# More about OPL and Integer Programming

DS 775

# Topics

- Optimization Programming Language (OPL)
  - using external data
  - our example: Excel -> R -> OPL
- Integer Programming
  - examples
  - how is different than linear programming?
  - much greater computational complexity

# Using External Data in OPL

- for larger models you're not going to want to type a .dat file
- options
  - avoid OPL and construct model in R or python, then solve through API or native solver
  - use tools in IBM Optimization Studio to import data (SQL, Excel, . . . )
  - use rich data tools in R or python to read data and output .dat file (munge!)

# Munging Example

- use R package XLconnect to read data from Excel spreadsheet into dataframes
- use R to write text .dat file
- run in IBM optimization studio
- example write transp4.dat file we saw last week
  - see transpsheet.xlsx and transpWrite.R that from this week's download folder

# Integer Programming

- Decision variables constrained to integer values
- Can produce 5 or 6 cars, but not 5.72 cars
- Often have binary (boolean) variables, 0 for no, 1 for yes

# Prototype Example

| Investment | Cost | Future Value |
|---|---|---|
| 1 | 6 | 9 |
| 2 | 3 | 5 |
| 3 | 5 | 6 |
| 4 | 2 | 4 |

- decision variables: $x_j = 0$ (no) or 1 (yes) to buy investment $j$
- constraints
  - total cost $\leq 10$
  - investments 3 and 4 aren't allowed together: $x_3 + x_4 \leq 1$
  - only invest in 3 after investing in 1: $x_3 \leq x_1$
  - only invest in 4 after investing in 2: $x_4 \leq x_2$

# Prototype Formulation

Maximize $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$ subject to:

$$\begin{array}{rcrcrcrcr}
6x_1 & + & 3x_2 & + & 5x_3 & + & 2x_4 & \leq & 10 \\
 & & & & x_3 & + & x_4 & \leq & 1 \\
-x_1 & & & + & x_3 & & & \leq & 0 \\
 & -x_2 & & & & + & x_4 & \leq & 0
\end{array}$$

and each $x_j$ has to be 0 or 1.

# Site Selection

- each $x_i$ is binary to build a production facility at location $i$
- minimize total cost
- subject to meeting production need

# Either-Or Constaints

- suppose we want one or the other or perhaps both of these:

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 \\ x_1 + 4x_2 &\leq 16 \end{aligned}$$

- introduce an extra binary variable $y$ and let $M$ be a very large number, then use

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 + My \\ x_1 + 4x_2 &\leq 16 + M(1 - y) \end{aligned}$$

# Fixed-Charge Problems

- build $x_j$ widgets at cost $c_j$
- addition fixed cost of $k_j$ if $x_j > 0$
- binary variable $y_j$ is 1 if we decide to make widgets and 0 otherwise
- to make sure $x_j$ is 0 if $y_j$ is zero we add a constraint $x_j \leq My_j$ (use large $M$)
- total cost of widgets is $\sum c_j x_j + k_j y_j$

# NP-Hard Problems

- many integer programs fall into this class of problems
- the computational complexity can increase exponentially with the number of variables
- by contrast the Simplex Method scales linearly
- Interior Point method scales cubically
- computational complexity can grow exponentially

# Numerical Methods for Integer Programming

- cutting plane methods
  - start with solution allowing real variables (linear relaxation)
  - add linear constraints to drive the solution to feasible integer solution
  - also called Branch-and-Cut
- branch and bound methods
  - split the search space recursively and remove parts that aren't competitive

# Employee Scheduling Problem

- schedule employees at restaurant to meet labor demand and minimize payroll
- decision variables
    - shift start time (integer)
    - length of shift (continuous)
- *Mixed Integer Program*

# Applications of (mixed) integer programming

- production planning (maximize production)
- scheduling (find feasible solution, minimize cost)
- telecomm network planning (minimizing cost)
- cellular networks (allocating frequencies)
- traveling salesman (minimize cost)

# Integer Programming in OPL

- just declare the relevant variables to be integer type and specify the range
- for binary variables the last two statements are equivalent

```
dvar int x in 0..maxint;
dvar int y in 0..5;
dvar z in 0..1;
dvar boolean z;
```

# Integer Programming in Solver

- Dave Reineke is helping with this slide so consider this a placeholder

# Large Integer Programs

- may require special numerical techniques
- some involve relaxation
- solve sequence of problems with continuous variables
- special methods for all binary variables
- others