

Modeling Languages for Optimization

DS 775

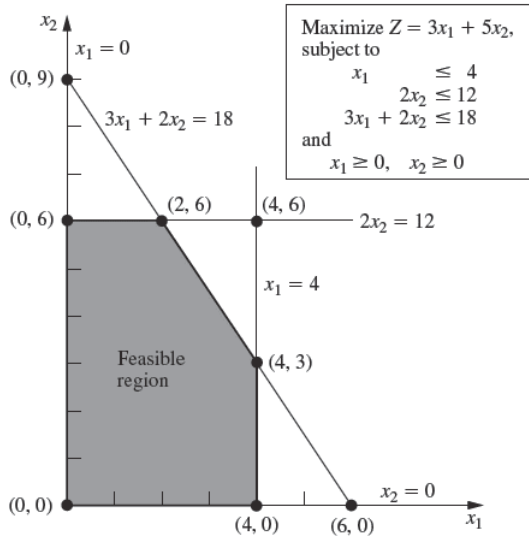
Topics

- Optimization Programming Language (OPL)
 - overview of IDE
 - programming basics
- Numerical Methods for Linear Programming

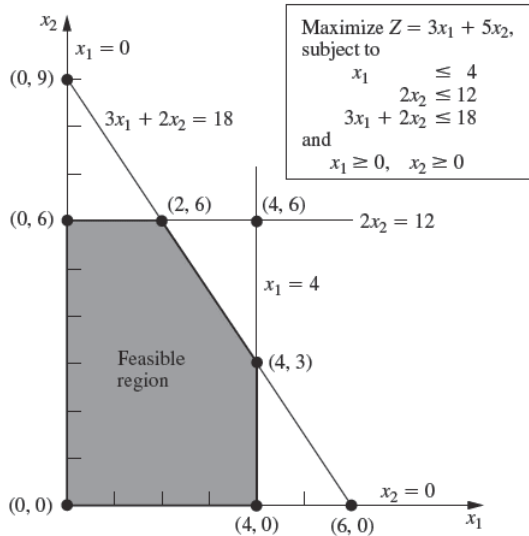
Two Numerical Methods

- Simplex Method (Dantzig 1947)
 - widely considered one of top 10 numerical algorithms from the 20th century ([link below](#))
- Interior Point Method (Karmakar 1984)

Wyndor Problem - Simplex Method



Wyndor Problem - Interior Point Method



Simplex - Pros and Cons

- Pros

- n variables, usually works with $O(n)$ operations
- exploits geometry to visit vertices
- good for small problems

- Cons

- worst case $O(2^n)$ operations
- gets expensive for large problems

Interior Point - Pros and Cons

- Pros

- modern methods require $O(n^3L)$ operations
- better for large, sparse problems

- Cons

- harder to understand
- for small problems Simplex is usually faster

Optimization Programming Language (OPL)

- easier model specification
- separate model from data
- relies on IBM CPLEX solvers for numerics
- interfaces to R, Python, C, and others
- similar to AMPL, LINGO, and others
- also can use with constraint programming

A simple OPL model

```
dvar float+ Doors; /* batches of doors */
dvar float+ Windows; /* batches of windows */

maximize /* profit per batch in thousands */
    3 * Doors + 5 * Windows;
subject to {
    ctPlant1Hours:
        Doors <= 4;
    ctPlant2Hours:
        2 * Windows <= 12;
    ctPlant3Hours:
        3 * Doors + 2 * Windows <= 18;
```

Introduction to Optimization Studio

The screenshot displays the IBM ILOG CPLEX Optimization Studio interface. The main window shows the source code for a linear programming model named `Wyndor1.mod`. The model is defined as follows:

```
1  /* *****  
2  * OPL 12.6.3.0 Model  
3  * Author: jbaggett  
4  * Creation Date: Jun 22, 2016 at 4:13:02 PM  
5  * ***** */  
6  
7  dvar float+ Doors; /* batches of doors */  
8  dvar float+ Windows; /* batches of windows */  
9  
10 maximize /* profit per batch in thousands */  
11     3 * Doors + 5 * Windows;  
12 subject to {  
13     ctPlant1Hours:  
14         Doors <= 4;  
15     ctPlant2Hours:  
16         2 * Windows <= 12;  
17     ctPlant3Hours:  
18         3 * Doors + 2 * Windows <= 18;  
19 }
```

The interface includes several panels:

- OPL Projects:** A sidebar on the left showing the project structure.
- Outline:** A panel on the right showing the model's structure, including decision variables (Doors, Windows), objective (simple), and constraints (ctPlant1Hours, ctPlant2Hours, ctPlant3Hours).
- Properties:** A panel on the right showing the properties of the selected element.
- Problems:** A panel at the bottom showing the list of problems (currently empty).

The status bar at the bottom indicates the current mode is `Writable`, the cursor is in `Insert` mode, and the time is `00:00:00`.

Separating the model and data

Indexing using names

Infeasible constraints

Model and Data - Separate Files

A Transportation Problem

- Shipment of products pairs of cities
 - transport has per product cost
 - transport cannot exceed a given limit
 - transport amount subject to supply and demand constraints
 - Assumes total supply = total demand
 - Minimize total cost subject
- Usually constraint equations are *sparse*
 - each equation involves only a few of the many decision variables

A Transportation Problem - a picture

Install Example into Workspace

First Model

Sparse Model with Tuples