Modelling in
Mixed Integer Linear Programming

sophie.demassey@mines-paristech.fr

October 16, 2023

1 Model examples

1.1 Integer Knapsack Problem

Input: n items, value c_j and weight $w_j \geq 0$ for each item j, a capacity $K \geq 0$.
Output: a maximum value subset of items whose total weight does not exceed capacity K.

$$\begin{aligned} \max \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} \sum_{j=1}^{n} w_j x_j \leq K & \\ x_j \in \{0,1\} & \qquad j = 1..n \end{aligned}$$

with $x_j = 1$ iff item j is selected

1.2 Uncapacitated Facility Location Problem

Input: n facility locations, m customers, cost c_j to open facility j, cost d_{ij} to serve customer i from facility on location j.
Output: a minimum (opening and service) cost assignment of the customers to the open facilities.

$$\begin{aligned} \min \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n}\sum_{i=1}^{m} d_{ij} y_{ij} & \\ \text{s.t.} \sum_{j=1}^{n} y_{ij} = 1 & \qquad i = 1..m \\ y_{ij} \leq x_j & \qquad j = 1..n,\ i = 1..m \\ x_j \in \{0,1\} & \qquad j = 1..n \\ y_{ij} \in \{0,1\} & \qquad j = 1..n,\ i = 1..m \end{aligned}$$

where $x_j = 1$ iff a facility is open at location j and $y_{ij} = 1$ iff customer i is served from facility j.

1.3 Scheduling Problem

Input: n tasks and one machine, duration p_i for each task i.
Output: a minimum makespan schedule of the tasks on the machine.

$$\begin{aligned}
\min\ & s_{n+1} & \\
\text{s.t. } & s_{n+1} \geq s_j + p_j & j = 1..n \\
& s_j - s_i \geq M x_{ij} + (p_i - M) & i,j = 1..n \\
& x_{ij} + x_{ji} = 1 & i,j = 1..n; i < j \\
& s_j \in \mathbb{Z}_+ & j = 1..n+1 \\
& x_{ij} \in \{0,1\} & i,j = 1..n
\end{aligned}$$

where $x_{ij} = 1$ iff task i precede task j, s_i is the starting time of task i, s_{n+1} is the makespan, and $M \geq \sum_{i=1}^{n} p_i$.

1.4 K-median Problem

Input: n data points, distance d_{ij} between each pair of points (i,j), a number $0 < k < n$.
Output: a selection of k points, the centers, minimizing the sum of the distances between each point and the nearest center.

$$\begin{aligned}
\min\ & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} y_{ij} & \\
\text{s.t. } & \sum_{j=1}^{n} y_{ij} = 1 & i = 1..n \\
& y_{ij} \leq x_j & i,j = 1..n \\
& \sum_{j=1}^{n} x_j = k & \\
& y_{ij} \in \{0,1\}, x_j \in \{0,1\} & i,j = 1..n
\end{aligned}$$

where $y_j = 1$ iff point j is a center and $x_{ij} = 1$ if j is the nearest center of i.

1.5 Market Split Problem

Input: 1 company with 2 divisions, m products, n retailers, availability d_j for each product j, demand a_{ij} of each retailer i for each product j.
Output: an assignement of the retailers to the divisions approaching a 50/50 production split for each product.

$$\begin{aligned}
\min\ & \sum_{j=1}^{m} s_j^+ + s_j^- & \\
\text{s.t. } & \sum_{i=1}^{n} a_{ij} x_i + s_j^+ - s_j^- = \frac{d_j}{2} & j = 1..m \\
& x_i \in \{0,1\} & i = 1..n \\
& s_j^+ \geq 0, s_j^- \geq 0 & j = 1..m
\end{aligned}$$

where $x_i = 1$ iff retailer i is assigned to division 1, $s_j^+ - s_j^-$ is the slack value (s_j^+ is the positive part and s_j^- is the negative part) between the volume produced by division 1 and the desired volume ($d_j * 50\%$).

1.6 Capacitated Transhipment Problem

Input: directed graph $G = (V, A)$, demand or supply b_i at each node n, capacity h_{ij} and unit flow cost c_{ij} on each arc (i, j).
Output: a minimum cost integer flow to satisfy the demand.

$$\begin{aligned}
\min \ & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{s.t. } & \sum_{j\in\delta^+(i)} x_{ij} - \sum_{j\in\delta^-(i)} x_{ij} = b_i && i \in V \\
& x_{ij} \le h_{ij} && (i,j) \in A \\
& x_{ij} \in \mathbb{Z}_+ && (i,j) \in A
\end{aligned}$$

where x_{ij} the flow on arc (i, j)

1.7 Traveling Salesman Problem

Input: a set V of cities, $E = V^2$, a distance $c_{ij} = c_{ji}$ between each cities i and j.
Output: a tour visiting every city exactly once.

$$\begin{aligned}
\min \ & \sum_{e\in E} c_e x_e \\
\text{s.t. } & \sum_{e\in E | i\in e} x_e = 2 && i \in V \\
& \sum_{\delta(Q)} x_e \ge 2 && \emptyset \subsetneq Q \subsetneq V \\
& x_e \in \{0,1\} && e \in E
\end{aligned}$$

where $x_e = 1$ iff the edge e belongs to the tour.

1.8 Uncapacitated Lot Sizing Problem

Input: n time periods, fix production cost f_t, unit production cost p_t, unit storage cost h_t at period t, demand d_t at each period t.
Output: a minimum (production and storage) cost production plan that satsify the demand.

$$\begin{aligned}
\min \ & \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t. } & s_{t-1} + x_t = d_t + s_t && t = 1..n \\
& x_t \le M_t y_t && t = 1..n \\
& y_t \in \{0,1\} && t = 1..n \\
& s_t, x_t \ge 0 && t = 1, \dots, n \\
& s_0 = 0
\end{aligned}$$

where $y_t = 1$ iff production occurs during period t, x_t is the amount produced during period t, y_t is the amount stored at the beginning of period t, and where $M_t \ge \sum_{i=t}^{n} d_i$ for each period t.

$$\begin{aligned}
\min \ & \sum_{t=1}^{n} f_t y_t + \sum_{i=1}^{n} \sum_{t=i}^{n} p_i z_{it} + \sum_{i=1}^{n} \sum_{t=i+1}^{n} \sum_{j=i}^{t-1} h_j z_{it} \\
\text{s.t. } & \sum_{i=1}^{t} z_{it} = d_t && t = 1..n \\
& z_{it} \le d_t y_i && i = 1..n; t = i..n \\
& y_t \in \{0,1\} && t = 1..n \\
& z_{it} \ge 0 && i = 1..n; t = i..n
\end{aligned}$$

where z_{it} is the amount produced in period i to satisfy demand of period t.

1.9 Bin Packing Problem

Input: n items, weight $w_j \geq 0$ for each item j, m containers each of capacity $K \geq 0$.
Output: an assignment of the items to a minimum number of containers.

$$
\begin{aligned}
\min \ & \sum_{i=1}^{n} y_i \\
\text{s.t.} \ & \sum_{j=1}^{m} w_j x_{ij} \leq K y_i && i = 1..n \\
& \sum_{i=1}^{n} x_{ij} = 1 && j = 1..m \\
& x_{ij} \in \{0,1\} && i = 1..n;\ j = 1..m \\
& y_i \in \{0,1\} && i = 1..n
\end{aligned}
$$

where $y_i = 1$ iff container i is used and $x_{ij} = 1$ iff item j is assigned to container i.
The Dantzig-Wolfe formulation (can be solved by delayed column generation):

$$
\begin{aligned}
\min \ & \sum_{s \in \mathscr{S}} x_s \\
\text{s.t.} \ & \sum_{s \in \mathscr{S}} a_{js} x_s = 1 && j = 1..n \\
& x_s \in \{0,1\} && s \in \mathscr{S}
\end{aligned}
$$

where $\mathscr{S} = \{s \subset \{1,\ldots,n\} \mid \sum_{j \in s} w_j \leq K\}$ is the set of all possible arrangements of items to one container, and $x_s = 1$ iff all the items in s (and no others) are assigned to the same container.

1.10 Multi 0-1 Knapsack Problem

Input: n items, value c_j and weight $w_j \geq 0$ for each item j, m containers, capacity $K_i \geq 0$ for each container i.
Output: a maximum value subset of items to assign to the containers such that the capacity of each container is not exceeded.

$$
\begin{aligned}
\max \ & \sum_{i=1}^{m} \sum_{j=1}^{n} c_j x_{ij} \\
\text{s.t.} \ & \sum_{j=1}^{n} w_j x_{ij} \leq K_i && i = 1..m \\
& \sum_{i=1}^{m} x_{ij} \leq 1 && j = 1..n \\
& x_{ij} \in \{0,1\} && j = 1..n, i = 1..m
\end{aligned}
$$

with $x_{ij} = 1$ iff item j is assigned to container i

The lagrangian dual:

$$
\begin{aligned}
&\min z_\pi \\
&\text{s.t. } \pi_i \geq 0 \quad i = 1..m \\
&\quad z_\pi = \quad \max \sum_{i=1}^{m} \sum_{j=1}^{n} c_j x_{ij} - \sum_{i=1}^{m} \pi_i (\sum_{j=1}^{n} w_j x_{ij} - K_i) \\
&\qquad\qquad \text{s.t. } \sum_{i=1}^{m} x_{ij} \leq 1 \qquad\qquad\qquad j = 1..n \\
&\qquad\qquad\quad x_{ij} \in \{0,1\} \qquad\qquad j = 1..n, i = 1..m
\end{aligned}
$$

where π_i is the penalty for violating the capacity of container i

An other relaxation (dualization of the coupling constraints):

$$
\begin{aligned}
&\min \sum_{i=1}^{m} z_u^j \quad + \sum_{j=1}^{n} u_j \\
&\text{s.t. } u_j \geq 0 \quad j = 1..n \\
&\quad z_u^i = \quad \max \sum_{j=1}^{n} (c_j - u_j) x_{ij} \\
&\qquad\qquad \text{s.t. } \sum_{j=1}^{n} w_j x_{ij} \leq K_i \qquad\qquad i = 1..m \\
&\qquad\qquad\quad x_{ij} \in \{0,1\} \qquad j = 1..n, i = 1..m
\end{aligned}
$$

2 Outline

2.1 Modeling booleans with binary variables

indicator	linearization
$\delta = 1 \implies y \geq a$	$y \geq L + (a - L)\delta$
$\delta = 0 \implies y \geq a$	$y \geq L + (a - L)(1 - \delta)$
$y < a \implies \delta = 1$	$y \geq L + (a - L)(1 - \delta)$
$\delta = 1 \implies y > a$	$y \geq L + (a + \epsilon - L)\delta$
$\delta = 1 \implies y \leq a$	$y \leq U + (a - U)\delta$
$\delta = 1 \iff y > a$	$m + (a + \epsilon - m)\delta \leq y \leq a + (U - a)\delta$
$\delta = 1 \implies y \geq x$ with $x \in [m, M]$, $m \geq L$	$y \geq x + (L - M)(1 - \delta)$

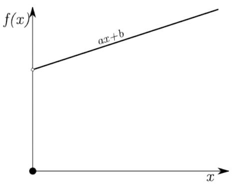
where $\delta \in \{0,1\}$, $y \in [L, U] \subseteq \mathbb{R}$, $L < a < U$, $\epsilon > 0$ small

- Given the optimization sense, it is often enough to enforce implication instead of equivalence, ex: $\min\{y \mid \delta \in \Delta, \delta = 1 \iff y > a\} = \min\{y \mid \delta \in \Delta, \delta = 1 \implies y > a\}$

2.2 Modeling logic/numeric relations with binary variables

condition	example	linearization
exclusive disjunction	*either c or* $\neg c$	$\delta = 1 \iff c$
exclusive disjunction	*either* c_1 *or* c_2	$\delta_1 + \delta_2 = 1$
disjunction	c_1 *or* c_2	$\delta_1 + \delta_2 \geq 1$
dependency	*if* c_1 *then* c_2	$\delta_2 \geq \delta_1$
exclusive alternative	*exactly* 1 *out of* n	$\sum_{i=1}^{n} \delta_i = 1$
counter	*exactly* k *out of* n	$\sum_{i=1}^{n} \delta_i = k$
bound	*at least* k *out of* n	$\sum_{i=1}^{n} \delta_i \geq k$
bound	*at most* k *out of* n	$\sum_{i=1}^{n} \delta_i \leq k$

2.3 Modeling non-linear functions with binary variables



set-up value:

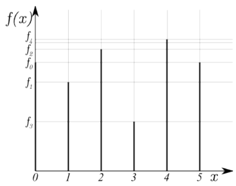
$f : [0, U] \subseteq \mathbb{R}_+ \to \mathbb{R}_+$

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ ax + b & \text{if } 0 < x \leq U \end{cases}$$

$f(x) = ax + b\delta$

$\epsilon\delta \leq x \leq U\delta$

$\delta \in \{0,1\}$



discrete value:

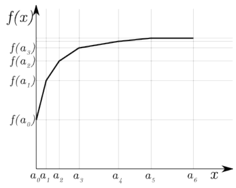
$f(x) = f_i$ if $x = i$

$f(x) = \sum_i \delta_i f_i$

$\sum_i i\delta_i = x$

$\sum_i \delta_i = 1$

$\delta_i \in \{0,1\}$ $i = 0..n$



piecewise linear:

$f(x) = \sum_i \lambda_i f(a_i)$

$\sum_i a_i \lambda_i = x$

$\sum_i \lambda_i = 1$

$\lambda_i \in [0,1]$ $i = 0..n$

with $SOS2(\lambda_i)$