Solving Linear Programs

- Simplex Method - developed by Dantzig in 1940's
  - Standard method
  - Exponential in number of variables
  - Algorithm Þ guaranteed to give optimal solution if one exists (eventually)
  - Searches extreme points of feasible region
- Khachian's Algorithm or Ellipsoid Method - late 70's
  - Also solves LP's
  - Polynomial in number of variables
    - Doesn't work well in practice
    - Showed faster algorithm was possible

Solving Linear Programs

- Karmarkar's Algorithm (middle 80's)
  - Another LP solver
  - Works very fast on large problems
  - Polynomial algorithm
  - Limited ability to do sensitivity analysis
- Specialty Codes - take advantage of special structure
  - Network simplex
  - Transportation method

***Bottom Line**: Simplex still most prominent algorithm*

Optimization Software

- **Big Hitters**
  - **(1) Optimization Subroutine Library (OSL) - IBM**
    - **– Callable subroutines**
      - **- Can be built into DSS**
      - **- Can be called from FORTRAN, C, PL/1**
    - **– Several Solvers**
      - **- Simplex & Interior Point for LP**
      - **- Quadratic & Network**
      - **- MIP (some preprocessing)**
    - **– Runs on any IBM platform + almost all WS**
    - **– Data entry**
      - **- Matrix Generator**
      - **- MPS file**
      - **- Spread Sheet**
    - **– ~$5000 for1 WS license**

Optimization Software

- **(2) CPLEX - very much like OSL**
  - **– Faster LP solver**
  - **– Also runs on Mac**
- **(3) Others**
  - **– XA**
  - **– XPRESS-MP**
  - **– LINDO**

- **Modeling Languages**
  - **(1) GAMS (2) AMPL (3) MPL**
    - **– Easy way to enter problem**
    - **– Front end to solver (OSL,CPLEX, etc.)**
    - **– Comes with own solver as well**

Spread Sheet Optimization

- Most Spread Sheets have one built in
  - EXCEL has "Solver"
  - Lotus 1-2-3 has one also
- Add-ins
  - (1) What's Best (from LINDO) (2) Premium Solver
    - Increases size of problem that can be solved
    - Add other types of solvers (IP, NLP, etc.)
- Why Spread Sheet optimization?
  - Ease of understanding for user and decision maker
  - Ease of data entry
  - Instant graphical output

Model Building in Spreadsheets

Spreadsheets have strengths and weaknesses

- **Cons**
  - Documentability - very difficult to document a spreadsheet
    - Formulas are hidden
    - Mitigated by cell notes and auditing tools
    - Hard to use other people's spreadsheets
  - Scalability - hard to change the dimensionality of problems
    - What do you do when the quantity of data changes?
    - Best to use range formulas (e.g., SUM, SUMPRODUCT)
  - Hyperscalability - addition of a new dimension to a model
    - Spreadsheets are two dimensional
    - Need to copy model multiple times into other sheets to overcome this problem - awkward
- **Pros - EVERYONE HAS ONE!**

Five Stages of Model Development

1. Decide what you want the model to do

2. Decide how to build the model

3. Build the model

4. Debug the model

5. Trash stages 1 - 4 and start again, now that you know what you really wanted to do in the first place

Steps in Using Excel Solver

(1) Logically organize data (label, etc.)
- **Coefficients for objective function**
- **Coefficients for constraints**
- **RHS of the constraints**

(2) Reserve cells for the decision variables
- **Called *Changing Cells***

(3) Create formula in a cell for the objective function
- **Called *Target Cell***

(4) Create a formula for the LHS of each constraint

(5) Open Solver Dialog box (Tools menu)

(6) Enter the appropriate info and run Solver

How Solver Views the Model

Target Cell **contains the equation for the objective function**

Solver need to know if this is to be maximized or minimized

Changing Cells **are the decision variable cells**

Constraint Cells **are the left-hand side formulas and any upper/lower bounds**

RHS for constraints must be cells containing numbers (or simple numbers)---not formulae

How to Use Solver

First, model must be specified to Solver

Target cell, changing cells, constraint cells

For LPs, use <u>Assume Linear Model</u> option

Ignore Estimates, Derivatives, and Search options for now (nonlinear problems)

Ignore Tolerance option for now (integer problems)

Choose <u>Solve</u> and hope for the best!

Answer and Sensitivity reports contain solution

Formulation

Solving an LP is easy---formulating one is sometimes tricky!

Some generic constraint types

- Resource constraints
- Product-Mix constraints
- Blending constraints
  - Unlimited ingredients
  - Limited ingredients
- Inventory constraints

Resource Constraints

$$\sum_{j=1}^{N} a_{ij} x_j \le b_i$$

Example: product 1 requires 15 minutes per part, 2 requires 12 minutes and product 3 requires 20 minutes. There are 8 hours available. The constraint is:

$$15x_1 + 12x_2 + 20x_3 \le 480$$

Product-Mix Constraints

Can only sell so much: $x_j \leq U_j$

Must sell at least so much: $x_j \geq L_j$

Substitutes have upper limit on their demand:

$$x_j + x_k + x_l \leq U_i$$

Blending Requirements (Type I)

There is some <u>proportion</u> of a characteristic desired in the mix and <u>unrestricted (or varying) quantities</u> of each ingredient. Each ingredient has some proportion of the characteristic.

Good characteristic: $\geq$ constraint

Bad characteristic: $\leq$ constraint

Usually a "minimize cost of mixture" problem

Think of making a <u>unit amount</u> (or recipe) of the good: sum of proportions = 1

Blending Type I, cont.

Example: dog food requires no more than 20% filler. Ingredient 1 is 12% filler, ingredient 2 is 22% filler and ingredient 3 is 26% filler. The constraints are:

$$.12x_1 + .22x_2 + .26x_3 \leq .20$$
$$x_1 + x_2 + x_3 = 1$$

Blending Requirements (Type II)

There is some <u>given amount of each ingredient</u>
Objective is typically to maximize profit
Proportion constraint now looks like:

$$\frac{\sum_{j=1}^{N} a_{ij} x_j}{\sum_{j=1}^{N} x_j} \leq b_i$$

which can be rewritten as:

$$\sum_{j=1}^{N} (a_{ij} - b_i) x_j \leq 0$$

Blending Type II, cont.

Example: In the dog food example, suppose we have 400 pounds of ingredient 1, 250 pounds of ingredient 2 and 600 pounds of ingredient 3. Then the constraints are:

$$-.08x_1 + .02x_2 + .06x_3 \le 0$$
$$x_1 \le 400$$
$$x_2 \le 250$$
$$x_3 \le 600$$

Inventory Constraints

Time dependent problem---subscript *t* indicates period (day, week, etc.)

Let d_t be the demand to be met (parameter) and p_t equal the production (decision)

The amount of inventory at the end of a period is a function of production, demand and previous inventory:

Normally, production and inventory have a cost associated with them in a minimize cost objective

$$I_t = I_{t-1} + p_t - d_t$$