Programming (COSP) home page: http://stoprog.org/

Case Studies:
http://www.rug.nl/feb/mhvandervlerk/spcourse/cases09.pdf
- Multi-Period Production Planning
- Water Management in the Bodrog River Area
- Budgeting Cost of Nursing in a Hospital
- Growing Maize and Sorghum in Kilosa District
- Product Mix Problem
- Investment Planning for Electricity Generation
- Multi-stage Asset Liability Management (ALM) for Pension Funds
- Optimizing the Timetable of the Haarlem-Maastricht Railway Connection
Books on Stochastic Programming

Books on SP, continued


An airline is selling tickets for flights to a particular destination. The flight will depart in 3 weeks’ time. It can use up to 6 planes, each costing 50,000 to hire. Each plane has:

- 37 First class seats
- 38 Business class seats
- 47 Economy class seats

Up to 10% of seats in any one category can be transferred to an adjacent category.

The airline wishes to decide a price for each of these seats. They may update these prices after one week and two weeks. Once a customer has purchased a ticket there is no cancellation option.
For administrative simplicity, three pricing options have been identified for each period (low, medium, high), and are given in Table 12.16 (page 257).

Demand is uncertain but will be affected by price. Forecasts have been made based on 3 scenarios in each period, with probabilities $p_1 = 0.1$, $p_2 = 0.7$, and $p_3 = 0.2$. The forecast demands are given in Table 12.17 (page 258).

For this problem, hindsight is given in Table 12.18; at the beginning of the next period, the demand for the price option chosen in the previous period is observable.
Yield Management

Decide:

▶ price levels (options) for the current period,
▶ how many seats to sell in each class (depending on demand),
▶ the provisional number of planes to book and provisional price levels and seats to sell in future periods

in order to maximize expected yield.

You should schedule to be able to meet commitments under all possible combinations of scenarios.
Develop a three-period stochastic programming model.

1. Solving the model for the first time will give recommended price levels and sales three weeks from departure, and recommended price levels and sales for subsequent weeks under all possible scenarios.

2. A week later the model will be rerun, taking into account the committed sales and revenue in the first week, to redetermine recommended prices and sales for the second week (i.e. with recourse) and the third week under all possible scenarios.

3. Repeat the procedure a week later.
Variables:

- $p_{1ch} = 1$ if price option $h$ is chosen for class $c$ in week 1, 0 otherwise, for $c = 1, 2, 3$, and $h = 1, 2, 3$.

- $p_{2ich} = 1$ if price option $h$ is chosen for class $c$ in week 2 as a result of scenario $i$ in week 1, 0 otherwise, for $c = 1, 2, 3$, $h = 1, 2, 3$, and $i = 1, 2, 3$.

- $p_{3ijch} = 1$ if price option $h$ is chosen for class $c$ in week 3 as a result of scenario $i$ in week 1 and scenario $j$ in week 2, 0 otherwise, for $c = 1, 2, 3$, $h = 1, 2, 3$, and $i, j = 1, 2, 3$.

Similarly,

- $s_{1ich}$ is number of tickets sold in week 1 for class $c$ under price option $h$ and scenario $i$.

- $S_{2ijch}$

- $S_{3ijkch}$
Similarly,

- $r_{1ich}$ is revenue in week 1 from class $c$ under price option $h$ and scenario $i$,
- $r_{2ijch}$
- $r_{3ijkch}$
- $u_{ijkc}$ is slack capacity in class $c$ under scenarios $i, j, k$ in successive weeks
- $v_{ijkc}$ is excess capacity in class $c$ under scenarios $i, j, k$ in successive weeks
- $n$ is number of planes to fly.
Data and Objective Function

- $Q_i$ is the probability of scenario $i$, $i = 1, 2, 3$
- $P$ and $D$ with appropriate subscripts are given prices and demands for time periods, scenarios, classes and options, i.e. $P_{1ch}, P_{2ch}, P_{3ch}$ and $D_{1ich}, D_{2jch}, D_{3kch}$
- $C_c$ is the seat capacity per plane for class $c$

Objective: maximize expected yield

$$\max \sum_{i,c,h} Q_i r_{1ich} + \sum_{i,j,c,h} Q_i Q_j r_{2ijch} + \sum_{i,j,k,c,h} Q_i Q_j Q_k r_{3ijkch} - 50n$$
For a particular price option and under each scenario, the sales cannot exceed the estimated demand and the revenue must be the product of the price and the sales:

\[ r_{1ich} - P_{1ch}s_{1ich} \leq 0 \]
\[ P_{1ch}s_{1ich} - r_{1ich} + P_{1ch}D_{1ich}p_{1ch} \leq P_{1ch}D_{1ich} \text{ for all } i, c, h \]
\[ r_{2ijch} - P_{2ch}s_{2ijch} \leq 0 \]
\[ P_{2ch}s_{2ijch} - r_{2ijch} + P_{2ch}D_{2jch}p_{2ich} \leq P_{2ch}D_{2jch} \text{ for all } i, j, c, h \]
\[ r_{3ijkch} - P_{3ch}s_{3ijkch} \leq 0 \]
\[ P_{3ch}s_{3ijkch} - r_{3ijkch} + P_{3ch}D_{3kch}p_{3ijkch} \leq P_{3ch}D_{3kch} \text{ for all } i, j, k, c, h \]
Seat capacities must be abided by for all scenarios:

\[ s_{1ich} + s_{2ijch} + s_{3ijkch} + u_{ijkc} - v_{ijkc} - C_c n \leq 0 \] for all \( i, j, k, c \)

Seat adjustment is possible between classes:

\[ u_{ijkc} - 0.1 C_c \leq 0 \]

\[ v_{ijkc} - 0.1 C_c \leq 0 \] for all \( i, j, k, c \)

\[ \sum_c u_{ijkc} - \sum_c v_{ijkc} = 0 \] for all \( i, j, k \)
Even More Constraints

Exactly one price option must be chosen in each class under each set of scenarios:

\[
\begin{align*}
\sum_{h} p_{1ch} &= 1 \text{ for all } c \\
\sum_{h} p_{2ich} &= 1 \text{ for all } i, c \\
\sum_{h} p_{3ijch} &= 1 \text{ for all } i, j, c
\end{align*}
\]
Numbers sold cannot exceed demand:

\[ s_{1ich} \leq D_{1ich} p_{1ch} \quad \text{for all } i, c, h \]
\[ s_{2ijch} \leq D_{2jch} p_{2ich} \quad \text{for all } i, j, c, h \]
\[ s_{3ijkch} \leq D_{3kch} p_{3ijkch} \quad \text{for all } i, j, k, c, h \]

Up to six planes can be flown:

\[ n \leq 6 \]

All \( p \) variables must be 0-1 integer, and \( n \geq 0 \) must be integer. The \( s, r, u, v \) variables must be non-negative, and may be treated as continuous (\( s \) may be rounded to integer values)
Yield Management - Solution

First Run: chooses the highest price option (1) for all classes for period 1, with maximum number of seats as 45, 45, 55 for First, Business and Economy classes, respectively. It also provides provisionally price options for periods 2 and 3 under all scenarios. Provisionally, book three planes. Expected revenue is 169,544

Second Run - use the demand given as hindsight for period 1: chooses price options low, high, high (3, 1, 1) for period 2, with maximum number of seats as 50, 60, 50 for First, Business and Economy classes, respectively. It also provides provisionally price options for period 3 under all scenarios. Provisionally, still book three planes. Expected revenue is 172,969

Third Run - use the demand given as hindsight for periods 1 and 2: chooses price options high, medium, high (1, 2, 1) for period 3, with maximum number of seats as 40, 25, 36 for First, Business and Economy classes, respectively.
If the model is altered to maximize yield subject to expected demand (normal deterministic model without recourse), three planes are still needed, and for the first period the price options are the same (high, high, high), however the number of seats in the first period are fewer (24, 39, 50). The revenue is less than that found with the SP with Recourse.
Next lecture, I’ll upload and discuss a nice introductory article: Sensitivity Analysis and Uncertainty in Linear Programming, Julia L. Higle and Stein W. Wallace, Interfaces, Vol. 33, No. 4, July – August 2003, pp. 53-60
Discuss the following introductory article:

Most LP models rely on large amounts of data, much of which represents “best-guess” estimates.

Sensitivity analysis, shadow prices, and parametric programming are useful to explore how solutions change as data elements vary.

However, sensitivity analysis is typically aimed at a deterministic problem.

To plan under uncertainty, it is “critical to properly reflect the manner in which decision and information are interspersed”

- Identify the times at which decisions are made
- Identify what will be known and what will remain uncertain when the decisions are made

“Typically, LP models do not offer such a reflection. As a consequence, the results of sensitivity analyses can be misleading.”
Product mix problem: manufacture desks, tables and chairs
A desk sells for $60, a table for $40, and a chair for $10. The manufacture of each type of furniture requires lumber and two types of skilled labor: carpentry and finishing.

**Example**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Cost ($)</th>
<th>Desk</th>
<th>Table</th>
<th>Chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lumber (board feet)</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Carpentry (hours)</td>
<td>5.2</td>
<td>2.0</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Finishing (hours)</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>Profit ($)</td>
<td>60</td>
<td>40</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Demand</td>
<td>150</td>
<td>125</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>
Decision variables:

- $y_d$ number of desks to produce
- $y_t$ number of tables to produce
- $y_c$ number of chairs to produce
- $x_l$ number of board feet of lumber to acquire
- $x_c$ number of labor hours to acquire for carpentry
- $x_f$ number of labor hours to acquire for finishing
- $s_d$ number of desks to sell
- $s_t$ number of table to sell
- $s_c$ number of chairs to sell
Formulation (P0)

\[
\begin{align*}
\text{max} & \quad -2x_l - 5.2x_c - 4x_f + 60s_d + 40s_t + 10s_c \\
\text{s.t.} & \quad -x_l + 8y_d + 6y_t + y_c \leq 0 \\
& \quad -x_c + 2y_d + 1.5y_t + 0.5y_c \leq 0 \\
& \quad -x_f + 4y_d + 2y_t + 1.5y_c \leq 0 \\
& \quad s_d \leq 150 \\
& \quad s_d - y_d \leq 0 \\
& \quad s_t \leq 125 \\
& \quad s_t - y_t \leq 0 \\
& \quad s_c \leq 300 \\
& \quad s_c - y_c \leq 0 \\
& \quad x_l, x_f, x_c, y_d, y_t, y_c, s_d, s_t, s_c \geq 0
\end{align*}
\]
This formulation is more than is necessary for this basic LP, however it explicitly identifies the sequence of three decisions:

1. Acquire resources \((x_l, x_c, x_f)\)
2. Produce items \((y_d, y_t, y_c)\)
3. Sell items \((s_d, s_t, s_c)\)

It is assumed that all data is available before any of these decisions are made.

With this assumption, a sensitivity analysis can be conducted to explore the “robustness” of a solution to inaccuracies in the data.

The sensitivity analysis to \((P0)\) indicates that the strategy to produce as many desks and tables that can be sold, but do not produce any chairs is valid for any set of non-negative demands.
Suppose demand is uncertain, but low, most likely, and high values are available (3 scenarios):

<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>Most Likely</th>
<th>High</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desks</td>
<td>50</td>
<td>150</td>
<td>250</td>
<td>150</td>
</tr>
<tr>
<td>Tables</td>
<td>20</td>
<td>110</td>
<td>250</td>
<td>125</td>
</tr>
<tr>
<td>Chairs</td>
<td>200</td>
<td>225</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>Probability</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

Let

- $s \in \{l, m, h\}$ represents scenario, for low, most likely and high
- $D_{ds}$ be demand for desks under scenario $s$
- $D_{ts}$ be demand for tables under scenario $s$
- $D_{cs}$ be demand for chairs under scenario $s$
- $p_l$, $p_m$, $p_h$ are probabilities of scenarios low, most likely and high
When is the uncertain demand revealed?

- (P1) Demand is determined before acquiring resources, and before production (e.g. made to order)
- (P2) Demand is determined after acquiring resources and after production (e.g. only know demand at time of sales)
- (P3) Demand is determined after acquiring resources, but before production (e.g. intermediate process, where production schedule can be adapted once orders are known)
Formulation (P1)

If demand is known at the start, "our decisions are not exposed to uncertainty" – all decisions are made with respect to a scenario (add scenario subscript to all variables)

\[
\begin{align*}
\text{max} & \quad \sum_{s \in \{l, m, h\}} \left( -2x_{ls} - 5.2x_{cs} - 4x_{fs} + 60s_{ds} + 40s_{ts} + 10s_{cs}p_s \right) \\
- x_{ls} & \quad + 8y_{ds} + 6y_{ts} + y_{cs} \quad \leq \quad 0 \\
- x_{cs} & \quad + 2y_{ds} + 1.5y_{ts} + 0.5y_{cs} \quad \leq \quad 0 \\
- x_{fs} & \quad + 4y_{ds} + 2y_{ts} + 1.5y_{cs} \quad \leq \quad 0 \\
s_{ds} & \quad \leq \quad D_{ds} \\
s_{ds} & \quad \leq \quad 0 \\
s_{ts} & \quad \leq \quad D_{ts} \\
s_{ts} & \quad \leq \quad 0 \\
s_{cs} & \quad \leq \quad D_{cs} \\
s_{cs} & \quad \leq \quad 0 \\
x_{ls}, x_{fs}, x_{cs}, y_{ds}, y_{ts}, y_{cs}, s_{ds}, s_{ts}, s_{cs} & \geq 0 \quad \text{for all } s \in \{l, m, h\}
\end{align*}
\]
Formulation (P1) is separable by scenario, and we can obtain scenario-specific solutions independently. Only the objective function combines the scenarios in expected profit.

Notice the difference between (P1), where scenario-specific solutions are available, and (P0), where a single solution is determined based on the expected demand.
If demand is known after acquisition and production decisions, then only sales variables have a scenario subscript.

\[
\max -2x_l - 5.2x_c - 4x_f + \sum_{s \in \{l, m, h\}} (60s_{ds} + 40s_{ts} + 10s_{cs}p_s)
\]

\[
\begin{align*}
-x_l & \quad +8y_d + 6y_t + y_c \quad \leq 0 \\
-x_c & \quad +2y_d + 1.5y_t + 0.5y_c \quad \leq 0 \\
-x_f & \quad +4y_d + 2y_t + 1.5y_c \quad \leq 0 \\
\end{align*}
\]

\[
\begin{align*}
s_{ds} & \quad -y_d \quad \leq D_{ds} \\
s_{ds} & \quad \leq 0 \\
&s_{ts} - y_t \quad \leq D_{ts} \\
&s_{ts} \quad \leq 0 \\
&s_{cs} - y_c \quad \leq D_{cs} \\
&s_{cs} \quad \leq 0 \\
\end{align*}
\]

\[
x_l, x_f, x_c, y_d, y_t, y_c, s_{ds}, s_{ts}, s_{cs} \geq 0 \quad \text{for all } s \in \{l, m, h\}
\]
Formulation (P2) is NOT separable by scenario, and acquisition $x$ and production $y$ decisions must be made before demand is known.

Notice the difference between (P2) and (P1): in (P2) the acquisition and production decisions must be made to account for the three separate scenarios on how much can be sold.

(P2) is also different from (P0) because the weighted profit/cost associated with each demand scenario is included in the objective function.
Formulation (P3)

The intermediate situation, where demand is known after acquisition and before production decisions, then production and sales variables have a scenario subscript

\[
\max - 2x_l - 5.2x_c - 4x_f + \sum_{s \in \{l, m, h\}} (60s_{ds} + 40s_{ts} + 10s_{cs} \rho_s)
\]

\[
-x_l \quad +8y_{ds} + 6y_{ts} + y_{cs} \quad \leq \quad 0
\]

\[
-x_c \quad +2y_{ds} + 1.5y_{ts} + 0.5y_{cs} \quad \leq \quad 0
\]

\[
-x_f \quad +4y_{ds} + 2y_{ts} + 1.5y_{cs} \quad \leq \quad 0
\]

\[
s_{ds} \quad -y_{ds} \quad \leq \quad D_{ds}
\]

\[
s_{ds} \quad -y_{ds} \quad \leq \quad 0
\]

\[
s_{ts} \quad -y_{ts} \quad \leq \quad D_{ts}
\]

\[
s_{ts} \quad -y_{ts} \quad \leq \quad 0
\]

\[
s_{cs} \quad -y_{cs} \quad \leq \quad D_{cs}
\]

\[
x_l, x_f, x_c, y_{ds}, y_{ts}, y_{cs}, s_{ds}, s_{ts}, s_{cs} \geq 0 \quad \text{for all } s \in \{l, m, h\}
\]
Formulation (P3) is NOT separable by scenario.

Acquisition decisions $x$ must be made before demand is known, however production decisions $y_{ds}, y_{ts}, y_{cs}$ can be adapted to the demand. An example would be where acquisition of raw materials has a long lead time, so that decision must be made very early, but the actual production would begin once orders are made (demand is known).

(P3) is also appropriate when the acquired resources can be used to create products for which there is demand.
Compare Solutions (P0), (P1), (P2) and (P3)

<table>
<thead>
<tr>
<th>Variables</th>
<th>Mean Demand</th>
<th>(P.1) Scenarios</th>
<th>(P.2)</th>
<th>(P.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lumber</td>
<td>1,950</td>
<td>Low: 520</td>
<td>Most Likely: 1,860</td>
<td>High: 3,500</td>
</tr>
<tr>
<td>Finishing labor</td>
<td>850</td>
<td>240</td>
<td>820</td>
<td>500</td>
</tr>
<tr>
<td>Carpentry labor</td>
<td>487.5</td>
<td>130</td>
<td>465</td>
<td>875</td>
</tr>
</tbody>
</table>

Resource Quantities

<table>
<thead>
<tr>
<th>Variables</th>
<th>Production Quantities</th>
<th>(P.2)</th>
<th>(P.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lumber</td>
<td>Low: 50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Most Likely: 150</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>High: 250</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>Finishing labor</td>
<td>Low: 20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Most Likely: 110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>High: 250</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>Carpentry labor</td>
<td>Low: 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Most Likely: 200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>High: 0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Objective value

4,165

1,142

1,730
Compare Solutions (P0), (P1), (P2) and (P3)

- (P0) is overly optimistic and a deterministic sensitivity analysis does not provide the insight of the other models.
- (P1) provides the optimal solution to each scenario - which may be useful for planning purposes.
- (P1) is also optimistic – it assumes demand is known ahead of time – as if a crystal ball or perfect information.
- (P2) is preferable to (P0) because it balances the “potential sunk cost” of producing items that cannot be sold with the “potential revenue” from selling a larger number of items.
- The expected profit from (P2) is lower than that from (P0), because the uncertainty is being accounted for.
(P3) is the only model where chairs are produced at all, and only in the low demand scenario. Although chairs on their own are not profitable, when the acquired resources are a sunk cost, the chairs provide an opportunity to recoup these costs, and can be viewed as a fallback position.

The expected profit from (P3) exceeds that from (P2), which is not surprising, because it is almost always economically advantageous to delay decisions until all the information is known.
Is expected profit the best objective function to use?

Table 5 in article demonstrates that expected profit does not distinguish between alternatives with different profit distributions. The 3 alternatives have the same expected value, but most people would attribute different levels of risk.

A utility function is one way to encapsulate the trade-off between expected profit and risk.

Coming up: article on risk measures, and hand-out on multiple objectives.
Discuss the two chapters on multi-objective optimization from:


minimize $f_1(x), f_2(x), \ldots, f_m(x)$
subject to $x \in S$

where $x \in S \subset \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ and each $f_i$ maps $S \rightarrow \mathbb{R}$ for $i = 1, \ldots, m$ and $m \geq 2$

Definition
A point $x^* \in S$ is Pareto optimal if there does not exist another point $x \in S$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \ldots, m$ and $f_j(x) < f_j(x^*)$ for at least one $j \in \{1, \ldots, m\}$. 
Definition
A feasible solution \( x \in S \) dominates another feasible solution \( y \in S \) if \( x \) is at least as good as \( y \) with respect to every objective function, that is \( f_i(x) \leq f_i(y) \) for all \( i = 1, \ldots, m \), and is strictly better with respect to at least one objective function, that is \( f_j(x) < f_j(y) \) for at least one \( j \in \{1, \ldots, m\} \).

The set of Pareto optimal solutions is the set of non-dominated solutions. A Pareto optimal solution is also called an efficient point.

The trade-off curve or efficient frontier is the graph of the (usually two) objective functions for all Pareto optimal points.
Example of Pareto optimal set and Efficient Frontier

Example
Two convex objective functions and a single integer variable

\[(P2) \quad \min f_1(x) = x^2 - 80x + 1700 \]
\[
\min f_2(x) = x^2 - 120x + 3800 \\
s.t. \quad x \in \mathbb{Z} \cap [0, 100] \]
Chemco is considering producing 3 products. Information is given for the per-unit contribution of labor requirements, raw material, profit and pollution. Let \( x_i \) be the number of units of product \( i \) to produce, \( i = 1, 2, 3 \)

- **Objective 1:** profit = \( 10x_1 + 9x_2 + 8x_3 \)
- **Objective 2:** pollution = \( 10x_1 + 6x_2 + 3x_3 \)

\[
\begin{align*}
\text{max} & \quad 10x_1 + 9x_2 + 8x_3 & \quad \text{profit} \\
\text{min} & \quad 10x_1 + 6x_2 + 3x_3 & \quad \text{pollution} \\
\text{s.t.} & \quad 4x_1 + 3x_2 + 2x_3 \leq 1,300 \quad \text{Labor} \\
& \quad 3x_1 + 2x_2 + 2x_3 \leq 1,000 \quad \text{Raw material} \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]
1. First, ignore pollution, and maximize profit subject to labor and raw material constraints: solution is 
   \[ x_1 = 0, \ x_2 = 300, \ x_3 = 200 \] and Profit=4,300, Pollution=2,400

2. Next, add a pollution constraint:
   \[ 10x_1 + 6x_2 + 3x_3 \leq POLL \]
   and set POLL=2,300 (Profit=4,266.67)

3. Set POLL=2200, 2100, 2000, ... and graph (POLL, Profit)
   (Draw trade-off curve on board.)
Proctor and Ramble places ads on football games and soap operas. If $F$ one-minute ads are placed on football games, and $S$ one-minute ads are placed on soap operas, then the number of men and women reached (in millions) and cost is given. P&R has a $1 million advertising budget and wants to maximize the number of men and the number of women who see its ads (2 objectives).

\[
\begin{align*}
\text{max} & \quad 20\sqrt{F} + 4\sqrt{S} \quad & \text{men reached} \\
\text{max} & \quad 4\sqrt{F} + 15\sqrt{S} \quad & \text{women reached} \\
\text{s.t.} & \quad 100F + 60S \leq 1,000 \quad & \text{Budget} \\
& \quad F, S \geq 0
\end{align*}
\]
Nonlinear Trade-off Curve

1. First, ignore women reached, and maximize men reached subject to budget constraint: solution is $F = 9.38, S = 1.04$ and women, men reached = $(27.55, 65.32)$.

2. Second, ignore men reached, and maximize women reached subject to budget constraint: women, men reached = $(62.5, )$

3. Next, add a nonlinear constraint:

$$4\sqrt{F} + 15\sqrt{S} \geq W$$

and solve with $W = 30, 35, 40, \ldots, 55, 60$ and $62.5$

(Draw trade-off curve on board.)
Trade-off Curve Procedure, or $\epsilon$-Method, for 2 Objectives

- Choose one objective, and optimize it alone (value=$v_1$). Determine the other objective function value at the solution (value = $v_2$). The point ($v_1$, $v_2$) is on the trade-off curve, or efficient frontier.

- Choose the second objective, and optimize it alone (value=$\bar{v}_2$). Determine the first objective function value at the solution (value = $\bar{v}_1$). The point ($\bar{v}_1$, $\bar{v}_2$) determines the other endpoint of the trade-off curve.

- Keep the first objective, and add the second objective function to the set of constraints, and vary its righthand side from $v_2$ to $\bar{v}_2$ to produce more points on the efficient frontier.

When there are more than two objectives, you can keep one as the objective and add the others in as constraints, and vary the righthand sides to explore the efficient frontier and Pareto optimal set.
More Examples

- Bank investment example with 8 variables and 3 objectives:
  - maximize profit,
  - minimize capital-adequacy ratio (ratio of required capital to actual capital)
  - minimize illiquid risk assets
- Dynamometer ring design with 3 variables and 2 objectives:
  - maximize sensitivity
  - maximize rigidity
- Hazardous waste disposal with about 45 variables and 2 objectives:
  - minimize distance
  - minimize population
Efficient points in a 2-dimensional domain and efficient frontier with 2 objectives

Preemptive Optimization: Illustrate on Bank Investment Example

Weighted Sum of Objectives: Illustrate on Hazardous Waste Example

Goal Programming
Notes 14: Risk in Optimization

IND E 599

December 1, 2010
Discuss Prof. Rockafellar’s tutorial on risk:

Optimization under Uncertainty

- Sensitivity analysis and parametric programming
- Chance constraints
- Stochastic programming
- Robust optimization

How to capture RISK?
What is Risk?

Definition (Risk - from on-line dictionary)

1. Hazard; danger; peril; exposure to loss, injury, or destruction. The imminent and constant risk of assassination, a risk which has shaken very strong nerves.

2. Hazard of loss; liability to loss in property. To run a risk; to incur hazard; to encounter danger.
   - To expose to risk, hazard, or peril; to venture; as, to risk goods on board of a ship; to risk one’s person in battle; to risk one’s fame by a publication.
   - To incur the risk or danger of; as, to risk a battle.

Risk: Combination of the likelihood of an occurrence of a hazardous event or exposure(s) and the severity of injury or ill health that can be caused by the event or exposure(s)
Uncertainty must be taken in a sense radically distinct from the familiar notion of Risk, from which it has never been properly separated. The term “risk,” as loosely used in everyday speech and in economic discussion, really covers two things which, functionally at least, in their causal relations to the phenomena of economic organization, are categorically different. ... The essential fact is that “risk” means in some cases a quantity susceptible of measurement, while at other times it is something distinctly not of this character; and there are far-reaching and crucial differences in the bearings of the phenomenon depending on which of the two is really present and operating. ... It will appear that a measurable uncertainty, or “risk” proper, as we shall use the term, is so far different from an unmeasurable one that it is not in effect an uncertainty at all.
Another distinction between risk and uncertainty is proposed in *How to Measure Anything: Finding the Value of Intangibles in Business* and *The Failure of Risk Management: Why It’s Broken and How to Fix It* by Doug Hubbard:

- **Uncertainty:** The lack of complete certainty, that is, the existence of more than one possibility. The “true” outcome/state/result/value is not known.

- **Measurement of uncertainty:** A set of probabilities assigned to a set of possibilities. Example: There is a 60% chance this market will double in five years.

- **Risk:** A state of uncertainty where some of the possibilities involve a loss, catastrophe, or other undesirable outcome.

- **Measurement of risk:** A set of possibilities each with quantified probabilities and quantified losses. Example: There is a 40% chance the proposed oil well will be dry with a loss of $12 million in exploratory drilling costs.
Many situations where decisions must be made “before the facts are in”

- Financial Optimization
  - Markowitz model of portfolio selection
  - Sharpe ratio combining expected return and variance
- Agriculture
- Building a facility
- Preparing for an emergency (earthquake, flood)
- Designing a bridge, airplane, roads
- Health care
Example (Deterministic Model (1.1))

minimize \( c_0(x) \)
over all \( x \in S \)
satisfying \( c_i(x) \leq 0 \) for \( i = 1, \ldots, m \)

Instead of \( c_i(x) \), suppose we have \( c_i(x, \omega) \) where \( \omega \in \Omega \) represents future states of knowledge (e.g. scenarios, or probability space).

Now consider a collection of functions

\[ c_i(x) : \omega \rightarrow c_i(x, \omega) \quad \text{for } i = 0, \ldots, m \]

Treat \( c_i(x) \) as random variables!

**How should risk be taken into account?**

How do we interpret the constraints? the objective function? Introduce safety margins? Protect against undesired outcomes?

*No conceptual distinction should be made between the treatment of the objective function \( c_0 \) and the constraint functions \( c_1, \ldots, c_m \)*
Example (Stochastic Model)

\[
\begin{align*}
\text{minimize} & \quad c_0(x) \\
\text{over all} & \quad x \in S \\
\text{satisfying} & \quad c_i(x) \leq 0 \text{ for } i = 1, \ldots, m
\end{align*}
\]

Must reinterpret the constraints and objective function.

Some traditional approaches that condense the random variables that depend on \( x \) back to numbers that depend on \( x \).
Approach 1: Guessing the Future

Identify a single element $\bar{\omega} \in \Omega$ as the best guess of the unknown information, and

\[
\begin{align*}
\text{minimize} & \quad c_0(x, \bar{\omega}) \\
\text{over all} & \quad x \in S \\
\text{satisfying} & \quad c_i(x, \bar{\omega}) \leq 0 \text{ for } i = 1, \ldots, m
\end{align*}
\]

A solution $x^*$ fails to hedge against the uncertainty, and “puts all the eggs in one basket.”

Another weakness is that small changes in $\bar{\omega}$ (e.g. change in scenario) may result in large changes in the solution. Thus, the dangers of not hedging could be compounded by instability.
Approach 2: Worst-Case Analysis

A conservative approach is to prepare for the worst:

\[
\text{minimize} \quad \sup_{\omega \in \Omega} c_0(x, \omega) \\
\text{over all} \quad x \in S \\
\text{satisfying} \quad \sup_{\omega \in \Omega} c_i(x, \omega) \leq 0 \quad \text{for} \quad i = 1, \ldots, m
\]

A solution \( x^* \) ensures that all constraints are satisfied, no matter what the future may bring (robust optimization). The goal is to “eliminate all risk” but there is a price for that.

A weakness is that the feasible set may be empty. There is no solution that is feasible under ALL possibilities.

Must find a balance between the practicality of \( x \) and the chance that the resulting design could be overpowered by some extreme event.
Approach 3: Relying on Expectations

Use the expectation of the random variable $c_i(x)$ to create a number that depends on $x$:

$$\minimize \ E[c_0(x)]$$
$$\text{over all } \ x \in S$$
$$\text{satisfying } \ E[c_i(x)] \leq 0 \text{ for } i = 1, \ldots, m$$

This seems normal for the objective function, but “ridiculous” for the constraints. Who would be satisfied if a constraint (e.g. safety of a structure) was satisfied on the average?

Expectations are suitable for situations modeling long-run behavior, where stochastic ups and downs can average out.

Not appropriate when there is a short-run focus with serious risks to be accounted for.
Approach 4: Standard Deviation Units as Safety Margins

Use the standard deviation to ensure that the expected value is reassuringly below 0:

\[
\begin{align*}
\text{minimize} & \quad \mu[c_0(x)] + \lambda_0 \sigma[c_0(x)] \\
\text{over all} & \quad x \in S \\
\text{satisfying} & \quad \mu[c_i(x)] + \lambda_i \sigma[c_i(x)] \leq 0 \text{ for } i = 1, \ldots, m
\end{align*}
\]

The major flaw in this approach is that it lacks a property called coherency.
Approach 5: Probabilities of Compliance

Use probabilistic or chance constraints that the inequalities hold and move objective function into a constraint:

\[
\begin{align*}
\text{minimize} & \quad x_{n+1} \\
\text{over all} & \quad x \in S \text{ and } x_{n+1} \in \mathbb{R} \\
\text{satisfying} & \quad Pr[c_i(x) \leq 0] \geq \alpha_i \text{ for } i = 1, \ldots, m \\
& \quad Pr[c_0(x) \leq x_{n+1}] \geq \alpha_0
\end{align*}
\]

This condenses a random variable into a number – in terms of quantiles, and value-at-risk (VaR). The \(\alpha_i\)-quantile must be \(\leq 0\).

A drawback is that there is no account for the magnitude of constraint violations. When \(c_i(x) > 0\), which occurs with probability \(1 - \alpha_i\), is it a mere inconvenience, or a disaster?

A major flaw in this approach is that it lacks a property called coherency.
Other Approaches

- **Constraint Consolidation**: put all random variables into one, such as \( c(x, \omega) = \max\{c_1(x, \omega), \ldots, c_m(x, \omega)\} \) and then use a chance constraint on \( c(x) \)

- **Stochastic Programming**: use recourse decisions to position the present to be able to respond well in the future. Stochastic programming traditionally uses expectation, however “improvements may need to be considered.”

- **Dynamic Programming**: focuses on many future stages – perhaps an infinite number of stages – and is concerned with “policies for controlling an uncertain system”

- **Penalty Staircases**: create a penalty function with thresholds, and \( \min E[penalty(c_0(x))] \), or incorporate penalty into chance constraints
Risk is associated with having to make a decision without fully knowing its consequences, due to future uncertainty, but also knowing that some of those consequences might be bad, or at least undesirable relative to others.

1. To many people, risk represents the degree of uncertainty (deviation from a constant) (*measures of deviation*)

2. To other people, risk is a surrogate for overall cost (e.g. mean, median, or worst value) (*measures of the risk of loss*)
“The core of the difficulty in optimization under uncertainty is the fact that a random variable is not, in itself, a single quantity.”

A strategy for optimization is to condense the random variable into a single quantity that measures the risk of loss.

For a random variable $X$, assign a value $R(X)$ to quantify the risk of loss.

(Draw on board the idea of optimizing a tolerance box, as an alternative)
Convert Stochastic Model using $R(X)$

Convert the stochastic model in terms of random variables to a model using $R(X)$:

**Example (Stochastic Model)**

$$\text{minimize } \ c_0(x)$$
$$\text{over all } \ x \in S$$
$$\text{satisfying } \ c_i(x) \leq 0 \text{ for } i = 1, \ldots, m$$

**Example (Risk of Loss Model)**

$$\text{minimize } \ R_0(c_0(x))$$
$$\text{over all } \ x \in S$$
$$\text{satisfying } \ R_i(c_i(x)) \leq 0 \text{ for } i = 1, \ldots, m$$
What properties should $R(X)$ have to be a good quantifier of the risk of loss?
$R(X)$ is a coherent measure of risk in the extended sense if:

- (R1) $R(C) = C$ for all constants $C$,
- (R2) $R((1 - \lambda)X + \lambda X') \leq (1 - \lambda)R(X) + \lambda R(X)$ for $\lambda \in (0, 1)$ convexity
- (R3) $R(X) \leq R(X')$ when $X \leq X'$ monotonicity
- (R4) $R(X) \leq 0$ when $||X^k - x||_2 \to 0$ with $R(X^k) \leq 0$ closedness

and is a coherent measure of risk in the basic sense if it also satisfies:

- (R5) $R(\lambda X) = \lambda R(X)$ for $\lambda > 0$ positive homogeneity
Property (R1), \( R(C) = C \) for all constants \( C \), implies that \( R(0) = 0 \), and condenses a random variable with the same outcome \( C \) to a single surrogate value of \( C \).

Properties (R1) and (R2) imply that

\[
R(X + C) = R(X) + C
\]

Property (R3), \( R(X) \leq R(X') \) when \( X \leq X' \), provides an ordering on risk of loss. If \( X(\omega) \leq X'(\omega) \) almost surely in the future states \( \omega \), the risk of loss seen in \( X \) should not exceed the risk of loss seen in \( X' \), with respect to \( R \).
Properties (R2) with (R5) lead to subadditivity,

\[ R(X + X') \leq R(X) + R(X') \]

which is a key property lacking in approaches popular in finance

Interpretation: when \( X \) and \( X' \) are loss variables for two different portfolios, the total risk of loss should be reduced, or at least not made worse, when the portfolios are combined into one

This refers to diversification, or risk-pooling in supply chain

Property (R2) says that forming a weighted combination of two portfolios should not increase overall loss potential. Otherwise, there might be something “gained by partitioning a portfolio into increasingly smaller fractions”
A common complaint among academics is that VaR is not subadditive. That means the VaR of a combined portfolio can be larger than the sum of the VaRs of its components. To a practicing risk manager this makes sense. For example, the average bank branch in the United States is robbed about once every ten years. A single-branch bank has about 0.004% chance of being robbed on a specific day, so the risk of robbery would not figure into one-day 1% VaR. It would not even be within an order of magnitude of that, so it is in the range where the institution should not worry about it, it should insure against it and take advice from insurers on precautions.
The whole point of insurance is to aggregate risks that are beyond individual VaR limits, and bring them into a large enough portfolio to get statistical predictability. It does not pay for a one-branch bank to have a security expert on staff. As institutions get more branches, the risk of a robbery on a specific day rises to within an order of magnitude of VaR. At that point it makes sense for the institution to run internal stress tests and analyze the risk itself. It will spend less on insurance and more on in-house expertise. For a very large banking institution, robberies are a routine daily occurrence. Losses are part of the daily VaR calculation, and tracked statistically rather than case-by-case. A sizable in-house security department is in charge of prevention and control, the general risk manager just tracks the loss like any other cost of doing business.
As portfolios or institutions get larger, specific risks change from low-probability/low-predictability/high-impact to statistically predictable losses of low individual impact. That means they move from the range of far outside VaR, to be insured, to near outside VaR, to be analyzed case-by-case, to inside VaR, to be treated statistically.

Even VaR supporters generally agree there are common abuses of VaR.
The risk of loss associated with a random variable $X$ is acceptable with respect to a choice of a coherent risk measure $R$ when $R(X) \leq 0$.

Property (R4) says that if a random variable $X$ can be approximated arbitrarily closely by acceptable random variables $X^k$, then $X$ should be acceptable too.
Theorem

Suppose the optimization problem uses $R_i(c_i(x))$ for $i = 0, 1, \ldots, m$ and each $R_i$ is a coherent measure of risk in the extended sense. Then,

- (a) Preservation of convexity
- (b) Preservation of certainty
- (c) Insensitivity to scaling
Approach 1: Guessing the Future

Identify a single element $\bar{\omega} \in \Omega$ as the best guess of the unknown information, and assess the risk in $c_i(x)$ as $R(c_i(x))$ with

$$R(X) = X(\bar{\omega})$$

for some choice of $\bar{\omega}$ having positive probability

This is a coherent measure of risk in the basic sense, but does not incorporate uncertainty into the optimization model.
Approach 2: Worst-Case Analysis

A conservative approach is to prepare for the worst, and assess the risk in $c_i(x)$ as $R(c_i(x))$ with

$$R(X) = \sup X$$

This is a coherent measure of risk in the basic sense, but is a very conservative optimization model.
Use the expectation of the random variable, and assess the risk in $c_i(x)$ as $R(c_i(x))$ with

$$R(X) = \mu[X] = E[X]$$

This is a coherent measure of risk in the basic sense, but requires a long-run to balance out the positive and negative costs over time.
Use the standard deviation in combination with the expected value, and assess the risk in $c_i(x)$ as $R(c_i(x))$ with

$$R_i(X) = \mu[X] + \lambda_i \sigma[X] \text{ for } \lambda_i > 0$$

This is NOT a coherent measure of risk, the monotonicity property (R3) fails. An excellent substitute that preserves convexity is “conditional value-at-risk.”
Use probabilistic or chance constraints, and assess the risk in $c_i(x)$ as $R(c_i(x))$ with

$$R_i(X) = q_{\alpha_i}(X) \text{ for } \alpha_i \in (0, 1)$$

the $\alpha_i$-quantile in the distribution of $X$.

This is NOT a coherent measure of risk, the convexity property (R2) fails. Even though (R5) holds, subadditivity is violated. An excellent substitute that preserves convexity is “conditional value-at-risk.”
Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR)

The value-at-risk and $\alpha$-quantile are identical:

$$VaR_\alpha(X) = q_\alpha(X) = \min\{z | F_X(z) \geq \alpha\}$$

The conditional value-at-risk is:

$$CVaR_\alpha(X) = \text{expectation of } X \text{ in the conditional distribution of its upper } \alpha \text{ tail, so that}$$

$$CVaR_\alpha(X) \geq VaR_\alpha(X) \text{ always}$$

Two other formulas for CVaR:

$$CVaR_\alpha(X) = \frac{1}{1 - \alpha} \int_\alpha^1 VaR_\beta(X) d\beta$$

$$\min_{C \in \mathbb{R}} \left\{ C + (1 - \alpha)^{-1} E[\max\{0, X - C\}] \right\}$$

(Draw graphs on board)
Approach 6: Safeguarding with Conditional Value-at-Risk

CVaR is a coherent measure of risk in the basic sense and behaves better in optimization than VaR.

For a choice of probability levels $\alpha_i \in (0, 1)$ for $i = 0, 1, \ldots, m$,

$$\min_{\alpha_0} \text{CVaR}_{\alpha_0}[c_0(x)]$$

over all $x \in S$

satisfying $\text{CVaR}_{\alpha_i}[c_i(x)] \leq 0$ for $i = 1, \ldots, m$

Approach 6 with CVaR is more cautious than Approach 5 with VaR $\text{CVaR}_{\alpha_i}[c_i(x)] \leq 0$ means not merely that $c_i(x) \leq 0$ at least $100\alpha_i\%$ of the time, but that the average of the worst $100(1 - \alpha_i)\%$ of all outcomes will be $\leq 0$. 
There is a linear programming reformulation to the CVaR model that has many computational advantages.

More on CVaR:

- Mixed CVaR and Spectral Profiles of Risk
- Safeguarding with mixtures of CVaR
- Duality considerations
- Aversity of CVaR and Mixed CVaR
Characterizations of Optimality

For a problem of the form:

\[
\begin{align*}
\text{minimize} & \quad R_0(c_0(x)) \\
\text{over all} & \quad x \in S \\
\text{satisfying} & \quad R_i(c_i(x)) \leq 0 \quad \text{for} \quad i = 1, \ldots, m
\end{align*}
\]

with each \( R_i \) a coherent measure of risk, how can we characterize solutions \( x^* \)?

Beyond the scope of this class, but work with subgradients of \( R_i(c_i(x)) \), develop Lagrange multipliers and duality using the risk envelopes. An active area of research.

Seminar on Thursday by Doug Martin on Utility and Risk
Notes 15: Review of Optimization Modeling

IND E 599

December 6, 2010
Your opportunity to go beyond “text-book” problems. Use optimization models to gain insight into the system. Provide a way to arrive at frequently made decisions, or decisions that are made infrequently but may have high impact.

Outline for presentations and final report:

- First half (7.5 minutes):
  - Project objectives - provide insight into ...? or provide a tool to ...? or inform decisions on ...?
  - Process to achieve those objectives (outline with high level description of models, variations, analyses, what-if’s ...)
  - Models - clearly define variables, objective function and constraints, data and parameters in our standard “math programming” format

- Second half (7.5 minutes)
  - Insights from analyses: provide graphs, figures, tables
  - Conclusion: How the models and analyses achieve the objectives

- For the final report, put data tables and technical details in an appendix.
Review: Optimization Models

Familiarity with common optimization models, such as:

- Facility location,
- Vehicle routing,
- Job shop scheduling,
- Flow shop scheduling,
- Production scheduling (min make span, min max lateness),
- Knapsack/multi-knapsack,
- Traveling Salesman,
- Capacitated Assignment Problem,
- Set covering/packing,
- Network Flow,
- Shortest path, and
- Max flow.
General Optimization Formulations

- Decision Variables
- Objective Function
- Constraints
- Data or parameters

Formulation

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_j(x) \leq b_j, \quad j = 1, \ldots, m_1 \\
& \quad h_j(x) = 0, \quad j = 1, \ldots, m_2 \\
& \quad x_i \geq 0, \quad i = 1, \ldots, n \\
& \quad x_1, x_2 \in \{0, 1\} \quad (\text{binary variables})
\end{align*}
\]

Is a solution feasible? Unique optimum? Multiple optima?
Linear Programming

- Product mix
- Blending
- Mini-max
- Ratio of linear functions
- Piecewise linear functions
- Absolute value functions

When is LP appropriate? Allocation of scarce resources when linearity (proportionality, additivity, divisibility, certainty) holds. Often used for long-range strategic planning, but not for short-term tactical decisions.
Computer Software for Optimization

Basically two categories:

- **Modeling Languages:**
  - AIMMS
  - AMPL (a math programming language)
  - GAMS (general algebraic modeling system)
  - LINDO
  - Matlab
  - Spreadsheets (Excel)

- **Solvers:**
  - ALPHAECP, APOPT, BARON, BDMLP, COIN-OR, CONOPT, CPLEX, DECIS, DICOPT, FortMP, GUROBI, IPOPT, KNITRO, LINDOGLOBAL/LINGO, LGO, MILES, MINOS, MOSEK, MPSGE, MSNLP, NLPEC, OQNLP, PATH, SBB, SCIP, SNOPT, XA, XPRESS
Economic Interpretation

- Sensitivity Analysis and Parametric Programming
- Shadow Prices, Reduced Costs
- Right-hand side Ranging
- Multiple Optima
- Degeneracy, Redundant Constraints
- Visualize in 2 Dimensions
- Extreme point solution (LP)
- Interior point solution (NLP)
Restrict decision variables to be integer (or binary) valued
May change an “easy” to solve LP to an NP-hard IP
Applications include:

- Set-up costs or set-up times, also called fixed-charge
- Logical conditions and disjunctive constraints
  - Either-or constraints
  - At least $K$ out of $N$ constraints satisfied
  - Specific relations using “or” and “and”
- Set covering
- Set packing
- Set partitioning
- Job-shop scheduling
  - Precedence constraints
  - Conflict constraints
  - Due dates
  - Minimize makespan, or minimize maximize tardiness
- Knapsack problem and multi-dimensional knapsack problem
- Cutting stock problem
- Traveling Salesman Problem
Network Type Problems

Many network models are LP’s with guaranteed integer solutions – and many specialized, very efficient algorithms

- Transportation problem
- Assignment problem
- Transshipment problem
- Minimum cost flow problem
- Shortest path problem
- Maximum flow problem
- Critical path method (CPM)
Multi-objective Optimization

- Pareto optimality
- Efficient frontier
- Goal programming
Incorporating Uncertainty

- chance-constrained programming
- scenario-based stochastic programming with recourse
- robust optimization
- sensitivity analysis related to stochastic programming
- coherent approaches to risk in optimization
Next step is to learn more about algorithms and the interaction between algorithms and models. For example, tradeoffs between accuracy and computational efficiency. In your projects, you are already experiencing this interaction.

Thank you for being a part of this course, and best of luck to you in the future, as you are now an “optimizer”.