Lecture 2 - INF-MAT 5360

- Integer programming
- Modeling with integer variables
- Introduction to OPL studio

Integer programming

Linear problems where the variables are integer valued

$$
\begin{aligned}
\max \quad & c^T x \\
\text{s.t.} \quad & Ax \leq b \\
& x \in \mathbb{Z}^n_+
\end{aligned}
$$

If only some of the problems are integer valued, the problem is a mixed integer problem (MIP).

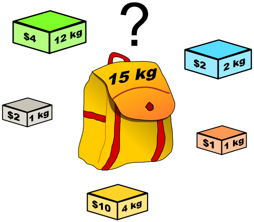
Integer variables

Integer variables often restricted to $0 - 1$ variables

- Modeling yes/no decisions
- Enforcing disjunctions
- Enforcing logical conditions
- Modeling fixed cost
- Modeling piecewise linear functions

Knapsack problem

- We have a knapsack with capacity b
- Can select from N items where item j has size a_j and profit c_j
- What items should be selected to maximize profit?



- Decision variables

$$x_j = \begin{cases} 1, & \text{item } j \text{ goes into the knapsack,} \\ 0, & \text{otherwise} \end{cases}$$

- Model formulation

$$\begin{aligned} \max \quad & \textstyle\sum_{j=1}^{N} c_j x_j \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{N} a_j x_j \leq b_j \\ & x_j \in \{0, 1\}, \ j = 1, \ldots, N \end{aligned}$$

Uncapacitated facility location

- Set of customers $I = \{1, \ldots, n\}$
- Each customer can be served from one of the facilities $J = \{1, \ldots, m\}$
- Each facility has a cost c_j if opened
- Customer i's demand can be serviced from facility j at cost f_{ij}
- Which facilities should be opened in order to meet customer demand at minimum cost?

- Decision variables

$$x_j = \begin{cases} 1, & \text{fac. } j \text{ is opened,} \\ 0, & \text{otherwise} \end{cases} \qquad y_{ij} = \begin{cases} 1, & \text{cust. } i \text{ served by fac. } j, \\ 0, & \text{otherwise} \end{cases}$$

- Model formulation

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j x_j + \sum_{i \in I} \sum_{j \in J} f_{ij} y_{ij} \\ \text{s.t.} \quad & \sum_{j \in J} y_{ij} = 1, \quad \text{for all } i \in I \\ & y_{ij} \leq x_i, \quad \text{for all } i \in I, j \in J \\ & x_j \in \{0, 1\}, \; j \in j \\ & y_{ij} \in \{0, 1\}, \; i \in I, j \in J \end{aligned}$$

Traveling Salesman Problem

Modeling tricks

- Selecting from a set

$$\sum_{j \in T} x_j \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} 1$$

- Disjunctive constraints
  At least one of $a^T x \geq b$ and $c^T x \geq d$ holds (assume non-negative coeff.)

$$\begin{array}{l} a^T x \geq yb \\ c^T x \geq (1-y)d \\ y \in \{0,1\} \end{array}$$

- Logical conditions

At most one of X, Y, Z	$x + y + z \leq 1$
Exactly two of X, Y, Z	$x + y + z = 2$
If X then Y	$x \leq y$
If X then not Y	$x + y \leq 1$
If not X then Y	$x + y \geq 1$
If X then Y or Z	$x \leq y + z$

- Indicator variables $\delta \in \{0, 1\}$

  Indicate whether or not an inequality holds

  - $\delta = 1 \;\Rightarrow\; a^T x \leq b$
    Can be represented by

    $$a^T x + M\delta \leq M + b$$

    where M is an upper bound on $a^T x - b$
  - $a^T x \leq b \;\Rightarrow\; \delta = 1$
    Can be represented by

    $$a^T x - (m - \epsilon)\delta \geq b + \epsilon$$

    where m is a lower bound on $a^T x - b$ and ϵ a small tolerance.

Modeling languages

- Computer language with syntax close to the standard mathematical description
- Clean separation between model and instance data
- Abstracts away the implementation details of the underlying solver
- Traditionally used for linear and integer programming
- Extended to quadratic programming, nonlinear programming and constraint programming
- Examples: AMPL, OPL, Mosel