Workshop in Mathematical Programming

Model building in Mathematical Programming

Oct. 10 – Nov. 14, 2006 Akiko Yoshise

Materials are available at

http://infoshako.sk.tsukuba.ac.jp/~yoshise/Course/MC/

Schedule :

I. Oct. 10

- What is Mathematical Programming
- How to get XPRESS-MP
- Case study I

II. Oct. 17

- Some Special Types of Mathematical Programming
- Case study I I
- **Assignment #1**
  - Due date Oct. 30

Schedule：

III. Oct. 24:

- Building Integer Programming Model
- Case study III
- **Assignment #2**
  - Due date Nov. 20

IV. Nov. 1:

- Solving Linear Programming Model
- Solving Integer Programming Model

V. Nov. 8:

- Discussions

VI. Nov. 15:

- **Presentation of Assignment#3**

Evaluation:

- Assignment #1 allotted 30
- Assignment #2 allotted 70
- Evaluation

$$\sum_{i=1}^{2} \text{Score of } \# i \times \left(\frac{\#\text{ of Students} - \#\text{ of Group members} + 1}{\#\text{ of Students}} \right)$$

What is Mathematical Programming

- **Optimization**
  - An **objective function** which we want to minimize or maximize
  - A set of **unknowns** or **variables** which affect the value of the objective function
  - A set of **constraints** that allow the unknowns to take on certain values but exclude others
- **Find values of the variables that minimize or maximize the objective function while satisfying the constraints**

Optimization Problem

- General form:

$$
\begin{aligned}
&\text{Minimize } f(x) \\
&\text{subject to } g_i(x) \le 0 \ \ (i = 1,2,\ldots,m) \\
&\qquad\qquad\quad x \in X
\end{aligned}
$$

What is Mathematical Programming

Step1: Formulate a mathematical model

Determine the objective function, the variables, the constraints

Step2: Solve the problem to find a solution

Choose a suitable algorithm or develop an algorithm

Step3: important Interpret the solution

Is the solution satisfactory? If not, **Go To Step1**

Formulating a mathematical model
Objective function

- What are you planning to do?
- Can you describe it using the word **Minimize** or **Maximize**

- No objective function
  - ⇒ Finding a feasible solution
- Multi objective functions
  - ⇒ Minimizing Weighted Sums of Functions
  - important — The magnitude of the functions

Formulating a mathematical model Variables and Constraints

- Find unknowns which affect the value of the objective function

- Constraints are given by equalities or inequalities of the which are
  - greater-than-or-equal-to (>)
  - less-than or-equal-to (<)
  - or equal-to (=)

  a constant term (the right hand side).

Formulating a mathematical model In Xpress-MP (IVE)

- Variables are declared by **mpvar** and expected to be **nonnegative** important
  - **is_binary: 0-1 Integer**
  - **is_integer:** (nonnegative) **Integer**
  - Others: (nonnegative) **Real**

Formulating a mathematical model In Xpress-MP (IVE)

- Basically, Objective function and Constraint functions are expected to be given by

Linear Expression:

important

A term (or sum of terms) of the form

（constant）×（variable）

Linear Expression

$$\text{Constants}: \alpha_i \,(i = 1,2,\ldots,n), \beta_j \,(j = 1,2,\ldots,m)$$

$$\text{Variables}: x_i \,(i = 1,2,\ldots,n), y_{ij} \,(i = 1,2,\ldots,n,\ j = 1,2,\ldots,m)$$

$$\text{Linear expression}: \ \sum_{i=1}^{n} \alpha_i \times x_i, \sum_{i=1}^{n} \alpha_i \times (y_{ij} - \beta_j), \cdots$$

$$\text{Nonlinear expression}: \ \sum_{i=1}^{n} x_i \times x_i, \sum_{i=1}^{n} \sum_{j=1}^{n} x_i \times x_j, \sum_{i=1}^{n} x_i^{\alpha_i}, \cdots$$

Why we should use Linear Expression?

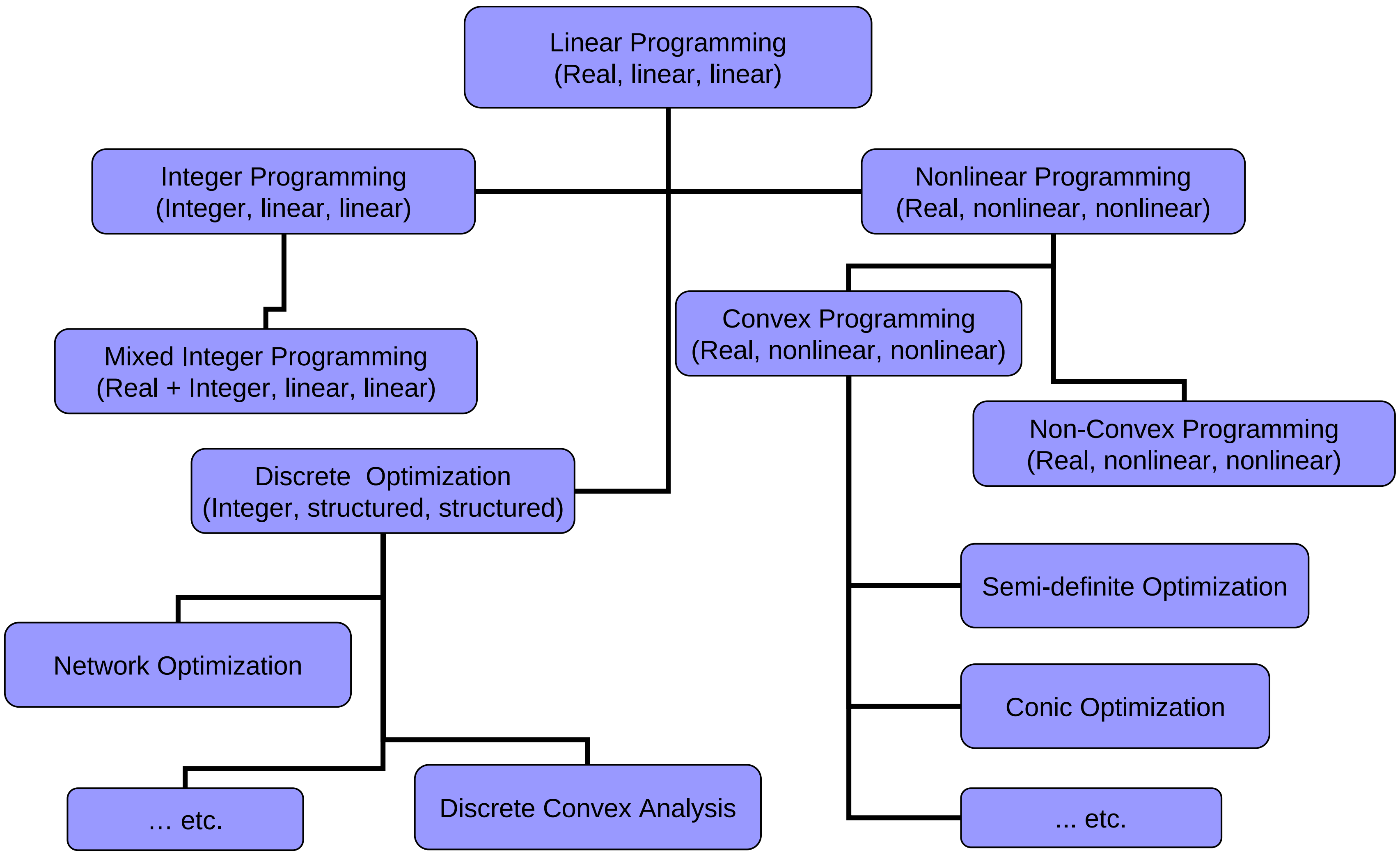
- Based on Linear Programming Problem ⇒ Lecture IV

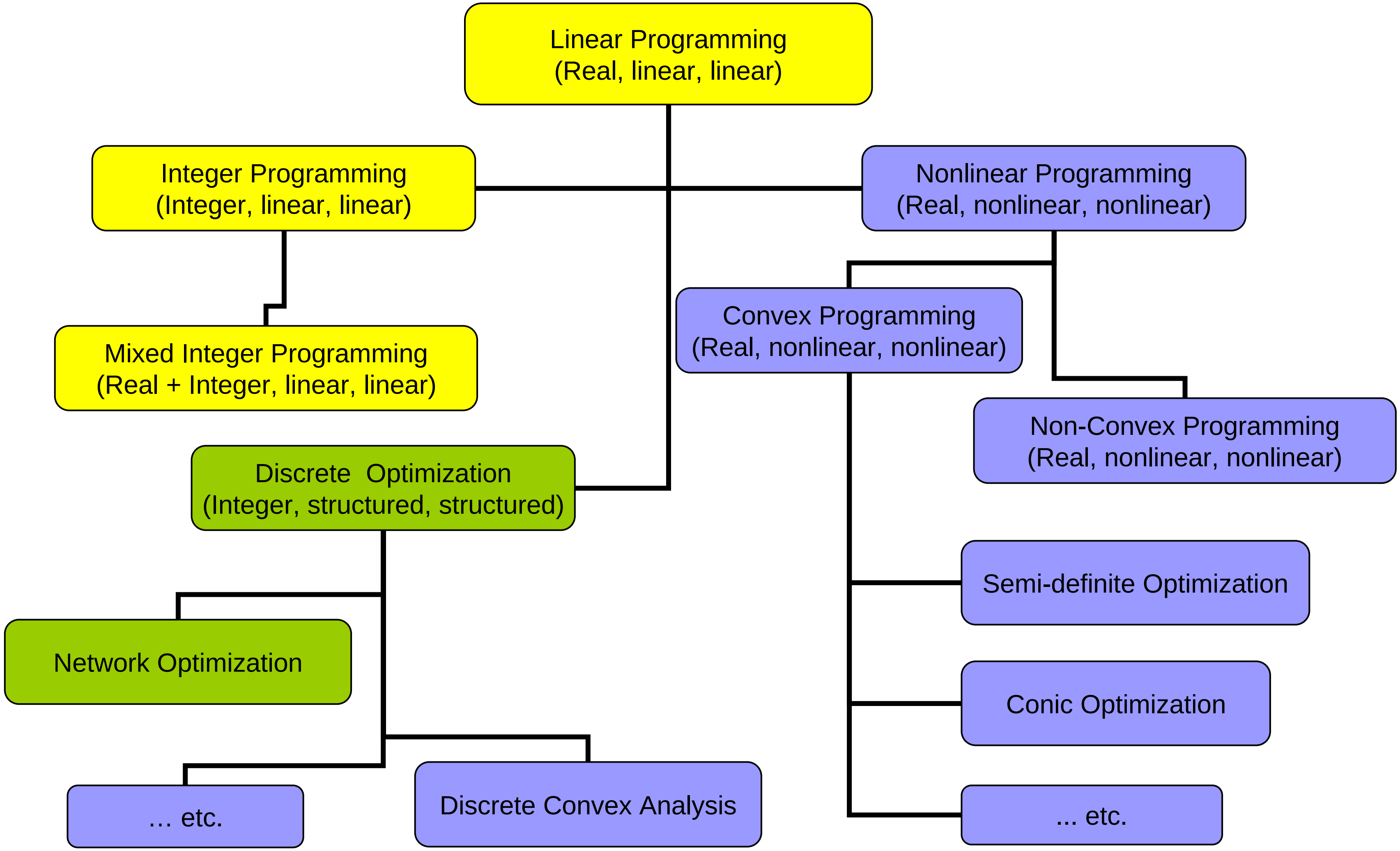
$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} = \sum_{i=1}^{n} c_i x_i$$

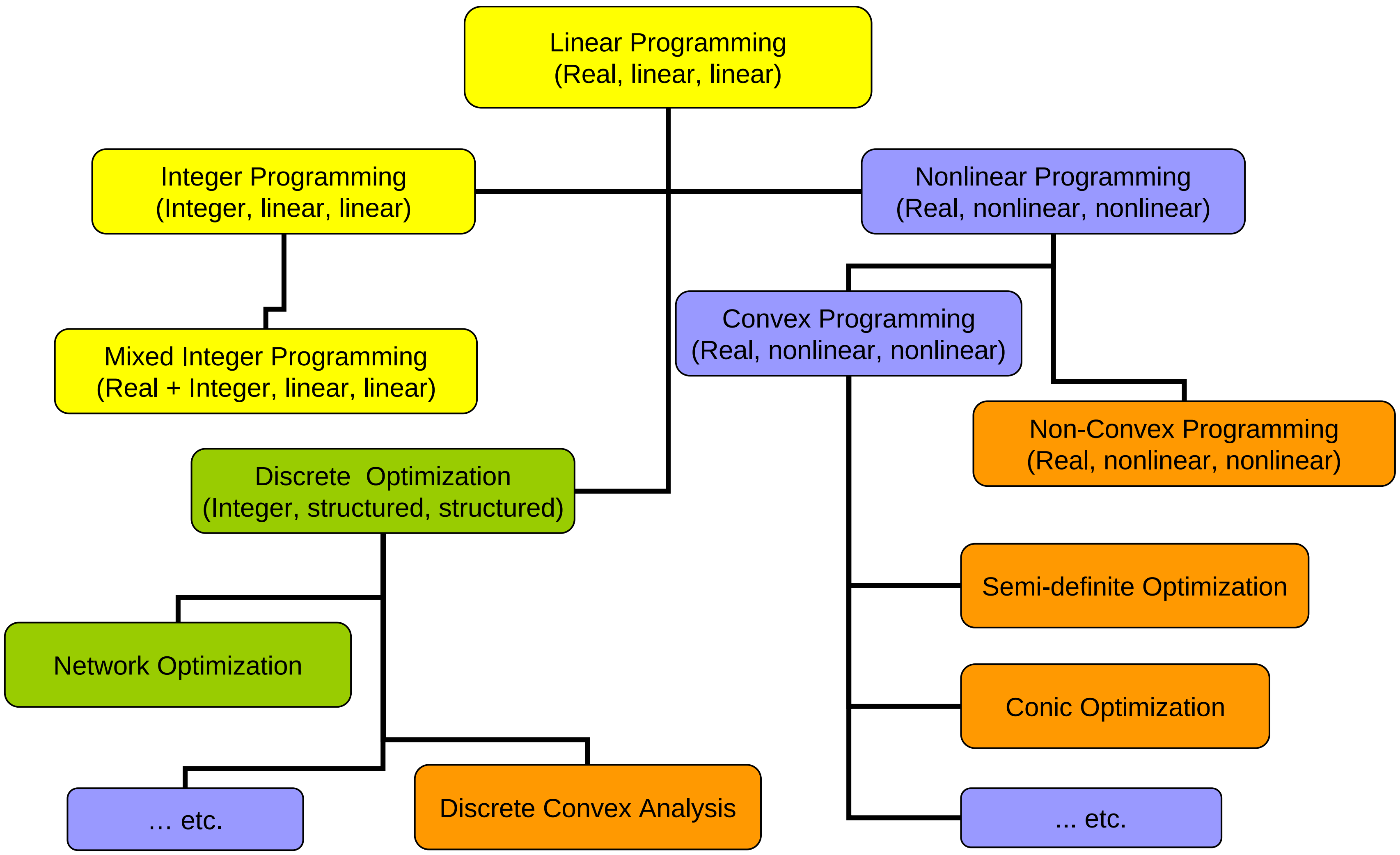
$$\text{Subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \left(\Leftrightarrow \sum_{j=1}^{n} a_{ij} x_j = b_i (i = 1,2,\ldots m)\right)$$

$$\mathbf{x} \geq \mathbf{0} \quad (\Leftrightarrow x_j \geq 0 (j = 1,2,\ldots n))$$

- Many problems can be expressed by **Linear Expression**
  ⇒ Lecture III


Linear Programming
(Real, linear, linear)
Integer Programming
(Integer, linear, linear)
Nonlinear Programming
(Real, nonlinear, nonlinear)
Convex Programming
(Real, nonlinear, nonlinear)
Mixed Integer Programming
(Real + Integer, linear, linear)
Non-Convex Programming
(Real, nonlinear, nonlinear)
Discrete Optimization
(Integer, structured, structured)
Semi-definite Optimization
Network Optimization
Conic Optimization
… etc.
Discrete Convex Analysis
... etc.


Linear Programming
(Real, linear, linear)
Integer Programming
(Integer, linear, linear)
Nonlinear Programming
(Real, nonlinear, nonlinear)
Convex Programming
(Real, nonlinear, nonlinear)
Mixed Integer Programming
(Real + Integer, linear, linear)
Non-Convex Programming
(Real, nonlinear, nonlinear)
Discrete Optimization
(Integer, structured, structured)
Semi-definite Optimization
Network Optimization
Conic Optimization
… etc.
Discrete Convex Analysis
... etc.



Linear Programming
(Real, linear, linear)
Integer Programming
(Integer, linear, linear)
Nonlinear Programming
(Real, nonlinear, nonlinear)
Convex Programming
(Real, nonlinear, nonlinear)
Mixed Integer Programming
(Real + Integer, linear, linear)
Non-Convex Programming
(Real, nonlinear, nonlinear)
Discrete Optimization
(Integer, structured, structured)
Semi-definite Optimization
Network Optimization
Conic Optimization
Discrete Convex Analysis
… etc.
... etc.

How to get XPRESS-MP

- Student version (IVE): http://www.dashoptimization.com/index.html
- Manual:
  - Essentials
  - Optimizer-ref
- Version 13 on **shakosv**
  - Remember some commands:

shakosv:[22]: mp-mosel

>> help

>> exec <file name>.mos

>>exit

The Burglar Problem

Xpress-MP Essentials p.108

- Burglar Bill breaks into a house one night with a sack to carry away items of interest to him. He identifies a number of items which have the following weights and estimated values:
- Bill can only carry items up to a total weight of 102 pounds. Subject to this, his aim is to maximize the estimated value of the items that he takes.

	Weight	Value
Camera	2	15
Necklace	20	100
Vase	20	90
Picture	30	60
TV	40	40
Video	30	15
Chest	60	10
Brick	10	1

- Formulating the Burglar Problem mathematically is relatively simple.
- Suppose that we have a variable, *camera*, which has
  - the value 1 if Bill takes the camera and
  - the value 0 otherwise.
- Suppose also that we have a similar set of variables for the other items.

Problem expression

Problem 1

Maximize: $15*camera + 100*necklace + 90*vase + 60*picture + 40*tv + 15*video + 10*chest + 1*brick$

Subject to: $2*camera + 20*necklace + 20*vase + 30*picture + 40*tv + 30*video + 60*chest + 10*brick \leq 102$

$camera, necklace, vase, picture, tv, video, chest, brick \in \{0, 1\}$

Listing 5.1 Declaring decision variables

```
model burglar
  declarations
    camera, necklace, vase, picture: mpvar
    tv, video, chest, brick: mpvar
  end-declarations

end-model
```

model

The model block defines the problem name and contains all statements to be considered part of the model. Any text appearing after the `end-model` is treated as if it were a comment and ignored.

declarations

In the `declarations` block are defined variables and their type, sets and constants.

Listing 5.2 Adding constraints to the model

```
model burglar
  declarations
    camera, necklace, vase, picture: mpvar
    tv, video, chest, brick: mpvar
  end-declarations

  camera   is_binary
  necklace is_binary
  vase     is_binary
  picture  is_binary
  tv       is_binary
  video    is_binary
  chest    is_binary
  brick    is_binary
```

```
  TotalWeight := 2*camera + 20*necklace + 20*vase +
                 30*picture + 40*tv + 30*video    +
                 60*chest + 10*brick <= 102

  TotalValue := 15*camera + 100*necklace + 90*vase +
                60*picture + 40*tv + 15*video      +
                10*chest + 1*brick

  maximize(TotalValue)

  writeln("Objective value is ", getobjval)

end-model
```

Assignment #1 (Due date Oct.24)

- You are working at an electric power company, and should solve a routing problem for supplying electric power from two generating plants to three demand points.

- The production of electricity at each plant and the amount required at each demand point are given as follows:

Supply and Demand Table:

	Plant S1	Plant S2
Production	100,000kw	150,000kw

	Demand D1	Demand D2	Demand D3
Amount	90,000kw	100,000kw	60,000kw

- There are three relay points, T1,T2 and T3, and the electricity can be supplied though these points.
- The cost required to send the unit of electricity from a point to another point is proportional to the distance between two points.

Distance Table:

blank means no line

	S1	S2	T1	T2	T3	D1	D2	D3
S1	0		50					
S2		0			40			100
T1			0	60	20	30	30	
T2				0	10		40	
T3					0		5	20
D1						0		30
D2							0	5
D3								0

- Formulate a mathematical model for finding a minimum cost routing from each plant to each demand point.
- Solve the problem using Xpress-MP.
- Remind “What is Mathematical Programming”
  - **Describe each step clearly!**

What is Mathematical Programming

Step1: Formulate a mathematical model

Determine the objective function, the variables, the constraints

Step2: Solve the problem to find a solution

Choose a suitable algorithm or develop an algorithm

Step3: important Interpret the solution

Is the solution satisfactory? If not, **Go To Step1**