Solving Business Problems with OPTMODEL and SAS

Pre-Conference Workshop
INFORMS Annual Meeting
Austin, TX

OR Center of Excellence
Advanced Analytics Division
SAS Institute

November 6, 2010
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Background on SAS

- Founded in 1976
- World’s largest privately held software company
- Over 400 offices globally
- Over 45,000 customer sites in 118 countries
- SAS customers or affiliates: 92 of top 100 companies in 2009 Fortune Global 500
- Hundreds of local user groups
SAS Data Sets, Language, Terminology

- SAS data set: tabular data file
  - Rows are "observations"
  - Columns are "variables"
  - Created with SAS data step and/or SQL
  - Read from / write to all major data formats

- SAS programming language base: data handling and exploration, elementary statistics, wide range of functions

- Software modules are procedures, abbreviated "PROC"
### Sample SAS Data Set

<table>
<thead>
<tr>
<th>desig</th>
<th>category</th>
<th>nengs</th>
<th>engtype</th>
<th>ab</th>
<th>ncrew</th>
<th>equip</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-130H</td>
<td>Attack&amp;Obs</td>
<td>4</td>
<td>turboprop</td>
<td>n</td>
<td>14</td>
<td>weapons</td>
</tr>
<tr>
<td>A-10</td>
<td>Attack&amp;Obs</td>
<td>2</td>
<td>turboprop</td>
<td>n</td>
<td>1</td>
<td>weapons</td>
</tr>
<tr>
<td>OA-10</td>
<td>Attack&amp;Obs</td>
<td>2</td>
<td>turboprop</td>
<td>n</td>
<td>1</td>
<td>weapons</td>
</tr>
<tr>
<td>B1-B</td>
<td>Bomber</td>
<td>4</td>
<td>turbofan</td>
<td>n</td>
<td>4</td>
<td>bomb_bay</td>
</tr>
<tr>
<td>B-52G</td>
<td>Bomber</td>
<td>8</td>
<td>turbojet</td>
<td>y</td>
<td>5</td>
<td>bomb_bay</td>
</tr>
<tr>
<td>B-52H</td>
<td>Bomber</td>
<td>8</td>
<td>turbojet</td>
<td>n</td>
<td>5</td>
<td>bomb_bay</td>
</tr>
<tr>
<td>F-15A/C</td>
<td>Fighter</td>
<td>2</td>
<td>turbofan</td>
<td>y</td>
<td>1</td>
<td>weapons</td>
</tr>
<tr>
<td>F-15B/D</td>
<td>Fighter</td>
<td>2</td>
<td>turbofan</td>
<td>y</td>
<td>2</td>
<td>weapons</td>
</tr>
<tr>
<td>F-16A/B</td>
<td>Fighter</td>
<td>1</td>
<td>turbofan</td>
<td>y</td>
<td>1</td>
<td>weapons</td>
</tr>
<tr>
<td>F-16C/D</td>
<td>Fighter</td>
<td>1</td>
<td>turbofan</td>
<td>y</td>
<td>2</td>
<td>weapons</td>
</tr>
<tr>
<td>F-111A/E</td>
<td>Fighter</td>
<td>2</td>
<td>turbofan</td>
<td>y</td>
<td>2</td>
<td>bomb_rks</td>
</tr>
<tr>
<td>F-111D</td>
<td>Fighter</td>
<td>2</td>
<td>turbofan</td>
<td>y</td>
<td>2</td>
<td>bomb_rks</td>
</tr>
<tr>
<td>C-5</td>
<td>Trans&amp;Tank</td>
<td>4</td>
<td>turbofan</td>
<td>n</td>
<td>6</td>
<td>none</td>
</tr>
<tr>
<td>C-9C</td>
<td>Trans&amp;Tank</td>
<td>2</td>
<td>turbofan</td>
<td>n</td>
<td>7</td>
<td>none</td>
</tr>
<tr>
<td>C-12</td>
<td>Trans&amp;Tank</td>
<td>2</td>
<td>turboprop</td>
<td>n</td>
<td>2</td>
<td>none</td>
</tr>
<tr>
<td>C-17</td>
<td>Trans&amp;Tank</td>
<td>4</td>
<td>turbofan</td>
<td>n</td>
<td>3</td>
<td>none</td>
</tr>
<tr>
<td>C-20A/B</td>
<td>Trans&amp;Tank</td>
<td>2</td>
<td>turbofan</td>
<td>n</td>
<td>5</td>
<td>none</td>
</tr>
<tr>
<td>C-21A</td>
<td>Trans&amp;Tank</td>
<td>2</td>
<td>turbofan</td>
<td>n</td>
<td>2</td>
<td>none</td>
</tr>
<tr>
<td>C-130</td>
<td>Trans&amp;Tank</td>
<td>4</td>
<td>turboprop</td>
<td>n</td>
<td>5</td>
<td>none</td>
</tr>
<tr>
<td>KC-135</td>
<td>Trans&amp;Tank</td>
<td>4</td>
<td>turbofan</td>
<td>n</td>
<td>5</td>
<td>none</td>
</tr>
<tr>
<td>C-141B</td>
<td>Trans&amp;Tank</td>
<td>4</td>
<td>turbofan</td>
<td>n</td>
<td>6</td>
<td>none</td>
</tr>
<tr>
<td>KC-10A</td>
<td>Trans&amp;Tank</td>
<td>3</td>
<td>turbofan</td>
<td>n</td>
<td>4</td>
<td>none</td>
</tr>
<tr>
<td>T-1A</td>
<td>Trainer</td>
<td>2</td>
<td>turbofan</td>
<td>n</td>
<td>3</td>
<td>none</td>
</tr>
<tr>
<td>T-3A</td>
<td>Trainer</td>
<td>1</td>
<td>propeller</td>
<td>n</td>
<td>2</td>
<td>none</td>
</tr>
<tr>
<td>T-37B</td>
<td>Trainer</td>
<td>2</td>
<td>turbojet</td>
<td>n</td>
<td>2</td>
<td>none</td>
</tr>
<tr>
<td>T-37C</td>
<td>Trainer</td>
<td>2</td>
<td>turbojet</td>
<td>y</td>
<td>2</td>
<td>weapons</td>
</tr>
<tr>
<td>T-38A</td>
<td>Trainer</td>
<td>2</td>
<td>turbojet</td>
<td>n</td>
<td>2</td>
<td>none</td>
</tr>
<tr>
<td>T-38B</td>
<td>Trainer</td>
<td>2</td>
<td>turbojet</td>
<td>y</td>
<td>2</td>
<td>weapons</td>
</tr>
</tbody>
</table>
SAS Data Sets and External Data Formats

- SAS has several ways (programmatic, point and click) to convert any data in tabular form into a SAS data set
  - Data step functionality
  - PROC IMPORT
  - Enterprise Guide

- Through SAS ACCESS engines, can create SAS views to any commonly used database
  - Data appears as SAS data sets to SAS, but remains in the native database format
  - SAS can write results back to the native database format
Overview of SAS Functions

- Mathematical, Arithmetic, Truncation
- Array
- Character
- Date and Time
- Financial
- Probability Distributions
- Trigonometric, Hyperbolic
- Random Number Generation
- Sample Statistics, Quantiles
SAS Products and Solutions

■ SAS products: breadth and depth
  » Data Integration: access, cleansing, metadata creation
  » Business Intelligence: query, reporting, visualization
  » Analytics: statistics, forecasting, data mining, text mining, quality control, optimization, scheduling, simulation

■ SAS solutions
  » Cross-Industry: Customer Intelligence, Performance Management, Supply Chain Intelligence, Risk Management
  » Industry-Specific: Retail, Oil and Gas, Energy and Utilities, Financial Services, Life Sciences, Insurance, Communications
SAS Products and Solutions

- SAS products: breadth and depth
  - Data Integration: access, cleansing, metadata creation
  - Business Intelligence: query, reporting, visualization
  - Analytics: statistics, forecasting, data mining, text mining, quality control, optimization, scheduling, simulation

- SAS solutions
  - Cross-Industry: Customer Intelligence, Performance Management, Supply Chain Intelligence, Risk Management
  - Industry-Specific: Retail, Oil and Gas, Energy and Utilities, Financial Services, Life Sciences, Insurance, Communications
SAS Presentations at INFORMS Austin: Sunday, Nov. 7

- SA40: Interfaces and Metaheuristics in COIN-OR
  - "Modeling Cone Optimization Problems with COIN OS" (I. Polik)

- SA48: Software Demonstrations
  - "Enterprise Guide Demonstration" (T. Bohannon)

- SC48: Software Demonstrations
  - "Dynamic Visualization with JMP" (M. Stephens)
SAS Presentations at INFORMS Austin: Monday, Nov. 8

- MA10: Models of Strategic Consumer Behavior
  » "The Perils of Commitments" (A. Parlakturk, M. Kabul)

- MA20: New Frontiers in Revenue Mgmt Applications
  » "Hotel Demand Modeling and Data Challenges" (A. Chien)

- MB33: Optimization Applications in Retail
  » "Optimization in Retail Pricing: Common Themes and Variations" (G. Ege)
  » "Regular Price Optimization in Retail Pricing" (R. Solanki, S. Gupta)
SAS Presentations at INFORMS Austin: Monday, Nov. 8

- MC11: Operations Management and Marketing
  - "Strategic Customers and Commitments in a Decentralized Supply Chain" (A. Parlakturk, M. Kabul)

- MD32: Computational Linear and Integer Programming
  - "SAS MILP Solver Developments" (A. Narisetty, P. Christophel, Y. Xu)

- MD40: Recent Developments in COIN-OR Projects
  - "Progress Update on COIN/DIP–Decomposition for Integer Programming" (M. Galati, T. Ralphs)
SAS Presentations at INFORMS Austin: Tuesday, Nov. 9

- TB18: OR/MS Software II
  - "Teaching OR and Analytics to Future Practitioners Using SAS On-Demand" (E. Hughes, M. Chari)
- TD46: Panel Discussion: Skills and Career Paths in Industry
  - Panelists: J. Camm, G. Ege, S. Kuppa, J. Levis, D. Samuelson
- TD48: Software Demonstrations
  - "Building and Solving Optimization Models with SAS" (E. Hughes, R. Pratt)
SAS Presentations at INFORMS Austin: Wednesday, Nov. 10

- WA57: Recent Advances in Portfolio Optimization
  - "Optimal Portfolio Strategy and Liquidity Capacity" (W. Chen)
- WA65: Inventory Management V
  - "A Multiple-Level Supply Chain Coordination Model by (Q,r) Policies in a Fuzzy Environment" (X. Wu, D. Warsing)
The Scope of SAS/OR

- Mathematical Optimization
- Genetic Algorithms
- Constraint Programming
- Discrete Event Simulation
- Project and Resource Scheduling
- Other areas of OR/MS
Broad Range of Optimization Solvers

- Linear Programming
  - Primal Simplex
  - Dual Simplex
  - Interior Point

- Mixed-Integer Linear Programming:
  - Branch and Bound with heuristics, cutting planes

- Quadratic Programming:
  - Interior Point

- Nonlinear Programming:
  - Active Set
  - Interior Point
SAS Revenue Optimization Suite:
SAS Regular Price Optimization

![SAS Merchandise Intelligence](image)

**Plan Details**
- **Name:** 007 Scenic Big Bags - Max Rev
- **Products:** 126
- **Locations:** 8

**Plan Settings/Optimization**
- **Goal:** Maximize Revenue ($)
- **Status:** Optimized [11/12/2007 5:04 PM]
- **Target units:** Default price change priority: Normal
- **Target start:** 09/25/2005
- **End:** 01/25/2006

**Plan Members**

<table>
<thead>
<tr>
<th>SKU</th>
<th>SKU Name</th>
<th>Market Count</th>
<th>Current Cost ($)</th>
<th>Current Price ($)</th>
<th>New Price ($)</th>
<th>Fix Price</th>
<th>Rule Violation Type Count</th>
<th>Override Competitive Price Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1130972</td>
<td>PL SHRIMP CRACKERS 4 oz</td>
<td>8</td>
<td>0.79</td>
<td>1.09</td>
<td>0.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6002705</td>
<td>PL SPICY 6 oz</td>
<td>8</td>
<td>1.29</td>
<td>1.99</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>1379945</td>
<td>PL SPICY BREADSTICKS 6 oz</td>
<td>8</td>
<td>1.19</td>
<td>2.09</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6002959</td>
<td>PL SWEET CHILLS CHIPS 6 oz</td>
<td>8</td>
<td>1.59</td>
<td>1.79</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6262183</td>
<td>PL Sweet Potato Chips 6 oz</td>
<td>8</td>
<td>1.19</td>
<td>1.99</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>1300178</td>
<td>PL TMT &amp; B&amp;B MND 6 oz</td>
<td>8</td>
<td>1.39</td>
<td>2.49</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6205305</td>
<td>PL TTD DUCK &amp; HOSEN SAUCE ...</td>
<td></td>
<td>1.39</td>
<td>2.49</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6025749</td>
<td>PL TTD OLIVE OIL SEA SALT/B...</td>
<td></td>
<td>0.79</td>
<td>1.99</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6205746</td>
<td>PL TTD OLIVE OIL SYL/POM ...</td>
<td></td>
<td>1.19</td>
<td>1.99</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6025244</td>
<td>PL TTD SEA SALT &amp; CIDER VGR...</td>
<td></td>
<td>1.39</td>
<td>2.49</td>
<td>2.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>1304710</td>
<td>PL TTD VEGETABLE CHIPS 4 oz</td>
<td>8</td>
<td>0.69</td>
<td>1.09</td>
<td>0.79</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6373398</td>
<td>PRINGLES AROMAS FETA 8 oz</td>
<td>8</td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6373374</td>
<td>PRINGLES AROMAS SPICY 8 oz</td>
<td>8</td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>4159952</td>
<td>PRINGLES BACON 6 oz</td>
<td>8</td>
<td>2.59</td>
<td>3.19</td>
<td>3.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>2433802</td>
<td>PRINGLES CHEESE &amp; ONION 8 oz</td>
<td>8</td>
<td>2.99</td>
<td>3.99</td>
<td>3.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6017189</td>
<td>PRINGLES DIPPER CHIPS ORG...</td>
<td></td>
<td>1.99</td>
<td>2.79</td>
<td>3.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>6037199</td>
<td>PRINGLES DIPPERS SOUR CREA...</td>
<td></td>
<td>1.99</td>
<td>2.99</td>
<td>3.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>3597040</td>
<td>PRINGLES HOT &amp; SPICY 6 oz</td>
<td>8</td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>3323281</td>
<td>PRINGLES LIGHT ORIGINAL 6 oz</td>
<td>8</td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>3323240</td>
<td>PRINGLES LIGHT SOUR CREAM ...</td>
<td></td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>4204049</td>
<td>PRINGLES MEXICAN LET 6 oz</td>
<td>8</td>
<td>2.18</td>
<td>2.99</td>
<td>3.29</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>2390799</td>
<td>PRINGLES ORIGINAL POTATO ...</td>
<td></td>
<td>2.99</td>
<td>3.79</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>3260001</td>
<td>PRINGLES PAPRIKA 8 oz</td>
<td>8</td>
<td>2.99</td>
<td>3.79</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>2334608</td>
<td>PRINGLES SALT &amp; VINEGAR 8 oz</td>
<td>8</td>
<td>3.49</td>
<td>4.19</td>
<td>3.99</td>
<td></td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>

**Plan Performance**
- **Regular Price Optimization**
- **sasdemo**
- **112051**
SAS Revenue Optimization Suite: SAS Promotion Optimization
SAS Revenue Optimization Suite: SAS Markdown Optimization
SAS Credit Scoring for Banking

Application Scorecard Performance Report

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Score Year</th>
<th>Score Month Name</th>
<th>Cumulative Events</th>
<th>Fraction of Events</th>
<th>Cumulative Non Events</th>
<th>Fraction of Non Events</th>
<th>False Positive Rate</th>
<th>False Negative Rate</th>
<th>KS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Card Application Existing Customers Above 30</td>
<td>2004</td>
<td>January</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score Range</td>
<td>2004</td>
<td>January</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-99</td>
<td>0.02</td>
<td>0.0989</td>
<td>0.01</td>
<td>0.0989</td>
<td>0.0989</td>
<td>0.0989</td>
<td>0.0989</td>
<td>0.0989</td>
<td></td>
</tr>
<tr>
<td>100-199</td>
<td>0.03</td>
<td>0.1526</td>
<td>0.03</td>
<td>0.1526</td>
<td>0.1526</td>
<td>0.1526</td>
<td>0.1526</td>
<td>0.1526</td>
<td></td>
</tr>
<tr>
<td>200-299</td>
<td>0.04</td>
<td>0.1989</td>
<td>0.04</td>
<td>0.1989</td>
<td>0.1989</td>
<td>0.1989</td>
<td>0.1989</td>
<td>0.1989</td>
<td></td>
</tr>
<tr>
<td>300-399</td>
<td>0.05</td>
<td>0.2447</td>
<td>0.05</td>
<td>0.2447</td>
<td>0.2447</td>
<td>0.2447</td>
<td>0.2447</td>
<td>0.2447</td>
<td></td>
</tr>
<tr>
<td>400-499</td>
<td>0.06</td>
<td>0.2849</td>
<td>0.06</td>
<td>0.2849</td>
<td>0.2849</td>
<td>0.2849</td>
<td>0.2849</td>
<td>0.2849</td>
<td></td>
</tr>
<tr>
<td>500-599</td>
<td>0.07</td>
<td>0.3239</td>
<td>0.07</td>
<td>0.3239</td>
<td>0.3239</td>
<td>0.3239</td>
<td>0.3239</td>
<td>0.3239</td>
<td></td>
</tr>
<tr>
<td>600-699</td>
<td>0.08</td>
<td>0.3629</td>
<td>0.08</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.3629</td>
<td></td>
</tr>
<tr>
<td>700-799</td>
<td>0.09</td>
<td>0.4009</td>
<td>0.09</td>
<td>0.4009</td>
<td>0.4009</td>
<td>0.4009</td>
<td>0.4009</td>
<td>0.4009</td>
<td></td>
</tr>
<tr>
<td>800-899</td>
<td>0.10</td>
<td>0.4389</td>
<td>0.10</td>
<td>0.4389</td>
<td>0.4389</td>
<td>0.4389</td>
<td>0.4389</td>
<td>0.4389</td>
<td></td>
</tr>
<tr>
<td>900-1000</td>
<td>0.11</td>
<td>0.4769</td>
<td>0.11</td>
<td>0.4769</td>
<td>0.4769</td>
<td>0.4769</td>
<td>0.4769</td>
<td>0.4769</td>
<td></td>
</tr>
</tbody>
</table>

Copyright ©2010, SAS Institute Inc. All rights reserved.
SAS Risk Management for Banking

Ad Hoc Reports Wizard - page 2 of 2

Select values for the cross classification variables and click "Finish" to run the report.

*PRODUCT_TYPE
Available:
- BERMUDA_OPTION
- BOND
- BOND_OPTION
- CAP_OPTION
- CASH
- CASHFLOW
- CASHFLOW_FORWARD
- CASHFLOW_ZERO

Selected:
- BERMUDA_OPTION
- BOND
- BOND_OPTION

*ALTYPE
Available:
- ASSET

Selected:
- ASSET

Apply

<Previous Next Finish Cancel>
SAS Social Network Analysis
OPTMODEL: Major Features

- Algebraic modeling language with optimization-oriented syntax:
  - Variables, constraints, bounds, objective
  - Algebraic expression of functions
  - Parameters, variables, arrays, index sets
- Tight integration with SAS programming environment
- Flexible input/output: read from and create any number of data sets
- Interactivity
- Direct access to all solvers (LP, MILP, QP, NLP)
OPTMODEL Programming Statements

- **Control and Looping:**
  - DO, IF, etc.
  - DO iterative, DO UNTIL, DO WHILE, etc.

- **Input/Output:**
  - READ DATA, CREATE DATA: working with SAS data sets
  - SAVE MPS, SAVE QPS
  - FILE, PRINT, PUT

- **Model Management:**
  - SOLVE
  - PROBLEM, USE PROBLEM (named models)
  - DROP/RESTORE (constraints)
  - FIX/UNFIX (variables)
  - EXPAND (selective or overall)
OPTMODEL Expressions Overview

- Aggregation operators (evaluate a set expression over an index set)
  - SUM, PROD, MAX, MIN
  - INTER aggregation, UNION aggregation
  - Boolean: AND, OR

- Set operators
  - IN, WITHIN
  - INTER, UNION, SYMDIFF, DIFF
  - CROSS, SLICE

- Conditional
  - IF-THEN-ELSE
  - Scalar or set expression
Basics of the SAS Macro Language

- Macros: parameterized blocks of SAS code
- Macro variables:
  - Assigning a value: `%let varname=<value>;
  - Using: &varname
- Declaring a macro:
  ```sas
  %macro macroname (param1, param2);
  <SAS code>
  %mend macroname;
  ```
- Invoking a macro: `%macroname (<value1>, <value2>);`
- Special code within macros:
  ```sas
  %if &varname=<value> %then %do; ...%end;
  %do n=1 %to &number;...%end;
  ```
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Shortest Path Problem

- Given directed network \((N, A)\) with source \(s\) and sink \(t\)
- Each arc \((i, j) \in A\) has cost \(c_{ij}\)
- Find shortest path from \(s\) to \(t\)
- Minimum cost network flow problem, sending one unit of flow from \(s\) to \(t\)
- Several approaches:
  - LP
  - MILP
  - Multiple objectives
  - QP
  - DP
  - NLP
Example Network
Optimal Solution
Shortest Path Data and Setup

OPTMODEL code
ShortestPath.sas
minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]
subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = b_i \quad i \in N \]
\[ x_{ij} \geq 0 \quad (i, j) \in A \]

where \( b_s = 1 \), \( b_t = -1 \), and \( b_i = 0 \) for \( i \in N \setminus \{s, t\} \)
Linear Programming

minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s,t\} \end{cases} \quad i \in N \]

\[ x_{ij} \geq 0 \quad (i,j) \in A \]

```plaintext
var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
  sum {<(i),j> in ARCS} Flow[i,j]
  - sum {<j,(i)> in ARCS} Flow[j,i]
  = (if i = source then 1 else if i = sink then -1 else 0);
```
minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s,t\} \end{cases} \]

\[ x_{ij} \geq 0 \quad (i,j) \in A \]

\begin{verbatim}
var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
    sum {<(i),j> in ARCS} Flow[i,j] - sum {<j,(i)> in ARCS} Flow[j,i] = (if i = source then 1 else if i = sink then -1 else 0);
\end{verbatim}
minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{if } i \in N \setminus \{s,t\} 
\end{cases} 
\quad i \in N \]

\[ x_{ij} \geq 0 \]

\begin{verbatim}
var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
    sum {<(i),j> in ARCS} Flow[i,j]
    - sum {<j,(i)> in ARCS} Flow[j,i]
    = (if i = source then 1 else if i = sink then -1 else 0);
\end{verbatim}
minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s,t\} \end{cases} \quad i \in N \]
\[ x_{ij} \geq 0 \quad (i,j) \in A \]

var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
  sum {<(i),j> in ARCS} Flow[i,j]
  - sum {<j,(i)> in ARCS} Flow[j,i]
  = (if i = source then 1 else if i = sink then -1 else 0);
minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{if } i \in N \setminus \{s,t\} 
\end{cases} \quad i \in N \]
\[ x_{ij} \geq 0 \quad (i,j) \in A \]

var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
  sum {<(i),j> in ARCS} Flow[i,j] - sum {<j,(i)> in ARCS} Flow[j,i] = (if i = source then 1 else if i = sink then -1 else 0);
Linear Programming

minimize
\[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to
\[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s,t\} \end{cases} \quad i \in N \]
\[ x_{ij} \geq 0 \quad (i,j) \in A \]

```plaintext
var Flow {ARCS} >= 0;
min TotalCost = sum {<i,j> in ARCS} cost[i,j] * Flow[i,j];
con balance {i in NODES}:
  sum {<(i),j> in ARCS} Flow[i,j]
  - sum {<j,(i)> in ARCS} Flow[j,i]
  = (if i = source then 1 else if i = sink then -1 else 0);
```
Linear Programming

OPTMODEL code
ShortestPathLP.sas
Mixed Integer Linear Programming

- Given directed network \((N, A)\) with source \(s\) and sink \(t\)
- Each arc \((i, j) \in A\) has cost \(c_{ij}\) and traversal time \(t_{ij}\)
- Find shortest path from \(s\) to \(t\) with side constraint on total time
- Constraint matrix not totally unimodular
- Need to impose integrality on flow variables
Mixed Integer Linear Programming

minimize

\[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to

\[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{if } i \in N \setminus \{s, t\} 
\end{cases} \quad i \in N \]

\[ \sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \]

\[ x_{ij} \in \{0, 1\} \]  \quad (i, j) \in A
Example Network

(c = 1, t = 1)

(1, 10) (2, 3) (1, 7)

(1, 2) (5, 7) (10, 1)

(10, 3) (12, 3) (2, 2)

Copyright ©2010, SAS Institute Inc. All rights reserved.
Optimal Solution for $T = 14$

Network diagram with edges labeled:
- (1, 10)
- (1, 2)
- (10, 3)
- (12, 3)
- (c = 1, t = 1)
- (2, 3)
- (5, 7)
- (10, 1)
- (2, 2)
Mixed Integer Linear Programming

OPTMODEL code
ShortestPathMILP.sas
Multiple Objectives

- Minimize total cost (primary objective)
- Add constraint restricting total cost to be at most twice this minimum
- Minimize total time (secondary objective)
- More generally, can have any number of objectives, any combination of max and min
Multiple Objectives

OPTMODEL code
ShortestPathMultipleObjectives.sas
Inverse Shortest Path Problem

- Given directed network \((N, A)\) with source \(s\) and sink \(t\)
- Each arc \((i, j) \in A\) has cost \(c_{ij}\)
- Given path \(P\) from \(s\) to \(t\)
- Find minimum perturbation of costs so that \(P\) is shortest path

Applications:
  - Telecommunications
  - Traffic flow
  - Seismic tomography
Example Network

```
1 --- 2 --- 4
<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
1 --- 3 --- 5
<table>
<thead>
<tr>
<th></th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>
1 --- 3 --- 5
<table>
<thead>
<tr>
<th></th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
1 --- 6
```

10

10
Given Path $P$
Optimal Perturbed Costs
Linear Programming Duality

minimize \[ \sum_{(i,j) \in A} c_{ij}x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{if } i \in N \setminus \{s, t\} 
\end{cases} \quad i \in N \]

\[ x_{ij} \geq 0 \quad (i,j) \in A \]

maximize \[ \pi_s - \pi_t \]

subject to \[ \pi_i - \pi_j \leq c_{ij} \quad (i,j) \in A \]

\[ \pi_i \text{ free} \quad i \in N \]
Linear Programming Duality

minimize \[ \sum_{(i,j) \in A} c_{ij} x_{ij} \]

subject to \[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \]
\[ x_{ij} \geq 0 \]

maximize \[ \pi_s - \pi_t \]

subject to \[ \pi_i - \pi_j \leq c_{ij} \]
\[ \pi_i \text{ free} \]
\[ i \in N \]
Quadratic Programming

minimize \[ \sum_{(i,j) \in A} (c'_{ij} - c_{ij})^2 \]

subject to

\[ \pi_i - \pi_j = c'_{ij} \quad (i, j) \in P \]
\[ \pi_i - \pi_j \leq c'_{ij} \quad (i, j) \in A \setminus P \]
\[ \pi_i \text{ free} \quad i \in N \]
\[ c'_{ij} \geq 0 \quad (i, j) \in A \]
Dynamic Programming

- OPTMODEL can be used as programming language
  - No need to call optimization solver
  - Rich variety of programming statements and set operations
- Dynamic Programming
  - Various ways to solve shortest path problem as DP
  - For example, Dijkstra’s algorithm
Dijkstra’s Algorithm

- Optimality conditions (Bellman’s equation):
  \[ d_i = \min_{(i,j) \in A} \{c_{ij} + d_j\} \text{ for } i \in N \]
- With \( d_t = 0 \), can interpret \( d_i \) as distance to sink
- Initialize distances: \( d_i = \infty \text{ for } i \neq t \)
- Repeat until source is permanently labeled:
  1. Find temporarily labeled node \( j \) with smallest label
  2. Permanently label \( j \) and update distances for other nodes \( i \):
     \[ d_i = \min(c_{ij} + d_j, d_i) \]
Dynamic Programming

OPTMODEL code

ShortestPathDP.sas
Nonlinear Programming

- *Minimum Pressure* Network Problem
  - No longer *shortest path* (flow is dispersed)
  - Interpretation: hydraulic networks
  - Conservation of mass and energy
  - Two state variables: pressure $p_i$ (nodes) and flow $x_{ij}$ (arcs)
  - Nonlinearities: power, pressure loss, drag
Example Network
Nonlinear Programming

Sources of nonlinearity:

- **Power**: $p_0 x_{0,1}$

- **Pressure loss**:
  \[ p_j = p_i - \left( \frac{\ell_{ij}}{2D_{ij}^3} \right) f_{ij} x_{ij}^2 \]

- **Drag** (Colebrook, $Re > 4 \times 10^4$):
  \[ \frac{1}{\sqrt{f_{ij}}} = -2.0 \log \left( \frac{\epsilon/D}{3.7} + \frac{2.51 \mu}{x_{ij} \sqrt{f_{ij}}} \right) \]
Sources of nonlinearity:

- **Power**: $p_0 x_{0,1}$
- **Pressure loss**:
  \[
  p_j = p_i - \left( \frac{\ell_{ij}}{2D_{ij}^3} \right) f_{ij} x_{ij}^2
  \]

- **Drag** (Colebrook, $Re > 4 \times 10^4$):
  \[
  \frac{1}{\sqrt{f_{ij}}} = -2.0 \log \left( \frac{e/D}{3.7} + \frac{2.51 \mu}{x_{ij} \sqrt{f_{ij}}} \right)
  \]
Nonlinear Programming

Sources of nonlinearity:

- **Power:** $p_0 x_{0,1}$
- **Pressure loss:**

$$p_j = p_i - \left( \frac{\ell_{ij}}{2D_{ij}^3} \right) f_{ij} x_{ij}^2$$

- **Drag (Colebrook, $Re > 4 \times 10^4$):**

$$\frac{1}{\sqrt{f_{ij}}} = -2.0 \log \left( \frac{\epsilon/D}{3.7} + \frac{2.51 \mu}{x_{ij} \sqrt{f_{ij}}} \right)$$
Nonlinear Programming

minimize \( p_0 x_{0,1} \)

subject to \( \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = 0 \quad i \in N \)

\( p_j = p_i - \delta_{ij} \quad (i, j) \in A \)

\( \delta_{ij} = \left( \frac{\ell_{ij}}{2 D_{ij}^3} \right) f_{ij} x_{ij}^2 \quad (i, j) \in A \)

\( \frac{1}{\sqrt{f_{ij}}} = -2.0 \log \left( \frac{\epsilon/D}{3.7} + \frac{2.51 \mu}{x_{ij} \sqrt{f_{ij}}} \right) \quad (i, j) \in A \)

\( f_{ij} > 0, \quad \delta_{ij} \text{ free} \quad (i, j) \in A \)

\( p_i = P_{\text{atm}} \quad i \in N \setminus N \)

\( \delta_{ij} \geq \delta_{LO}, \quad x_{ij} \geq D_j \quad (i, j) \in A \setminus A \)
minimize $p_0 x_{0,1}$

subject to

\[ \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = 0 \quad i \in N \]

\[ p_j = p_i - \delta_{ij} \quad (i, j) \in A \]

\[ \delta_{ij} = \left( \frac{\ell_{ij}}{2D_{ij}^3} \right) f_{ij} x_{ij}^2 \quad (i, j) \in A \]

\[ \frac{1}{\sqrt{f_{ij}}} = -2.0 \log \left( \frac{\epsilon/D}{3.7} + \frac{2.51 \mu}{x_{ij} \sqrt{f_{ij}}} \right) \quad (i, j) \in A \]

\[ f_{ij} > 0, \quad \delta_{ij} \text{ free} \quad (i, j) \in A \]

\[ p_i = P_{\text{atm}} \quad i \in \mathcal{N} \setminus N \]

\[ \delta_{ij} \geq \delta_{\text{LO}}, \quad x_{ij} \geq D_j \quad (i, j) \in \mathcal{A} \setminus \mathcal{A} \]
Example Network

1 -- 2 -- 4 -- 6
|     |     |     |
| 6   | 3   | 50  |

1 -- 3 -- 5 -- 6
|     |     |     |
| 25  | 60  |      |
Optimal Pressures (atm)

1.19

1.06

1.03

1.00

50

25

1.07

1.01

60
Optimal Flows (L/s)

145  
19  
74  
71  
25  
26  
39  
47  
44  
5  
32  
3  
5  
18  
60  
50

Copyright ©2010, SAS Institute Inc. All rights reserved.
Minimum Pressure Problem

OPTMODEL code
MinPressureNLP.sas
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Multicommodity Flow Problem

- Given directed network \((N, A)\) and commodities \(K\)
- Each node \(i \in N\) has supply \(b_i^k\) for commodity \(k \in K\)
- Each arc \((i, j) \in A\) has:
  - cost \(c_{ij}^k\) per unit of flow of commodity \(k \in K\)
  - capacity \(u_{ij}^k\) for commodity \(k \in K\)
  - capacity \(u_{ij}\) that limits sum of flows of commodities
- Satisfy all supplies and demands at minimum cost, while respecting arc capacities
- Two approaches:
  - Direct LP
  - Dantzig-Wolfe Decomposition
Example Network with Two Commodities

\[(c = 1, u = 5)\]

10
1
---
3
---
2
-10

(5, 30)

(1, 10)

3
---
4

(1, 30)

(5, 30)

5
---
6

(1, 30)

(5, 30)

20

-20
Optimal Solution

(c = 1, u = 5)
minimize \[ \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \]

subject to \[ \sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \]

\[ \sum_{(i,j) \in A} x_{ij}^k - \sum_{(j,i) \in A} x_{ji}^k = b_i^k \quad i \in N, \quad k \in K \]

\[ 0 \leq x_{ij}^k \leq u_{ij}^k \quad (i, j) \in A, \quad k \in K \]
Multicommodity Flow Problem: Direct LP

**OPTMODEL code**

MulticommodityFlow.sas
Constraint Matrix for Example Network

\[
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\]
Block-Angular Structure

$$
\begin{pmatrix}
A_1 & A_2 & \cdots & A_{|K|} \\
B_1 & & & \\
& B_2 & & \\
& & \ddots & \\
& & & B_{|K|}
\end{pmatrix}
$$
Linear Programming Formulation

minimize \[ \sum_{k \in K} \sum_{(i, j) \in A} c_{ij}^k x_{ij}^k \]

subject to \[ \sum_{k \in K} x_{ij}^k \leq u_{ij} \quad (i, j) \in A \] (1)

\[ \sum_{(i, j) \in A} x_{ij}^k - \sum_{(j, i) \in A} x_{ji}^k = b_i^k \quad i \in N, \quad k \in K \] (2)

\[ 0 \leq x_{ij}^k \leq u_{ij}^k \quad (i, j) \in A, \quad k \in K \] (3)

Let \( X^k = \{ x^k \in \mathbb{R}^{|A|} : x^k \text{ satisfies (2)-(3)} \} \) be the set of feasible flows with respect to commodity \( k \).
Dantzig-Wolfe Decomposition

minimize \[ \sum_{k \in K} c^k x^k \]

subject to \[ \sum_{k \in K} (x^k)_{ij} \leq u_{ij} \quad (i, j) \in A \]
\[ x^k \in X^k \quad k \in K \]

- Let \( \{x^k_p\} \) \( p \in P^k \) be all extreme points of \( X^k \) (\( p \) is a proposal)
- Express \( x^k \in X^k \) as convex combination of \( x^k_p \)
- \( x^k = \sum_{p \in P^k} \lambda^k_p x^k_p \), where \( \sum_{p \in P^k} \lambda^k_p = 1 \) and \( \lambda^k_p \geq 0 \)
- \( c^k_p = c^k x^k_p \) is cost of proposal \( p \in P^k \)
- \( f^k_{pij} = (x^k_p)_{ij} \) is flow of commodity \( k \) along \( (i, j) \) for \( p \in P^k \)
Dantzig-Wolfe Decomposition

\[
\text{minimize} \quad \sum_{k \in K} c^k x^k \\
\text{subject to} \quad \sum_{k \in K} (x^k)_{ij} \leq u_{ij} \quad (i, j) \in A \\
\quad x^k \in X^k \quad k \in K
\]

- Let \( \{x^k_p\}_{p \in P^k} \) be all extreme points of \( X^k \) (\( p \) is a proposal)
- Express \( x^k \in X^k \) as convex combination of \( x^k_p \)
- \( x^k = \sum_{p \in P^k} \lambda^k_p x^k_p \), where \( \sum_{p \in P^k} \lambda^k_p = 1 \) and \( \lambda^k_p \geq 0 \)
- \( c^k_p = c^k x^k_p \) is cost of proposal \( p \in P^k \)
- \( f^k_{pij} = (x^k_p)_{ij} \) is flow of commodity \( k \) along \( (i, j) \) for \( p \in P^k \)
Master Problem

minimize \[
\sum_{k \in K} \sum_{p \in P^k} c_p^k \lambda_p^k
\]

subject to \[
\sum_{k \in K} \sum_{p \in P^k} f_{p,ij}^k \lambda_p^k \leq u_{ij} \quad (i,j) \in A
\]
\[
\sum_{p \in P^k} \lambda_p^k = 1 \quad k \in K
\]
\[
\lambda_p^k \geq 0 \quad k \in K, \ p \in P^k
\]
Restricted Master Problem

minimize \( \sum_{k \in K} \sum_{p \in \bar{P}^k} c_p^k \lambda_p^k \)

subject to \( \sum_{k \in K} \sum_{p \in \bar{P}^k} f_{p,ij}^k \lambda_p^k \leq u_{ij} \) \((i, j) \in A\)

\( \sum_{p \in \bar{P}^k} \lambda_p^k = 1 \) \( k \in K \)

\( \lambda_p^k \geq 0 \) \( k \in K, \ p \in \bar{P}^k \)

Replace \( P^k \) with small subset \( \bar{P}^k \)
Restricted Master Problem

$$\begin{align*}
\text{minimize} & \quad \sum_{k \in K} \sum_{p \in \bar{P}^k} c_p^k \lambda_p^k \\
\text{subject to} & \quad \sum_{k \in K} \sum_{p \in \bar{P}^k} f_{pij}^k \lambda_p^k \leq u_{ij} \quad (i, j) \in A \\
 & \quad \sum_{p \in \bar{P}^k} \lambda_p^k = 1 \quad k \in K \\
 & \quad \lambda_p^k \geq 0 \quad k \in K, \ p \in \bar{P}^k
\end{align*}$$

Dual variables $v_{ij}$ and $w_k$ used to price out columns
Subproblems

Given $v_{ij}$ and $w_k$, find, for each $k \in K$, column with negative reduced cost (if one exists)

minimize \[ \sum_{(i,j) \in A} (c_{ij}^k - v_{ij})x_{ij}^k - w_k \]
subject to \[ \sum_{(i,j) \in A} x_{ij}^k - \sum_{(j,i) \in A} x_{ji}^k = b_i^k \quad i \in N \]
\[ 0 \leq x_{ij}^k \leq u_{ij}^k \quad (i, j) \in A \]

Single commodity network flow
Dynamic Column Generation

- Dual of dynamic generation of subtour elimination constraints in TSP
- Extreme points too numerous to enumerate explicitly
- Generate negative reduced cost columns as needed
- Column generation problem separable by commodity and has network structure
- Bounds on optimal objective throughout algorithm
Multicommodity Flow Problem: Dantzig-Wolfe Decomposition

OPTMODEL code
MulticommodityFlow.sas
Commodity 1, Proposal 1: Cost = 110
Commodity 2, Proposal 1: Cost = 100

\[(c = 1, u = 5)\]

\[(5, 30)\] (1, 10) (1, 30) (1, 30) (5, 30) 20 20 20 20
Commodity 1, Proposal 2: Cost = 60
Commodity 2, Proposal 2: Cost = 80

(c = 1, u = 5)
Optimal Solution: Cost = 150

(c = 1, u = 5)
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Business Problem

- Determine schedule for allocation of cash inventory at branch banks to service ATMs
- Define a polynomial fit for predicted cash flow need per ATM per day
- Predictive model factors include:
  - days of week
  - weeks of month
  - holidays
  - salary disbursement days
  - locations of branches
- Cash allocation plans finalized at beginning of month, deviations from plan are costly
Business Problem

- Optimization problem: determine multipliers for fit to minimize mismatch based on predicted withdrawals
- Constraints:
  - Regulatory agencies enforce minimum cash reserve ratio at branch banks (per day)
  - For each ATM, limit on number of cashout days based on predictive model (customer satisfaction)
Data

- $A$ is set of ATMs
- $D$ is set of dates
- $\text{normAve}_{ad}$ is normal average
- $\text{normStd}_{ad}$ is normal standard deviation
- $\text{impactAve}_{ad}$ is impact average
- $\text{impactStd}_{ad}$ is impact standard deviation
- $\text{po}_{ad}$ is period 1 value
- $\text{pt}_{ad}$ is period 2 value
- $\text{with}_{ad}$ is actual withdrawal
- $\text{alloc}_{d}$ is daily allocation
- $\text{clim}_a$ is limit on number of cashouts allowed
- $\text{ima}_{ad} = \text{normAve}_{ad} \text{ impactAve}_{ad}$
- $\text{ims}_{ad} = \text{normStd}_{ad} \text{ impactStd}_{ad}$
Mixed Integer Nonlinear Programming

- **Nonconvex** quadratically constrained integer problem
- Linearize absolute value, add binaries for count constraints
- NEOS and other MINLP solvers fail to solve

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} \sum_{d \in D} |f_{ad} - \text{with}_{ad}| \\
\text{subject to} & \quad (p_t_{ad}(1 - x_{1a}) + p_o_{ad}x_{1a})(1 - x_{2a}) + \\
& \quad \text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \\
& \quad \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \\
& \quad \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A \\
& \quad x_{1a}, x_{2a} \in [0, 1] \quad a \in A \\
& \quad x_{3a} \geq 0 \quad a \in A
\end{align*}
\]
Mixed Integer Nonlinear Programming

- **Nonconvex** quadratically constrained integer problem
- Linearize absolute value, add binaries for count constraints
- NEOS and other MINLP solvers fail to solve

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} \sum_{d \in D} |f_{ad} - \text{with}_{ad}| \\
\text{subject to} & \quad (p_{td}(1 - x_{1a}) + p_{o_{ad}x_{1a}})(1 - x_{2a}) + \\
& \quad \text{i}_{a_{ad}}x_{2a} + \text{i}_{m_{ad}}x_{3a} = f_{ad} \quad a \in A, \ d \in D \\
& \quad \sum_{a \in A} f_{ad} \leq \text{alloc}_{d} \quad d \in D \\
& \quad \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_{a} \quad a \in A \\
& \quad x_{1a}, x_{2a} \in [0, 1] \quad a \in A \\
& \quad x_{3a} \geq 0 \quad a \in A
\end{align*}
\]
Mixed Integer Nonlinear Programming

- **Nonconvex** quadratically constrained integer problem
- Linearize absolute value, add binaries for count constraints
- NEOS and other MINLP solvers fail to solve

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} \sum_{d \in D} |f_{ad} - \text{with}_{ad}| \\
\text{subject to} & \quad (p_{t_{ad}}(1-x_{1a}) + p_{o_{ad}}x_{1a})(1-x_{2a}) + \\
& \quad \text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \\
& \quad \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \\
& \quad \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A \\
& \quad x_{1a}, x_{2a} \in [0, 1] \quad a \in A \\
& \quad x_{3a} \geq 0 \quad a \in A
\end{align*}
\]
Linearization of Absolute Value

minimize \[ \sum_{a \in A} \sum_{d \in D} |f_{ad} - \text{with}_{ad}| \]

subject to \[ \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \]

\[ (p_{tad}(1 - x_{1a}) + p_{o_{ad}}x_{1a})(1 - x_{2a}) + \]
\[ \text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \]
\[ \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A \]
\[ x_{1a}, x_{2a} \in [0, 1] \quad a \in A \]
\[ x_{3a} \geq 0 \quad a \in A \]
Linearization of Absolute Value

minimize $|y|$  
subject to $Ay \leq b$  

$\Rightarrow$

minimize $y^+ + y^-$  
subject to $A(y^+ - y^-) \leq b$  
$y^+ \geq 0$  
$y^- \geq 0$

$$y = y^+ - y^-$$
$$|y| = y^+ + y^-$$
Linearization of Absolute Value

\[
\begin{align*}
\text{minimize} & \quad |y| \\
\text{subject to} & \quad Ay \leq b
\end{align*}
\]  \iff \quad \begin{align*}
\text{minimize} & \quad y^+ + y^- \\
\text{subject to} & \quad A(y^+ - y^-) \leq b \\
& \quad y^+ \geq 0 \\
& \quad y^- \geq 0
\end{align*}

In optimal solution,

\[
\begin{align*}
y^+ & \geq 0 \text{ and } y^- = 0 \quad \Rightarrow \quad y = y^+ \\
y^+ & = 0 \text{ and } y^- \geq 0 \quad \Rightarrow \quad y = -y^-
\end{align*}
\]
Linearization of Absolute Value

minimize \[ \sum_{a \in A} \sum_{d \in D} |f_{ad} - \text{with}_{ad}| \]

subject to \[ \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \]

\[ (\text{pt}_{ad}(1 - x_{1a}) + \text{po}_{ad}x_{1a})(1 - x_{2a}) + \]
\[ \text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \quad d \in D \]

\[ \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A \]

\[ x_{1a}, \quad x_{2a} \in [0, 1] \quad a \in A \]
\[ x_{3a} \geq 0 \quad a \in A \]
Linearization of Absolute Value

minimize

\[ \sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right) \]

subject to

\[ \sum_{a \in A} f_{ad} \leq alloc_d \quad d \in D \]

\[ (pt_{ad}(1 - x_{1a}) + po_{ad}x_{1a})(1 - x_{2a}) + \]

\[ \text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \]

\[ f_{ad} - \text{with}_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, \ d \in D \]

\[ \| \{d \in D : f_{ad} < \text{with}_{ad}\} \|_0 \leq \text{clima} \quad a \in A \]

\[ x_{1a}, \ x_{2a} \in [0, 1] \quad a \in A \]

\[ x_{3a} \geq 0 \quad a \in A \]

\[ y_{ad}^+, \ y_{ad}^- \geq 0 \quad a \in A, \ d \in D \]
Linearization of Products

minimize
\[ \sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right) \]

subject to
\[ \sum_{a \in A} f_{ad} \leq alloc_d \quad d \in D \]
\[ (pt_{ad}(1 - x_{1a}) + po_{ad}x_{1a})(1 - x_{2a}) + \]
\[ ima_{ad}x_{2a} + ims_{ad}x_{3a} = f_{ad} \quad a \in A, \; d \in D \]
\[ f_{ad} - with_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, \; d \in D \]
\[ \| \{ d \in D : f_{ad} < with_{ad} \} \|_0 \leq clim_a \quad a \in A \]
\[ x_{1a}, \; x_{2a} \in [0, 1] \quad a \in A \]
\[ x_{3a} \geq 0 \quad a \in A \]
\[ y_{ad}^+, \; y_{ad}^- \geq 0 \quad a \in A, \; d \in D \]
minimize \[ \sum_{a \in A} \sum_{d \in D} \left( y^+_{ad} + y^-_{ad} \right) \]

subject to \[ \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \]

\[ (\text{ima}_{ad} - \text{pt}_{ad})x_{2a} + \text{pt}_{ad} + (\text{po}_{ad} - \text{pt}_{ad})x_{1a} + \]

\[ (\text{pt}_{ad} - \text{po}_{ad})x_{1a}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, d \in D \]

\[ f_{ad} - \text{with}_{ad} = y^+_{ad} - y^-_{ad} \quad a \in A, d \in D \]

\[ \|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A \]

\[ x_{1a}, x_{2a} \in [0, 1] \quad a \in A \]

\[ x_{3a} \geq 0 \quad a \in A \]

\[ y^+_{ad}, y^-_{ad} \geq 0 \quad a \in A, d \in D \]
Discretization

- Let \( x_{1a} \in \{0.00, 0.05, 0.10, \ldots, 0.95, 1.00\} \)
- Let \( S = \{0, 1, 2, \ldots, 20\} \) and \( w_s = s/20 \)

\[
x_{1a} = \sum_{s \in S} w_s x_{1a}^s
\]

\[
\sum_{s \in S} x_{1a}^s = 1 \quad \Rightarrow \quad \sum_{s \in S} w_s z_a^s = \sum_{s \in S} w_s x_{1a}^s x_{2a}
\]

\( x_{1a}^s \in \{0, 1\} \forall s \in S \)

\( z_a^s = x_{1a}^s x_{2a} \forall s \in S \)
Linearization of Products

\[ z_a^s = x_{1a}^s x_{2a} \]
\[ x_{1a}^s \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]

\[ \sum_{s \in S} z_a^s = x_{2a} \]
\[ \sum_{s \in S} x_{1a}^s = 1 \]
\[ 0 \leq z_a^s \leq x_{1a}^s \]
\[ x_{1a}^s \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]
Linearization of Products

\[ z_a^s = x_{1a}^s x_{2a} \]
\[ x_{1a}^s \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]

\[ \sum_{s \in S} z_a^s = x_{2a} \]
\[ \sum_{s \in S} x_{1a}^s = 1 \]

\[ 0 \leq z_a^s \leq x_{1a}^s \]
\[ x_{1a}^s \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]

when \( x_{1a}^s = 0 \), we have \( z_a^s = 0 \) in both cases
Linearization of Products

\[ z^s_a = x^s_{1a} x_{2a} \]
\[ x^s_{1a} \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]
\[ \sum_{s \in S} z^s_a = x_{2a} \]
\[ \sum_{s \in S} x^s_{1a} = 1 \]
\[ 0 \leq z^s_a \leq x^s_{1a} \]
\[ x^s_{1a} \in \{0, 1\} \]
\[ x_{2a} \in [0, 1] \]

when \( x^s_{1a} = 1 \), we have \( z^s_a = x_{2a} \) in both cases
Linearization of Products

\[ x_{1a}x_{2a} = \sum_{s \in S} w_s z_s^a \]

with

\[ \sum_{s \in S} z_s^a = x_{2a} \]
\[ \sum_{s \in S} x_{1a}^s = 1 \]
\[ 0 \leq z_s^a \leq x_{1a}^s \quad s \in S \]
\[ x_{1a}^s \in \{0, 1\} \quad s \in S \]
\[ x_{2a} \in [0, 1] \]
Linearization of Products

minimize

$$\sum_{a \in A} \sum_{d \in D} \left( y^+_{ad} + y^-_{ad} \right)$$

subject to

$$\sum_{a \in A} f_{ad} \leq alloc_d \quad d \in D$$

$$(ima_{ad} - pt_{ad})x_{2a} + pt_{ad} + (po_{ad} - pt_{ad})x_{1a} +$$

$$(pt_{ad} - po_{ad})x_{1a}x_{2a} + ims_{ad}x_{3a} = f_{ad} \quad a \in A, d \in D$$

$$f_{ad} - with_{ad} = y^+_{ad} - y^-_{ad} \quad a \in A, d \in D$$

$$\|\{d \in D : f_{ad} < with_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A$$

$$x_{1a}, x_{2a} \in [0, 1] \quad a \in A$$

$$x_{3a} \geq 0 \quad a \in A$$

$$y^+_{ad}, y^-_{ad} \geq 0 \quad a \in A, d \in D$$
Linearization of Products

minimize \[ \sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right) \]

subject to

\[ \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \]

\[ (\text{ima}_{ad} - \text{pt}_{ad})x_{2a} + \text{pt}_{ad} + (\text{po}_{ad} - \text{pt}_{ad}) \sum_{s \in S} w_s x_{1a}^s + \]

\[ (\text{pt}_{ad} - \text{po}_{ad}) \sum_{s \in S} w_s z_a^s + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \]

\[ f_{ad} - \text{with}_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, \ d \in D \]

\[ \| \{ d \in D : f_{ad} < \text{with}_{ad} \} \|_0 \leq \text{clim}_a \quad a \in A \]

\[ \sum_{s \in S} z_a^s = x_{2a} \quad a \in A \]

\[ \sum_{s \in S} x_{1a}^s = 1 \quad a \in A \]

\[ 0 \leq z_a^s \leq x_{1a}^s \quad a \in A, \ s \in S \]

\[ x_{1a}^s \in \{0, 1\} \quad a \in A, \ s \in S \]

\[ x_{2a} \in [0, 1] \quad a \in A \]

\[ x_{3a} \geq 0 \quad a \in A \]

\[ y_{ad}^+, \ y_{ad}^- \geq 0 \quad a \in A, \ d \in D \]
Cashout Counting Constraints

Want number of times that withdrawals exceed allocation at most \( \text{clim}_a \). Recall that

\[
(\text{pt}_{ad}(1 - x_{1a}) + \text{po}_{ad}x_{1a})(1 - x_{2a}) + \\
\text{ima}_{ad}x_{2a} + \text{ims}_{ad}x_{3a} = f_{ad} \quad a \in A, \ d \in D \\
f_{ad} - \text{with}_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, \ d \in D \\
\|\{d \in D : f_{ad} < \text{with}_{ad}\}\|_0 \leq \text{clim}_a \quad a \in A
\]

At optimality,

- \( y_{ad}^- > 0 \) implies withdrawal exceeds allocation
- \( y_{ad}^- = 0 \) implies withdrawal does not exceed allocation
- \( y_{ad}^- \leq \text{with}_{ad} \)
Cashout Counting Constraints

Let

- $v_{ad} = 1$ if ATM $a$ has a cashout on date $d$, 0 otherwise
- $M_{ad}$ be an upper bound for optimal $y_{ad}^-$, i.e. with $ad$

$$
\sum_{d \in D} v_{ad} \leq \text{clim}_a \quad a \in A
$$

$$
y_{ad}^- \leq M_{ad} v_{ad} \quad a \in A, d \in D
$$

$$
v_{ad} \in \{0, 1\} \quad a \in A, d \in D
$$

$$
y_{ad}^- \geq 0 \quad a \in A, d \in D
$$
Mixed Integer Linear Programming

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right) \\
\text{subject to} & \quad \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \\
& \quad (\text{ima}_{ad} - \text{pt}_{ad}) x_{2a} + \text{pt}_{ad} + (\text{po}_{ad} - \text{pt}_{ad}) \sum_{s \in S} w_s x_{1a}^s + \\
& \quad (\text{pt}_{ad} - \text{po}_{ad}) \sum_{s \in S} w_s z_s^a + \text{ims}_{ad} x_{3a} = f_{ad} \quad a \in A, \ d \in D \\
& \quad f_{ad} - \text{with}_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, \ d \in D \\
& \quad \sum_{d \in D} v_{ad} \leq \text{clim}_a \quad a \in A \\
& \quad y_{ad}^- \leq \text{M}_{ad} v_{ad} \quad a \in A, \ d \in D \\
& \quad v_{ad} \in \{0, 1\} \quad a \in A, \ d \in D \\
& \quad \sum_{s \in S} z_s^a = x_{2a} \quad a \in A \\
& \quad \sum_{s \in S} x_{1a}^s = 1 \quad a \in A \\
& \quad 0 \leq z_s^a \leq x_{1a}^s \quad a \in A, \ s \in S \\
& \quad x_{1a}^s \in \{0, 1\} \quad a \in A, \ s \in S \\
& \quad x_{2a} \in [0, 1], \ x_{3a} \geq 0 \quad a \in A \\
& \quad y_{ad}^+, \ y_{ad}^- \geq 0 \quad a \in A, \ d \in D
\end{align*}
\]
Mixed Integer Linear Programming

minimize

$$\sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right)$$

subject to

$$\sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D$$

$$(\text{ima}_{ad} - \text{pt}_{ad})x_{2a} + \text{pt}_{ad} + (\text{po}_{ad} - \text{pt}_{ad}) \sum_{s \in S} w_sx_{1a}^s +$$

$$(\text{pt}_{ad} - \text{po}_{ad}) \sum_{s \in S} w_sz_a^s + \text{im}_s \sum_{a \in A} x_{3a} = f_{ad} \quad a \in A, \ d \in D$$

$$f_{ad} - \text{with}_ad = y_{ad}^+ - y_{ad}^- \quad a \in A, \ d \in D$$

$$\sum_{d \in D} v_{ad} \leq \text{clim}_a \quad a \in A$$

$$y_{ad}^- \leq M_{ad}v_{ad} \quad a \in A, \ d \in D$$

$$v_{ad} \in \{0, 1\} \quad a \in A, \ d \in D$$

$$\sum_{s \in S} z_a^s = x_{2a} \quad a \in A$$

$$\sum_{s \in S} x_{1a}^s = 1 \quad a \in A$$

$$0 \leq z_a^s \leq x_{1a}^s \quad a \in A, \ s \in S$$

$$x_{1a}^s \in \{0, 1\} \quad a \in A, \ s \in S$$

$$x_{2a} \in [0, 1], \ x_{3a} \geq 0 \quad a \in A$$

$$y_{ad}^+, \ y_{ad}^- \geq 0 \quad a \in A, \ d \in D$$
Mixed Integer Linear Programming

\[
\begin{align*}
\text{minimize} & \quad \sum_{a \in A} \sum_{d \in D} \left( y_{ad}^+ + y_{ad}^- \right) \\
\text{subject to} & \quad \sum_{a \in A} f_{ad} \leq \text{alloc}_d \quad d \in D \\
& \quad \sum_{s \in S} (\text{ima}_{ad} - \text{pt}_{ad})x_{2a} + \text{pt}_{ad} + (\text{po}_{ad} - \text{pt}_{ad}) \sum_{s \in S} w_s x_{1a}^s + \sum_{s \in S} w_s z_a^s + \text{ims}_{ad} x_{3a} = f_{ad} \quad a \in A, d \in D \\
& \quad f_{ad} - \text{with}_{ad} = y_{ad}^+ - y_{ad}^- \quad a \in A, d \in D \\
& \quad \sum_{d \in D} v_{ad} \leq \text{clim}_a \quad a \in A \\
& \quad y_{ad}^- \leq M_{ad} v_{ad} \quad a \in A, d \in D \\
& \quad v_{ad} \in \{0, 1\} \quad a \in A, d \in D \\
& \quad \sum_{s \in S} z_a^s = x_{2a} \quad a \in A \\
& \quad \sum_{s \in S} x_{1a}^s = 1 \quad a \in A \\
& \quad 0 \leq z_a^s \leq x_{1a}^s \quad a \in A, s \in S \\
& \quad x_{1a}^s \in \{0, 1\} \quad a \in A, s \in S \\
& \quad x_{2a} \in [0, 1], \quad x_{3a} \geq 0 \\& a \in A \\
& \quad y_{ad}^+, y_{ad}^- \geq 0 \quad a \in A, d \in D 
\end{align*}
\]
Block-Angular Structure

\[
\begin{pmatrix}
A_1 & A_2 & \cdots & A_{|K|} \\
B_1 \\
B_2 \\
\vdots \\
B_{|K|}
\end{pmatrix}
\]
Dantzig-Wolfe LP Bounds
Price and Branch

- Use Dantzig-Wolfe decomposition to solve master LP
  - Yields LP lower bound
  - Tighter than usual LP relaxation
- Use direct MILP to solve restricted master problem
  - Uses only columns already generated
  - If integer feasible, yields upper bound
- In this case, integrality gap is 0 at root node
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
Business Problem

- Given transactional data (withdrawals, deposits, replenishments) for past 3 months
- Forecasting problem: estimate hourly demand for each ATM for the next month
- Optimization problem: determine which hours to replenish each ATM over the next month to avoid cashouts
- “replenish” means fill to capacity
- “cashout” means ATM inventory < next 4 hours of demand
Business Problem

- Four possible objectives to minimize:
  - Cashout hours
  - Cashout events (consecutive cashout hours at same ATM)
  - Lost demand (in dollars)
  - Number of replenishments

- Budget limits total number of replenishments over the month

- Limit on number of simultaneous replenishments varies throughout the day

- Eligible replenishment hours depend on ATM:
  - all day: 4am-noon, 1pm-11pm
  - overnight: 9pm-7am

- Run replenishment scheduling every two weeks for one-month rolling horizon
Forecasting: SAS Forecast Studio
Forecasting Challenge: Intermittent Demand
Forecasting Challenge: Missing Data
Forecasting Challenge: Abrupt Demand Shifts
PROC OPTMODEL

- Reads forecasted demand for each ATM and period
- Builds mixed integer linear programming (MILP) instance
- Calls MILP solver
- Outputs optimal solution
### Optimization Results and Business Impact

<table>
<thead>
<tr>
<th>Objective</th>
<th>Baseline</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cashout Hours</td>
<td>?</td>
<td>15</td>
</tr>
<tr>
<td>Cashout Events</td>
<td>391</td>
<td>15</td>
</tr>
<tr>
<td>Lost Demand</td>
<td>?</td>
<td>$0</td>
</tr>
<tr>
<td>Number of Replenishments</td>
<td>11,424</td>
<td>9,828</td>
</tr>
</tbody>
</table>

- 2-hr runtimes well within overnight requirements
- *Significantly increased customer satisfaction* (main goal)
- $1.4 million projected annual savings
- Similar results using historical demands
Possible Enhancements

- Account for interest rates (opportunity cost)
- Consider lost revenue from out-of-network fees
- Incorporate forecast uncertainty in optimization
- Provide truck routing (in addition to scheduling)
- Allow partial replenishments (fill to less than capacity)
- Include cash accepting machines
ATM Replenishment Optimization Demo

- Written in Adobe Flash/Flex by technical student
- Calls PROC OPTMODEL code as stored process
- Communication via SAS data sets and SAS macro language

Launch ATM Optimization Demo
Calling SAS from Flex

- Use Flex HTTPService object to transmit data
- Invoke SAS stored processes via HTTP
- Call SAS code from stored process
- Write results to XML webout data type
- Automatically store results in Flex XMLListCollection object
Calling SAS from Flex

```
public function callSAS_Solve():void {
  SAS_Solve.url="http:" + serverPath + "_program=" + progPath + "SolveATM" + authStr + params;
  SAS_Solve.send(); }

<mx:HTTPService id="SAS_Solve"
  resultFormat="e4x"
  result="PostSolution(event)" />
```
Calling SAS from Flex

```xml
<mx:Button id="SolveBtn" label="Solve"
click= "{InitializeBeforeSolution()}
{callSAS_Solve()}
x="1000" y="370"/>

public function callSAS_Solve():void {
SAS_Solve.url="http:" + serverPath + "_program=" + progPath + "SolveATM" + authStr + params;
SAS_Solve.send(); }

<mx:HTTPService id="SAS_Solve"
resultFormat="e4x"
result="PostSolution(event)" />
```
Calling SAS from Flex

```
public function callSAS_Solve():void {
SAS_Solve.url="http:" + serverPath + 
"_program=" + progPath + 
"SolveATM" + authStr + params;
SAS_Solve.send(); }
```

```
<mx:HTTPService id="SAS_Solve"
resultFormat="e4x"
result="PostSolution(event)" />
```
Calling SAS from Flex

```xml
<mx:Button id="SolveBtn" label="Solve"
click="{InitializeBeforeSolution()}
callSAS_Solve()"
x="1000" y="370"/>

public function callSAS_Solve():void {
  SAS_Solve.url="http:" + serverPath +
  "_program=" + progPath +
  "SolveATM" + authStr + params;
  SAS_Solve.send(); }

<mx:HTTPService id="SAS_Solve"
resultFormat="e4x"
result="PostSolution(event)" />
```
public function callSAS_Solve():void{
SAS_Solve.url="http://" + serverPath + 
"_program=" + progPath + "SolveATM" + authStr + params;
SAS_Solve.send(); }
public function callSAS_Solve():void{
SAS_Solve.url="http://" + serverPath + 
"_program=" + progPath + "SolveATM" + authStr + params;
SAS_Solve.send(); }
public function callSAS_Solve():void{
  SAS_Solve.url="http://" + serverPath + "_program=" + progPath + "SolveATM" + authStr + params;
  SAS_Solve.send();
}
Calling SAS from Flex

```latex
<mx:XMLListCollection
id="ATMSolutionNewScenario"
source="{SAS_Solve.lastResult.sol.ROWS}"/>
```
Calling SAS from Flex

```cpp
<mx:XMLListCollection id="ATMSolutionNewScenario" source="{SAS_Solve.lastResult.sol.ROWS}" />
```
```java
@macro createXML;
%do i=1 %to &ATMdata0;
data _null_;  
  set root.&ATMdata&i end=end;
  file _webout;
  array nums[*] _numeric_;  
  array chars[*] _character_;  
  if _n_ = 1 then do;
    %if &i = 1 %then %do; put '<TABLES>'; %end;
    put '<&ATMData&i>"';
    end;
    put '<ROWS>';  
    do i = 1 to dim(chars);
      cvrname=vname(chars(i));
      put '</' cvrname +(-1) '>'; cvrname +(-1) '>'; chars(i) +(-1) '</ cvrname +(-1) '>'; end;
    do j = 1 to dim(nums);
      nvarname=vname(nums(j));
      put '</' nvarname +(-1) '>'; nums(j) +(-1) '</ nvarname +(-1) '>'; end;
    put '</ROWS>';  
    if end then do;
      put '</&ATMData&i>";
    %if &i = &ATMData0 %then %do; put '</TABLES>'; %end;
    end;
    run;

<mx:XMLListCollection id="ATMSolutionNewScenario"
  source="{"SAS_Solve.lastResult.sol.ROWS}"
  />
Calling SAS from Flex

```java
<mx:XMLListCollection
id="ATMSolutionNewScenario"
source="{SAS_Solve.lastResult.sol.ROWS}"
/>
```
Calling SAS from Flex

- Use Flex HTTPService object to transmit data
- Invoke SAS stored processes via HTTP
- Call SAS code from stored process
- Write results to XML webout data type
- Automatically store results in Flex XMLListCollection object
Outline

1. SAS/OR Introduction
2. Shortest Path Problem
3. Multicommodity Flow Problem
4. Customer Case Study: ATM Cash Management
5. Customer Case Study: ATM Replenishment
6. Sample SAS/OR COE Projects
SAS/OR COE Engagement Model

Data & other analytics
- Data collection & cleansing
- Forecasting
- Other analytics

Customer

Data Issues

Problem Description

Model Clarification & Proposals

Data Requirements

Data Tables

COE

Business Problem

Formulation

Solution algorithm

OR Model

OR Model Output

OUTPUT ANALYSIS

Reports
Other Key COE Projects

Sample projects:

- Stochastic optimization
- Pricing and inventory optimization
- Investment portfolio optimization
- Optimal binning for credit risk
- Digital advertising
- Shipping rate optimization
Stochastic Deal Optimization Overview

- **Customer:** Large US Airplane Engine Manufacturer
- **Pricing service contracts of engines ($50B portfolio):**
  - Find minimum yearly price stream to achieve financial metrics
  - Specify price intervals and bounds and maximum number of price changes
  - Replaces current business rule process

- **MILP**

- **SAS tools:**
  - SAS/OR: OPTMODEL, MILP solver
  - Embedded in SAS/FM application
  - Potential for Monte-Carlo simulations from SAS/ETS

- Completely custom (replacing homegrown Excel spreadsheets)
Pricing and Inventory Optimization

- Customer: Large US Retailer
- Product promotion optimization
  - Retail pricing and inventory network
  - Non-convex NLP
  - Discrete side constraints
- Multi-objective NLP
- SAS tools:
  - SAS/OR: OPTMODEL, NLP and MILP solvers
- Customized heuristics
Investment Portfolio Optimization

- Customer: Large US Government Agency
- Investment portfolio optimization
  - Portfolio mix: T-Bills, T-Notes, CDs
  - Suggested investment strategy for a (e.g.) 1-month period
  - Discrete: minimum investment requirements
- MILP
- SAS tools:
  - SAS Forecast Studio
  - SAS/OR: OPTMODEL, MILP solver
- Generalized network flow formulation
Optimal Binning for Credit Scoring

- Customer: Enterprise Miner (SAS Internal)
- Rigorous binning with constraints
  - Monotonicity constraints
  - Nonlinear measures of risk
  - Discrete binning requirements
- MINLP
- SAS tools
  - SAS/OR: OPTMODEL, MILP solver
  - SAS PROC OPTGRAPH
  - SAS Enterprise Miner
- Reformulation or DP heuristic
Digital Ad Optimization

- Customer: Large US Media Companies
- Digital Advertising for Media Companies
  - Guidelines for suggested (optimal), and walk-away prices
  - Price elasticity with forecasts
- Pricing forecasting, optimization, and reporting
- SAS tools:
  - SAS Forecast Server
  - SAS/OR
  - SAS Business Intelligence
- Price guidance infrastructure built with portal interface
Shipping Rate Optimization

- Customer: Large US Retailer
- Shipping Rate Optimization
  - Maximize sales revenue + shipping revenue (or profit)
  - Pricing sensitivity analysis with optimization
  - Logistic regression + NLP model

- NLP
- SAS tools:
  - SAS/STAT: PROC LOGISTIC
  - SAS/OR: OPTMODEL, NLP solver
- Customer sensitivity models and optimization model