AM/ES 121: Intro to Optimization Models and Methods

Lesson Plan

• Review syllabus, schedule
  – Course website: am121.seas.harvard.edu

• Overview of topics (with motivating examples)
  – Linear Programming
  – Integer Programming
  – Stochastic Programming
Mathematical Programming

• Set of methods to model and solve optimization problems with continuous or integer variables

• Programming ≠ computer programming
• Programming = Planning

Mathematical Programming

- Real-world problems
  - Model
    - LP
    - IP
    - MDP
  - Method
    - Simplex
    - Branch-and-Bound

• We emphasize modeling
• Methods with elegant math
• Important applications
Prominent Applications

- Optimal crew scheduling saves American airlines 20 million dollars per year
- Traffic control of Hanshin Expressway in Osaka, Japan saves 17 million driver hours per year
- Google PageRank— a tool for indexing the Web –is defined by a Markov chain.
- Clearing Nationwide Kidney Exchanges, potentially saving hundreds of lives.

Campaign stops: The problem of the travelling politician

By WILLIAM COOK (Georgia Tech)

- What's the quickest way to hit every county in Iowa?
The Shortest Possible Baseball Roadtrip

• Ben Blatt (class of 2013)
  
  – http://www.wsj.com/articles/SB10001424052702303657404576357
  560436672964
  
  – https://harvardsportsanalysis.wordpress.com/2011/06/03/roadtrip/
Syllabus

• **Prerequisite:** AM 21b or Math 21b will help with the linear algebra in the first 1/3 of the course

• **Recommended Text:** “Operations Research: Models and Methods”, by Jensen and Bard.

• **Optional text:** “AMPL”, by Fourer, Gay and Kernighan

• **Attend sections!**

• **A lot going on at the beginning of the course!**

• Piazza
  – All announcements will be on Piazza
  – Ask questions publicly whenever appropriate

Syllabus

• 7 problem sets (mixture of modeling, computational exercises, and theory). Encourage use of Matlab/Mathematica
  – late days
• Two “Extreme optimization” team assignments
  – what’s that?
  (Extreme values: communication, simplicity, feedback, courage, respect)
• 2 in-class midterms; No final.
• Grade breakdown is (roughly) 30% problem sets, 20% extreme optimization, 50% midterms
Syllabus

• Collaboration policy
  – Students are strongly encouraged to collaborate in planning and thinking through solutions, but must write up their own solutions without checking over their written solution with another student.
  – Do not pass solutions to problem sets nor accept them from another student. (It’s NOT OK to get solutions to previous years’ problem sets.)

• Undergraduate engineering students, particularly S.B. concentrators, should enroll under ES 121. This will ensure the course count as an engineering elective.

Get Ready for the Course

• Know where the course website locates http://am121.seas.harvard.edu/
• Complete Assignment 0
  – Sign up for Piazza
  – Install AMPL
  – Complete an online sectioning form
• Check out the Course Logistics Overview document on the Resources page of the course website.
Rough Schedule

• 6 weeks on linear programming
• 4 weeks on integer programming
• 2 weeks stochastic programming
• interlude of applications:
  – 2 extreme optimization projects
  – 1 guest lecture on cool applications!
• 2 breakout modeling sessions

Optimization

• The problem of making decisions to maximize or minimize an objective, maybe in the presence of complicating constraints
  – Decision variables
  – Objective function
  – Constraints
A Very Simple Optimization Problem

- minimize $y = (x-2)^2 + 3$

- minimize $y = (x-2)^2 + 3$
  s.t. $x \geq 3$

In this course, we focus on linear models.

A Linear Program

$$\min \ z = 50x_1 + 100x_2$$

s.t.  
$$7x_1 + 2x_2 \geq 28$$
$$2x_1 + 12x_2 \geq 24$$
$$x_1, x_2 \geq 0$$
\[
\begin{align*}
\text{min } z &= 50x_1 + 100x_2 \\
\text{s.t. } 7x_1 + 2x_2 &\geq 28 \\
2x_1 + 12x_2 &\geq 24 \\
x_1, x_2 &\geq 0
\end{align*}
\]

\[5 \times (1) : 35x_1 + 10x_2 \geq 140\]
\[7.5 \times (2) : 15x_1 + 90x_2 \geq 180\]
\[+: 50x_1 + 100x_2 \geq 320\]

graphical version of problem
(solution is \(x_1 = 3.6, x_2 = 1.4\), value 320)
Distributing Goods through a Distribution Network

- Decision variables: Amount to ship on seven shipping lanes, $x_{F1-F2}$, $x_{F1-DC}$, $x_{F1-W1}$, $x_{F2-DC}$, $x_{DC-W2}$, $x_{W1-W2}$, $x_{W2-W1}$.
- Objective: Minimize the total shipping cost
- Constraints
  - Amount shipped in – amount shipped out = required amount
  - Amount shipped on a shipping lane does not exceed its capacity
  - Amount shipped on any shopping lane is nonnegative
Produce Animal Feed Mix

- Produce the feed mix that satisfy the nutrition requirements with minimal cost
- The mixture must meet:
  - Calcium: at least 0.8% but not more than 1.2%
  - Protein: at least 22%
  - Fiber: at most 5%

- Nutrient Contents and Costs of Ingredients

<table>
<thead>
<tr>
<th>Ingredient</th>
<th>Calcium</th>
<th>Protein</th>
<th>Fiber</th>
<th>Unit cost (cents/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limestone</td>
<td>0.38</td>
<td>0.0</td>
<td>0.0</td>
<td>10.0</td>
</tr>
<tr>
<td>Corn</td>
<td>0.001</td>
<td>0.09</td>
<td>0.02</td>
<td>30.5</td>
</tr>
<tr>
<td>Soybean meal</td>
<td>0.002</td>
<td>0.5</td>
<td>0.08</td>
<td>90.0</td>
</tr>
</tbody>
</table>

What we’ll cover for LP

- Modeling: Given a problem, formulate the linear programming mathematical model.

- Solve the formulated LPs
  - Simplex method
  - Sensitivity analysis
  - Duality
Integer Programming (IP)

- Decision variables can only take integer values
- Linear programming model + integrality constraints
- Eg.

\[
\begin{align*}
\text{max} & \quad 5x_1 + 8x_2 \\
\text{s.t.} & \quad x_1 + x_2 \leq 6 \\
& \quad 5x_1 + 9x_2 \leq 45 \\
& \quad x_1, \quad x_2 \geq 0, \text{ integer}
\end{align*}
\]

LP vs. IP

- LP
  - Continuous decision variables
  - Potentially larger feasible space
- IP
  - Discrete decision variables – have to take integral values
  - Potentially smaller feasible space
  - Generally harder to solve
- The natural idea of rounding LP solution for IP does not work in general.
\[
\begin{align*}
\text{max } x_1 + 0.64x_2 \\
50x_1 + 31x_2 & \leq 250 \\
3x_1 - 2x_2 & \geq -4 \\
x_1, x_2 & \in \mathbb{Z}_+
\end{align*}
\]

Sudoku

\begin{center}
\begin{tabular}{c|c|c|c|c}
  6 & 1 & 4 & 5 \\
  8 & 3 & 5 & 6 \\
  2 &   &   & 1 \\
  8 & 4 & 7 & 6 \\
  6 &   & 3 &   \\
  7 & 9 & 1 & 4 \\
  5 &   &   & 2 \\
  7 & 2 & 6 & 9 \\
  4 & 5 & 8 & 7 \\
\end{tabular}
\end{center}
Sudoku

• Decision variables
  – \( b_{i,j}^v \) : equals 1 if cell at row \( i \) and column \( j \) has value \( v \), otherwise 0.

• Constraints
  – Each row must have one 1 ... 9
  – Each column must have one 1 ... 9
  – Each 3-by-3 section must have one 1 ... 9
  – Each cell must have a value from 1 ... 9

• Objective
  – Anything

The Assignment Problem

• There are \( n \) people available to carry out \( n \) jobs
• Each person has to be assigned to exactly one job
• Person \( i \) if assigned to job \( j \) incurs cost \( c_{i,j} \)
• Goal: find a minimum cost assignment
What we’ll cover for IP

• Modeling: Given a problem, formulate the integer programming mathematical model.

• Solve the formulated IPs
  — Branch-and-Bound
  — Cutting plane
  — Branch-and-cut
  — advanced solution techniques

Introducing Uncertainty

• Both LP and IP are deterministic, with known model parameters

• But many real world problems have uncertainty at the time a decision needs to be made

• Stochastic programming is an approach for modeling optimization problems that involve uncertainty
What we’ll cover for stochastic programming

- Markov chains (to model the underlying stochastic process)
- Markov decision processes
- Two-stage stochastic optimization

Markov Chains in One Slide

- A stochastic process (a sequence of random variables) that has Markov property
- Markov property: state of the process at time \( t + 1 \) only depends on the state at time \( t \) (history does not matter!)
- Eg. Machine condition

<table>
<thead>
<tr>
<th>State</th>
<th>Condition</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Good as new</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Minor deterioration</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>Major deterioration</td>
<td>2000</td>
</tr>
<tr>
<td>3</td>
<td>Inoperable</td>
<td>6000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>7/8</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3/4</td>
<td>1/8</td>
<td>1/8</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Markov Decision Processes (MDP)

• Rather than passively observing the Markov chain, we can proactively make a decision about which action to take at a state to affect the transition probability

• The decision process of choosing the optimal action is referred as a Markov Decision Process

A MDP Example: Machine Maintenance

<table>
<thead>
<tr>
<th>State</th>
<th>Condition</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Good as new</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Minor deterioration</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>Major deterioration</td>
<td>2000</td>
</tr>
<tr>
<td>3</td>
<td>Inoperable</td>
<td>6000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>7/8</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3/4</td>
<td>1/8</td>
<td>1/8</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision</th>
<th>Action</th>
<th>Relevant States</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Do nothing</td>
<td>0, 1, 2</td>
</tr>
<tr>
<td>2</td>
<td>Overhaul (return the system to state 1)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Replace (return the system to state 0)</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>

Goal: find a maintenance plan that minimizes cost
What to expect after the course

• Be able to transform a real world problem into a mathematical programming model
• Master the techniques to solve the formulated mathematical programming model
• Be able to solve the problem using commercial software packages, CPLEX/AMPL
• Feel empowered!