Integer-Linear and Non-Linear Optimization Models

GW Business

Recall: Three Major Types of Optimization Models

LP: Linear Models — *… computationally **easy** to solve*

The Objective Function and all Constraints are linear functions, and decision variables are continuous (not required to be integer)
→ Select Solving Method **Simplex LP**

ILP: Linear Models with Integer Variables — *… computationally **less easy** to solve*

The Objective Function and all Constraints are linear functions, and some decision variables are required to take integer values
→ Select Solving Method **Simplex LP**, declare **Integer** or **Binary** constraints on those decision variables that require it

NLP: Non-Linear Models — *… computationally **difficult** to solve*

Something is not linear: the Objective Function, or some Constraint(s), or both
→ Select Solving Method **GRG Nonlinear**

Models with Integer or Binary Decision Variables

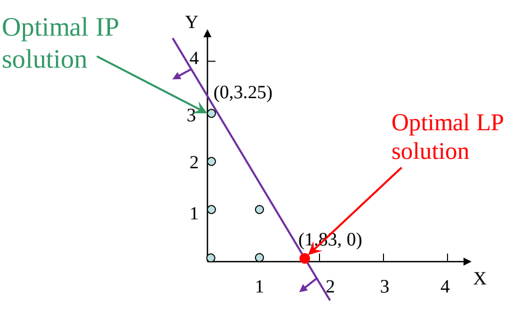
- Used when some <u>decision variables</u> are required to take integer values
- Why care?
  - Can't build 1.37 aircraft carriers
  - Need a Yes/No decision: X = 1 or 0 (binary variable)
  - Rounding may not give the best, or even a feasible, solution
- Achieved by putting =*integer* or =*binary* constraints on the decision variables
  Note: integer and binary constraints can be declared on <u>decision variables only,</u> not on formula cells
- Harder, slower to solve than continuous (without integer) Linear Optimization
- Solving without the integer constraints provides an <u>upper bound</u> on objective function value of optimal integer solution
  ⇒ can estimate by how much the current integer solution might be improved
  Solver stops when it finds an integer solution within a specified Tolerance (e.g. 5%)
  Go to: *Data* > *Solver* > *Options* > *All Methods* tab > *Integer Optimality (%)*
- Sensitivity Report NOT available for models with integer or binary constraints

Issues in Solving Integer Optimization

$$\textit{Max}:\ 21\,X + 11\,Y$$

Subject to:

$$7\,X + 4\,Y \le 13$$
$$X,\ Y \ge 0$$



Feasible Set: Triangle with corners (0,0), (1.83, 0), (0, 3.25)

Optimal Linear-Programming solution: X = 1.83, Y = 0

Rounded to X = 2, Y = 0 is **infeasible**

Rounded to X = 1, Y = 0 is **not optimal**

Optimal Integer-Programming solution: X = 0, Y = 3

Examples of Use of Binary Decision Variables

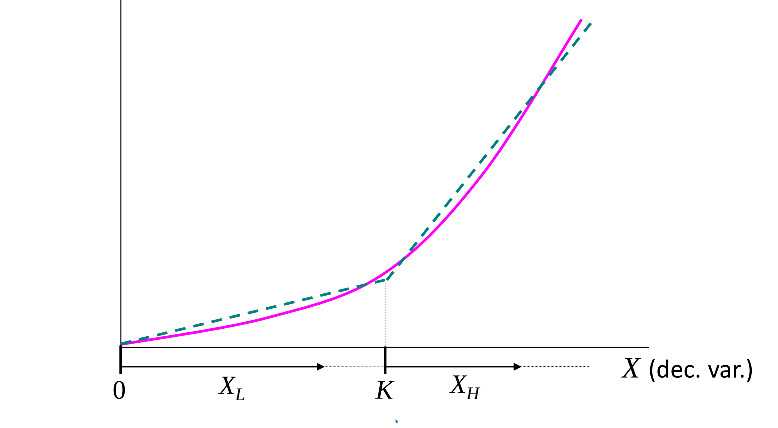
- ✓ Capital budgeting
  Invest in a project or not (Yes/No decision)
  Select at most n in a list (Sum of binary variables ≤ n)
- ✓ Facility location
  Build a plant at a location or not
- ✓ Fixed cost/Set-up cost
  Fixed cost incurred if any quantity is produced, not otherwise
- ✓ Threshold levels
  If a car model is produced at a plant, then at least 2,000 units must be produced: $X = 0$ or $X \geq 2{,}000$

5

Use of binary variables to emulate IF() and MAX() with a Linear model formulation

❑ **Dealing with increasing returns**

Contribution of X
to the objective



6

- Obtain a piecewise linear approximation by defining new incremental variables:
  - X_L = part of X with Low returns, $X_L \leq K$
  - X_H = part of X with High returns, $X_H \geq 0$
- Modify the objective function and constraints accordingly
- Now: X_H can take values > 0 <u>only if</u> X_L becomes = K.
  This is achieved by adding a **new decision variable**, X_0,
  and **new constraints** as follows:

$$\begin{cases} X_0 = \textit{binary} \text{ (0 or 1)} \\ X_L \geq K X_0 \\ X_H \leq M X_0 \quad \text{with } M = \text{a large enough arbitrary value you choose} \end{cases}$$

  This set of "interlocking" constraints will act just like an '**IF**' statement, except it's an Integer-Linear optimization formulation, easy for Solver to handle!
- A worked-out example (from Exercise Set)

7

Use of binary variables to emulate IF() and MAX()

Dealing with fixed costs

Objective Function: Min *Cost* = $c_0 + c_1 X_1 + c_2 X_2 + \ldots + c_n X_n$

- If c_0 is incurred regardless of decisions, then just ignore it (sunk cost)
- If c_0 is a set-up cost, for example incurred only if X_1 is used, then:
  - Define a new decision variable X_0:
    add constraint: X_0 = *binary*
  - Modify the objective function:
    Min Cost = $c_0 \mathbf{X_0} + c_1 X_1 + c_2 X_2 + \ldots + c_n X_n$
  - Add the constraint: $X_1 \leq M X_0$
    where M is an arbitrary large constant you choose,
    larger than maximum conceivable value of X_1

8

Intermediate Summary

Obtaining a linear formulation of an optimization model (with or without integer variables) is key to solving large scale problems efficiently and reliably

Many non-linear, non-smooth models can be linearized by using appropriate modeling techniques

Linearizing sometimes increases the size of the model (but that's less of an issue!)

Yet not all applications can be formulated with a linear model; hence non-linear optimization methods are needed.

9

9

Non-Linear Optimization Models

Non-linear optimization model

Whenever the objective function or some constraint(s) involve a non-linear operation of the decision variables, we have a so-called Non-Linear Programming (NLP) model

Where does non-linearity come from?

Non-constant returns to scale –*Economies or diseconomies of scale/scope, e.g., marginal costs/benefits that change as volume changes*

Interactions effects – *e.g., demand depends on price, and revenue is the product of price times demand, leading to nonlinearity*

Threshold effects and contingencies – *e.g., the value of an option in the future is the maximum of i) the value if exercised, or ii) the value if not exercised*

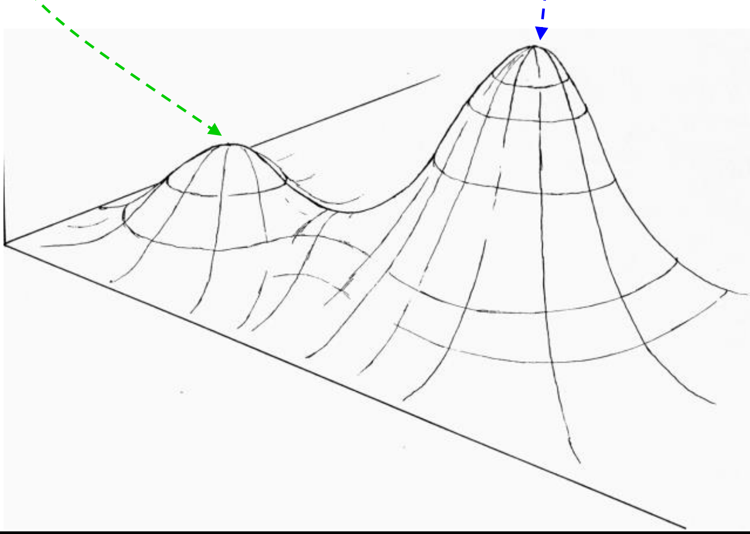
11

11

Local vs. Global Optimum: Graphical illustration

Local optimum: a solution that's better than all those in its neighborhood, but may not be the overall best

Global optimum: the absolute best solution (where you want to be)



12

12

Solving Non-Linear Optimization

Solution algorithm

- the so-called Gradient Method = "mountain climbing" (*in the fog!*)
- Seek incremental improvements of Objective Function
- Much less powerful than Linear case, restricted to small size problems

Difficulty in solving

- solution search process is computationally laborious, not 100% reliable
- can get "trapped" by <u>local</u> optima

- Need to experiment solution search with different starting values
- Use guess values of decision variables, if available, as starting values in the solution search.

13

Sensitivity Report for Non-Linear Optimization

The Sensitivity Report includes:

- for each Decision Variable:

  "Reduced Gradient" same as *Reduced Cost*

- for each Constraint:

  "Lagrange Multiplier" same as *Shadow Price*

Lagrange Multipliers and Reduced Gradients have the same economic interpretation as their linear counterparts, but unlike them, they are not constant over any range

$\Rightarrow$ Interpretation valid for small changes only

14

Risk Minimization, Diversification, and Portfolio Selection

Efficient Portfolio Selection

Goal: select an investment portfolio that achieves a **best risk-return** balance, that is, lowest risk for a given return, or highest return for a given risk.

Could be a portfolio of stocks, assets, projects, start-up companies,…

Portfolio optimization model can be built from either:

- Historical returns for each asset i, $i = 1, \dots, n$
- Forecasts of returns (e.g., from MC Simulation)
- Probabilistic scenarios of future returns for each asset i, $i = 1, \dots, n$
- Or just expected returns and variances/covariances

  r_i = expected return from Asset i

  σ_i^2 = variance of returns from Asset i

  σ_{ij} = covariance of returns from Asset i and Asset j

If only the Covariance matrix is known, we can only compute ***Variance*** *as a risk measure*

15

15

Mean-Variance Portfolio selection (Markowitz 1952)

❑ **Decision Variables**

w_i = proportion of portfolio invested in Stock *i*, so-called portfolio **weights**

❑ **Objective** *Min Risk* or *Max Return* ?

Minimize Portfolio Risk, for a specified Expected Return

Q. What is the portfolio variance as a function of weights?

A. For a two-stock portfolio: $\sigma^2(\text{Portfolio}) = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\sigma_{12}$

General matrix formula: $\sigma^2(\text{Portfolio}) = [w_i][\sigma_{ij}][w_i]^T$

Note: Variance is **quadratic**, hence non-linear, in the decision variables!

→ Need to use non-linear optimization solving method

❑ **Constraints**

- Achieve a target expected return, R: $\sum_i r_i w_i \geq R$
- Invest total budget: $\sum_i w_i = 1$
- No short-selling: $w_i \geq 0$ (non-negativity)
- other?...

16

16

Recall: Combination of Random Variables

Weighted Sum of two random variables

$$Z = a + b\,X + c\,Y$$

$$\mathrm{E}[Z] = a + b\mathrm{E}[X] + c\mathrm{E}[Y]$$

$$\mathrm{Var}(Z) = b^2\mathrm{Var}(X) + c^2\mathrm{Var}(Y) + 2bc\,\mathrm{Covar}(X, Y)$$

where $\mathrm{Covar}(X,Y) = \mathrm{E}[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])]$ is the covariance of X and Y, which can be positive, negative, or zero.

Application:

b = proportion invested in Stock X

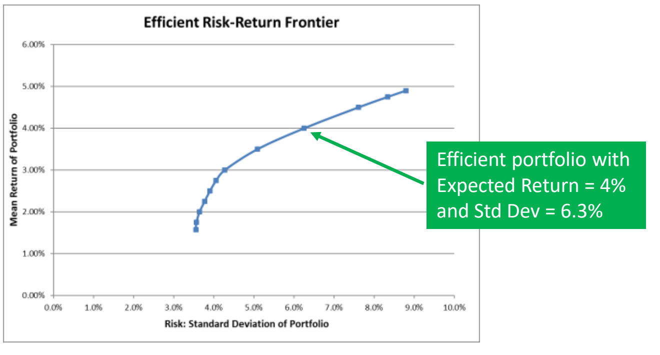
c = proportion invested in Stock Y

17

17

Portfolio selection: Efficient Frontier

- See Practice Exercise example for complete model implementation in Excel
- By setting the portfolio return constraint at different values and re-solving, we compute optimal portfolios with different Risk-Return profiles
- We plot the Mean vs. the Standard Deviation (or Variance) of the optimal portfolios so obtained: this is the so-called Risk-Return Efficient Frontier.



18

18

Summary on Optimization Models

Optimization is a general decision modeling tool with innumerable business applications:

- Design and manage **Business Operations**
- Make **Resource Allocation** decisions
- Build **Efficient Portfolios** of products, projects, or investments

Optimization modeling requires specifying:

- The set of **Decisions Variables**
- The **Objective Function**
- The set of **Constraints**

Linearity allows to solve very large problems

19

19