ORIE 3310/5310 : Optimization II

1. Linear and Integer Programming

Prof. Oktay Günlük
Cornell University, School of ORIE

Spring 2024

Linear Programming Review (22/1/2024)

Notation

- A vector $x \in \mathbb{R}^n$ is an array of real numbers:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \text{or} \qquad x^T = [x_1, x_2, \ldots, x_n].$$

- The inner (or, "dot") product of two vectors $x, y \in \mathbb{R}^n$ is:

$$x^T y = \sum_{i=1}^{n} x_i y_i. \qquad \leftarrow \text{this is a number}$$

- A $m \times n$ matrix A is an array of real numbers a_{ij}:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

We say $A \in \mathbb{R}^{m \times n}$ (number of rows first, columns second)

Notation

- Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- The matrix-vector product of A and x is:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_{1i} x_i \\ \sum_{i=1}^n a_{2i} x_i \\ \vdots \\ \sum_{i=1}^n a_{mi} x_i \end{bmatrix}$$

- The number of columns of A must equal the number of rows of x.
- The number of rows of Ax equals the number of rows of A.

Notation

- Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- The matrix-vector product of A and x is:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_{1i} x_i \\ \sum_{i=1}^n a_{2i} x_i \\ \vdots \\ \sum_{i=1}^n a_{mi} x_i \end{bmatrix}$$

- The number of columns of A must equal the number of rows of x.
- The number of rows of Ax equals the number of rows of A.

Notation

Given two matrices A and B, you can multiply them

$$AB$$

only if the number of columns of A is same as the number of rows of B:

$$A \in \mathbb{R}^{m\times k} \quad \text{and} \quad B \in \mathbb{R}^{k\times n}.$$

The resulting matrix $D = AB$ has m rows and n columns: $D \in \mathbb{R}^{m\times n}$

$$D = \underbrace{\begin{bmatrix} a_{11} & \dots & a_{1k} \\ \dots & \vdots & \vdots \\ a_{m1} & \dots & a_{mk} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} b_{11} & \dots & b_{1n} \\ b_{21} & \dots & b_{2n} \\ \dots & \vdots & \vdots \\ b_{k1} & \dots & b_{kn} \end{bmatrix}}_{B} = \begin{bmatrix} \cdots & d_{1j} = \sum_{i=1}^{k} a_{1i}b_{ij} & \cdots \\ \cdots & d_{2j} = \sum_{i=1}^{k} a_{2i}b_{ij} & \cdots \\ \cdots & \vdots & \cdots \\ \cdots & \underbrace{d_{mj} = \sum_{i=1}^{k} a_{mi}b_{ij}}_{j\text{th column of } D} & \cdots \end{bmatrix}$$

Remember matrix multiplication is not commutative:

AB is not same as BA

A linear programming problem

$$\begin{array}{lrl}
\text{minimize} & 2x_1 - x_2 + 4x_3 & \text{objective function} \\
\text{subject to} & \left.\begin{array}{r} x_1 + x_2 + x_4 \leq 2 \\ 3x_2 - x_3 = 5 \\ x_3 + x_4 \geq 3 \end{array}\right\} & \text{constraints} \\
 & x_1, x_2, x_3, x_4 \geq 0 & \text{non-negativity}
\end{array}$$

- $x^T = [x_1, x_2, x_3, x_4]$ are the decision variables.
- The objective function is linear and it can be written as $c^T x$, where

$$c^T = [2, -1, 4, 0]$$

- The constraints are linear equations and inequalities

  They have the form $a^T x = b$, or $a^T x \leq b$, or $a^T x \geq b$.

  – Example: The first constraint is of the form $a^T x \leq b$, with

$$a^T = [1, 1, 0, 1] \qquad \text{and} \qquad b = 2$$

A linear programming problem

$$
\begin{array}{lrl}
\text{minimize} & 2x_1 - x_2 + 4x_3 & \text{objective function} \\
\text{subject to} & \left.\begin{array}{r} x_1 + x_2 + x_4 \leq 2 \\ 3x_2 - x_3 = 5 \\ x_3 + x_4 \geq 3 \end{array}\right\} & \text{constraints} \\
 & x_1, x_2, x_3, x_4 \geq 0 & \text{non-negativity}
\end{array}
$$

- $x^T = [x_1, x_2, x_3, x_4]$ are the decision variables.
- The objective function is linear and it can be written as $c^T x$, where

$$c^T = [2, -1, 4, 0]$$

- The constraints are linear equations and inequalities

  They have the form $a^T x = b$, or $a^T x \leq b$, or $a^T x \geq b$.

  – Example: The first constraint is of the form $a^T x \leq b$, with

$$a^T = [1, 1, 0, 1] \qquad \text{and} \qquad b = 2$$

Writing LPs in " $\geq$ " form

$$
\begin{array}{lll}
\text{minimize} & 2x_1 - x_2 + 4x_3 & \text{objective function} \\
\text{subject to} & & \\
& \left.\begin{array}{r} x_1 + x_2 + x_4 \leq 2 \\ 3x_2 - x_3 = 5 \\ x_3 + x_4 \geq 3 \end{array}\right\} & \text{constraints} \\
& x_1, x_2, x_3, x_4 \geq 0 & \text{non-negativity}
\end{array}
$$

Notice that:

- The first constraint can also be rewritten as

$$-x_1 - x_2 - x_4 \geq -2$$

- The second constraint is equivalent to the two constraints:

$3x_2 - x_3 \geq 5$ and $3x_2 - x_3 \leq 5$ (i.e., $-3x_2 + x_3 \geq -5$).

Writing LPs in " $\geq$ " form

$$
\begin{array}{lll}
\text{minimize} & 2x_1 - x_2 + 4x_3 & \text{objective function} \\
\text{subject to} & & \\
& \left.\begin{array}{r} x_1 + x_2 + x_4 \leq 2 \\ 3x_2 - x_3 = 5 \\ x_3 + x_4 \geq 3 \end{array}\right\} & \text{constraints} \\
& x_1, x_2, x_3, x_4 \geq 0 & \text{non-negativity}
\end{array}
$$

Notice that:

- The first constraint can also be rewritten as

$$-x_1 - x_2 - x_4 \geq -2$$

- The second constraint is equivalent to the two constraints:

$$3x_2 - x_3 \geq 5 \quad \text{and} \quad 3x_2 - x_3 \leq 5 \quad (\text{i.e., } -3x_2 + x_3 \geq -5).$$

Writing LPs in " $\geq$ " form

$$
\begin{array}{lll}
\text{minimize} & 2x_1 - x_2 + 4x_3 & \text{objective function} \\
\text{subject to} & \left.\begin{aligned} x_1 + x_2 + x_4 &\leq 2 \\ 3x_2 - x_3 &= 5 \\ x_3 + x_4 &\geq 3 \end{aligned}\right\} & \text{constraints} \\
 & x_1, x_2, x_3, x_4 \geq 0 & \text{non-negativity}
\end{array}
$$

Notice that:

- The first constraint can also be rewritten as

$$-x_1 - x_2 - x_4 \geq -2$$

- The second constraint is equivalent to the two constraints:

$$3x_2 - x_3 \geq 5 \quad \text{and} \quad 3x_2 - x_3 \leq 5 \quad (\text{i.e., } -3x_2 + x_3 \geq -5).$$

Therefore any linear programming problem can then be written as:

$$\text{minimize} \quad c^T x \quad \text{subject to} \quad Ax \geq b, \quad x \geq 0.$$

Example LP in matrix notation

- We can write any linear programming problem in " $\geq$ " form.

$$
\begin{array}{lrcl}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s. t.} & x_1 + x_2 + x_4 & \leq & 2 \\
& 3x_2 - x_3 & = & 5 \\
& x_3 + x_4 & \geq & 3 \\
& x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

Example LP in matrix notation

- We can write any linear programming problem in "$\geq$" form.

$$
\begin{array}{llcl}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & x_1 + x_2 + x_4 & \leq & 2 \\
 & 3x_2 - x_3 & = & 5 \\
 & x_3 + x_4 & \geq & 3 \\
 & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
\qquad
\begin{array}{llcr}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & -x_1 - x_2 - x_4 & \geq & -2 \\
 & 3x_2 - x_3 & \geq & 5 \\
 & -3x_2 + x_3 & \geq & -5 \\
 & x_3 + x_4 & \geq & 3 \\
 & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

- A vector x is feasible to the first LP if and only if it is feasible to the second LP.

Example LP in matrix notation

- We can write any linear programming problem in "$\geq$" form.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b, \; x \geq 0 \end{aligned}$$

where

$$c^T = (2, \; -1, \; 4, \; 0),$$

$$A = \begin{bmatrix} -1 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

and, $b^T = (-2, \; 5, \; -5, \; 3)$.

$$\begin{aligned} \min \quad & 2x_1 - x_2 + 4x_3 & & \\ \text{s.t.} \quad & -x_1 - x_2 - x_4 & \geq & -2 \\ & 3x_2 - x_3 & \geq & 5 \\ & -3x_2 + x_3 & \geq & -5 \\ & x_3 + x_4 & \geq & 3 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{aligned}$$

Example LP in matrix notation

- We can write any linear programming problem in "$\geq$" form.

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b, \ x \geq 0 \end{aligned}$$

where

$$c^T = (2, \ -1, \ 4, \ 0),$$

$$A = \begin{bmatrix} -1 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

and, $b^T = (-2, \ 5, \ -5, \ 3)$.

$$\begin{array}{lrcr} \min & 2x_1 - x_2 + 4x_3 & & \\ \text{s.t.} & -x_1 - x_2 - x_4 & \geq & -2 \\ & 3x_2 - x_3 & \geq & 5 \\ & -3x_2 + x_3 & \geq & -5 \\ & x_3 + x_4 & \geq & 3 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

Data for this LP:

- n variables, $c \in \mathbb{R}^n$
- m constraints, $b \in \mathbb{R}^m$
- $A \in \mathbb{R}^{m \times n}$

Example LP in matrix notation cont.

- Sometimes it is more natural to write LPs in " $\leq$ " form.

$$\begin{array}{lrcr} \min & 2x_1 - x_2 + 4x_3 & & \\ \text{s. t.} & -x_1 - x_2 - x_4 & \geq & -2 \\ & 3x_2 - x_3 & \geq & 5 \\ & -3x_2 + x_3 & \geq & -5 \\ & x_3 + x_4 & \geq & 3 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{array} \qquad \begin{array}{lrcr} \min & 2x_1 - x_2 + 4x_3 & & \\ \text{s. t.} & x_1 + x_2 + x_4 & \leq & 2 \\ & -3x_2 + x_3 & \leq & -5 \\ & 3x_2 - x_3 & \leq & 5 \\ & -x_3 - x_4 & \leq & -3 \\ & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

$$\begin{array}{ll} \min & c^T x \\ \text{s. t.} & Ax \geq b \\ & x \geq 0 \end{array} \longrightarrow \begin{array}{ll} \min & c^T x \\ \text{s. t.} & -Ax \leq -b \\ & x \geq 0 \end{array}$$

Example LP in matrix notation cont.

- Sometimes it is more natural to write LPs in "$\leq$" form.

$$
\begin{array}{lrcr}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & -x_1 - x_2 - x_4 & \geq & -2 \\
& 3x_2 - x_3 & \geq & 5 \\
& -3x_2 + x_3 & \geq & -5 \\
& x_3 + x_4 & \geq & 3 \\
& x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
\qquad
\begin{array}{lrcr}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & x_1 + x_2 + x_4 & \leq & 2 \\
& -3x_2 + x_3 & \leq & -5 \\
& 3x_2 - x_3 & \leq & 5 \\
& -x_3 - x_4 & \leq & -3 \\
& x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

$$
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & Ax \geq b \\
& x \geq 0
\end{array}
\longrightarrow
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & -Ax \leq -b \\
& x \geq 0
\end{array}
$$

Example LP in matrix notation cont. again

- We can also write LPs in (standard) " $=$ " form.

$$
\begin{array}{llcl}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & x_1 + x_2 + x_4 & \leq & 2 \\
& 3x_2 - x_3 & = & 5 \\
& x_3 + x_4 & \geq & 3 \\
& x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
\qquad
\begin{array}{llcl}
\min & 2x_1 - x_2 + 4x_3 & & \\
\text{s.t.} & x_1 + x_2 + x_4 + s_1 & = & 2 \\
& 3x_2 - x_3 & = & 5 \\
& x_3 + x_4 - s_3 & = & 3 \\
& x_1, x_2, x_3, x_4 & \geq & 0 \\
& s_1, s_3 & \geq & 0
\end{array}
$$

$$
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & Ax \leq b \\
& x \geq 0
\end{array}
\longrightarrow
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & Ax + Is = b \\
& x \geq 0, \quad s \geq 0
\end{array}
$$

$$
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & Ax \geq b \\
& x \geq 0
\end{array}
\longrightarrow
\begin{array}{ll}
\min & c^T x \\
\text{s.t.} & Ax - Is = b \\
& x \geq 0, \quad s \geq 0
\end{array}
$$

Example LP in matrix notation cont. again

- We can also write LPs in (standard) "$=$" form.

$$\begin{array}{lrcl} \min & 2x_1 - x_2 + 4x_3 & & \\ \text{s.t.} & x_1 + x_2 + x_4 & \le & 2 \\ & 3x_2 - x_3 & = & 5 \\ & x_3 + x_4 & \ge & 3 \\ & x_1, x_2, x_3, x_4 & \ge & 0 \end{array}$$

$$\begin{array}{lrcl} \min & 2x_1 - x_2 + 4x_3 & & \\ \text{s.t.} & x_1 + x_2 + x_4 + s_1 & = & 2 \\ & 3x_2 - x_3 & = & 5 \\ & x_3 + x_4 - s_3 & = & 3 \\ & x_1, x_2, x_3, x_4 & \ge & 0 \\ & s_1, s_3 & \ge & 0 \end{array}$$

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \longrightarrow \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax + Is = b \\ & x \ge 0, \quad s \ge 0 \end{array}$$

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \ge b \\ & x \ge 0 \end{array} \longrightarrow \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax - Is = b \\ & x \ge 0, \quad s \ge 0 \end{array}$$

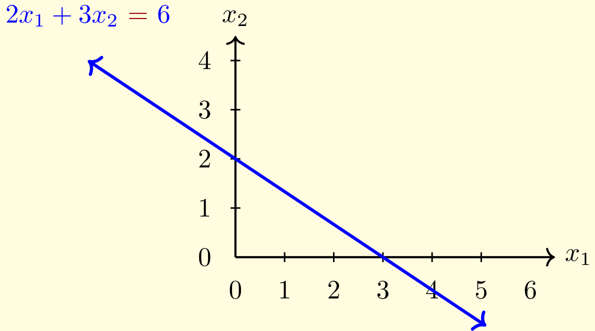
Halfspaces

- A hyperplane in $\mathbb{R}^n$ is the set points $x \in \mathbb{R}^n$ that satisfy the equation

$$a^T x \;=\; b \iff a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b$$

This set of points divide the space $\mathbb{R}^n$ into two halfspaces.

Example: Consider the line $2x_1 + 3x_2 = 6$ in $\mathbb{R}^2$



Halfspaces

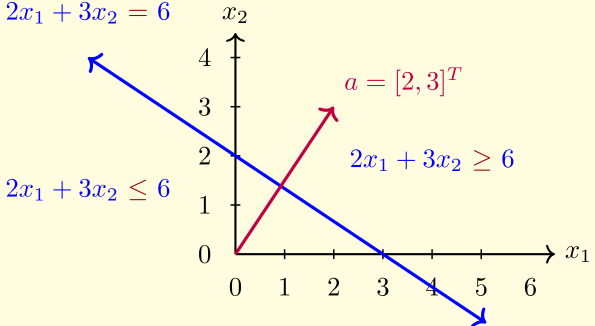
- All points in one of the halfspaces satisfy the inequality

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \;\geq\; b$$

- Points in the other satisfy

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \;\leq\; b.$$

- The vector $a = (a_1, a_2, \ldots, a_n)^T$ points in the direction where the points satisfying $a_1x_1 + a_2x_2 + \ldots + a_nx_n \;\geq\; b$ are.

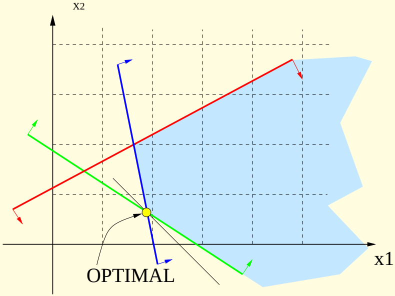


Geometrically an LP looks like

Intersecting all constraints (halfspaces) gives the feasible set P (polyhedron)

Solving the LP $\equiv$ find a point $x \in P$ with minimum $c^T x$.

$$
\begin{array}{ll}
\min & 3x_1 + 3x_2 \\
 & 2x_1 + 3x_2 \geq 6 \\
 & -x_1 + 2x_2 \leq 2 \\
 & 5x_1 + 1x_2 \geq 10
\end{array}
$$



If there is an opt. solution, then there is an extreme point opt. solution

$$\underbrace{2x_1 + 3x_2 = 6,\ 5x_1 + x_2 = 10}_{\text{both constraints satisfied at equality (tight)}} \implies x = \left(\frac{24}{13}, \frac{10}{13}\right)$$

Optimal solutions of LPs

- Given a generic LP with n variables:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

- There are four possible (mutually exclusive) outcomes:

(a) The feasible set P might be empty (problem is infeasible)

$$P = \left\{ x \in \mathbb{R}^n \,:\, Ax \geq b,\ x \geq 0 \right\} = \emptyset$$

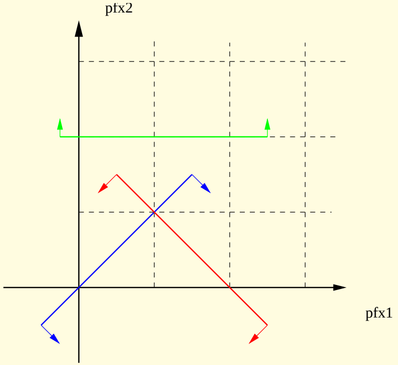
(b) There might be a unique optimal solution (one and only one).

(c) There might be multiple optimal solutions.
(all with the same objective value)

(d) The objective value might be unbounded ($-\infty$): No feasible solution is optimal.

Example: Infeasible problem

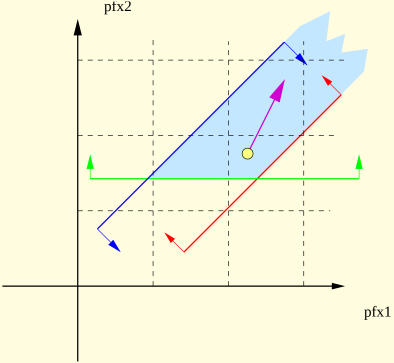
$$
\begin{aligned}
\min \quad & x_1 + x_2 \\
& x_1 - x_2 \geq 0 \\
& x_1 + x_2 \leq 2 \\
& x_2 \geq 2
\end{aligned}
$$



$$P = \{x \in \mathbb{R}^2 \ : \ x_1 - x_2 \geq 0, \ x_1 + x_2 \leq 2, \ x_2 \geq 2\} = \emptyset$$

Example: Unbounded problem

$$
\begin{aligned}
\max \quad & x_1 + 2x_2 \\
& 2x_1 - 2x_2 \geq -1 \\
& x_1 - x_2 \leq 1 \\
& x_2 \geq 3/2
\end{aligned}
$$



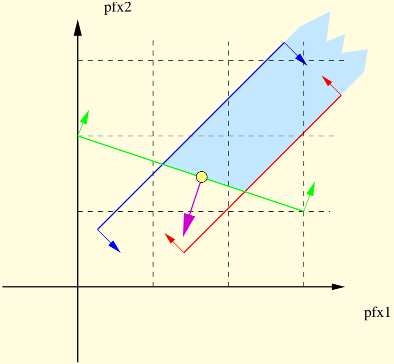
1. $P = \{x \in \mathbb{R}^2 \,:\, 2x_1 - 2x_2 \geq -1,\ x_1 - x_2 \leq 1,\ x_2 \geq 3/2\} \neq \emptyset$
2. For any $x \in P$, there exists a $x' \in P$ such that $c^T x' \geq c^T x + 1$

Note: If an LP is unbounded, its feasible region must be unbounded.

. But, an LP can be bounded while its feasible region is unbounded.

Example: Multiple optima

$$
\begin{aligned}
\min \quad & x_1 + 3x_2 \\
& 2x_1 - 2x_2 \ge -1 \\
& x_1 - x_2 \le 1 \\
& x_1 + 3x_2 \ge 6
\end{aligned}
$$



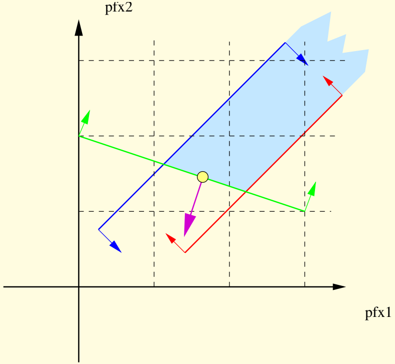
$$P = \{x \in \mathbb{R}^2 \;:\; 2x_1 - 2x_2 \ge -1,\; x_1 - x_2 \le 1,\; x_1 + 3x_2 = 6\} \neq \emptyset$$

In conclusion an LP is either

- Feasible or infeasible.
- If feasible, the optimal value is either bounded or unbounded.
- If feasible and bounded, optimial solution may not be unique.

Example: Multiple optima

$$
\begin{aligned}
\min \quad & x_1 + 3x_2 \\
& 2x_1 - 2x_2 \geq -1 \\
& x_1 - x_2 \leq 1 \\
& x_1 + 3x_2 \geq 6
\end{aligned}
$$



$$P = \{x \in \mathbb{R}^2 \ : \ 2x_1 - 2x_2 \geq -1, \ x_1 - x_2 \leq 1, \ x_1 + 3x_2 = 6\} \neq \emptyset$$

In conclusion an LP is either

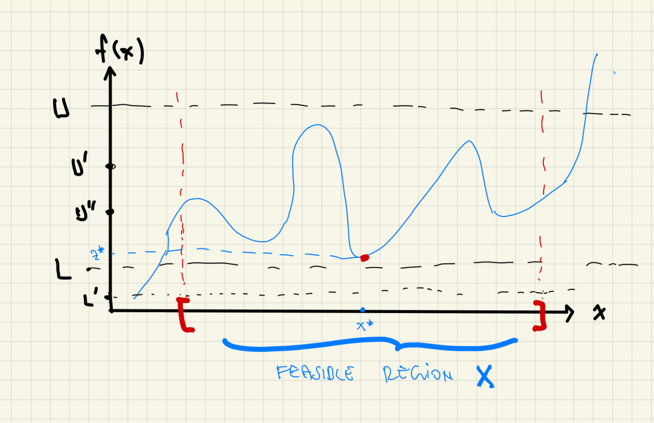
- Feasible or infeasible.
- If feasible, the optimal value is either bounded or unbounded.
- If feasible and bounded, optimial solution may not be unique.

Upper/lower bounds and LP duality

Upper and lower bounds for minimization problems

- Consider an optimization problem:

$$\text{the optimal value} \longrightarrow \quad z^* = \min_{x \in X} \ f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$



- U is an upper bound for z^*, (so are U' and U'')
- L is a lower bound for z^* (L' is a lower bound as well)

Upper bounds for minimization problems

- Given an optimization problem:

$$z^* = \min_{x \in X} \ f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- An upper bound $U \in \mathbb{R}$ is a number that can be certified to be greater than or equal to z^*

In other words: $$\min_{x \in X} \ f(x) \leq U$$

- This does not mean

$$f(x) \leq U \qquad \text{for all } x \in X$$

It is perfectly OK if there are solutions $x' \in X$ such that

$$f(x') > U$$

- We only need the condition to hold for the optimal solution $x^* \in X$

$$f(x^*) \leq U$$

- We can obtain upper bounds without solving the problem to optimality.
  - For example, **any** feasible point $\bar{x} \in X$ gives an upper bound

$$U = f(\bar{x})$$

Upper bounds for minimization problems

- Given an optimization problem:

$$z^* = \min_{x \in X} \; f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- An upper bound $U \in \mathbb{R}$ is a number that can be certified to be greater than or equal to z^*

In other words: $$\min_{x \in X} \; f(x) \leq U$$

- This does not mean

$$f(x) \leq U \qquad \text{for all } x \in X$$

It is perfectly OK if there are solutions $x' \in X$ such that

$$f(x') > U$$

- We only need the condition to hold for the optimal solution $x^* \in X$

$$f(x^*) \leq U$$

- We can obtain upper bounds without solving the problem to optimality.
  - For example, **any** feasible point $\bar{x} \in X$ gives an upper bound

$$U = f(\bar{x})$$

Upper bounds for minimization problems

- Given an optimization problem:

$$z^* = \min_{x \in X} \; f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- An upper bound $U \in \mathbb{R}$ is a number that can be certified to be greater than or equal to z^*

In other words: $$\min_{x \in X} \; f(x) \leq U$$

- This does not mean

$$f(x) \leq U \qquad \text{for all } x \in X$$

It is perfectly OK if there are solutions $x' \in X$ such that

$$f(x') > U$$

- We only need the condition to hold for the optimal solution $x^* \in X$

$$f(x^*) \leq U$$

- We can obtain upper bounds without solving the problem to optimality.
  - For example, **any** feasible point $\bar{x} \in X$ gives an upper bound

$$U = f(\bar{x})$$

Upper bounds for minimization problems: LP example

- Consider the LP with 4 variables:

$$
\begin{aligned}
z^* = \min \quad & 2x_1 - x_2 + 4x_3 \\
\text{s.t.} \quad & x_1 + x_2 + x_4 \le 6 \\
& 3x_2 - x_3 = 5 \\
& x_3 + x_4 \ge 2 \\
& x_1, x_2, x_3, x_4 \ge 0
\end{aligned}
$$

- And consider the point (or, solution): $\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}$

- The point $\bar{x}$ is feasible as it satisfies all constraints.
- It has an objective function value of : $2(0) - 1(2) + 4(1) + 0(2) = 2$
- Since we are minimizing, we conclude that z^* cannot be larger than 2.
- Therefore, $U = 2$ is **an** upper bound for this LP. $\longleftarrow \quad z^* \le 2$

A slight detour

The LP:

$$
\begin{array}{rrrcl}
z^* = & \min & 2x_1 - x_2 + 4x_3 & & \\
& \text{s.t.} & x_1 + x_2 + x_4 & \leq & 6 \\
& & 3x_2 - x_3 & = & 5 \\
& & x_3 + x_4 & \geq & 2 \\
& & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

A feasible solution:

$$
\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \quad \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}
$$

with

$c^T \bar{x} = 2 \longleftarrow$ an UB on z^*

Question: Is this solution optimal? $\rightarrow$ Not sure: maybe, maybe not.

Question: Is this an extreme point solution? $\rightarrow$ NO!

- The feasible set lives in $\mathbb{R}^4$.
- An ext. pt. must satisfy 4 linearly indep. constraints as equality
- $\bar{x}$ only satisfies two: $3x_2 - x_3 = 5$ and $x_1 \geq 0$ constraint

Here is an ex. pt: $\hat{x}^T = [0, 5/3, 0, 2]$ with $c^T \hat{x} = -5/3 \leftarrow$ a better UB

A slight detour

The LP:

$$\begin{aligned} z^* = \min \quad & 2x_1 - x_2 + 4x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_4 \le 6 \\ & 3x_2 - x_3 = 5 \\ & x_3 + x_4 \ge 2 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

A feasible solution:

$$\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}$$

with

$c^T \bar{x} = 2 \longleftarrow$ an UB on z^*

Question: Is this solution optimal? $\rightarrow$ Not sure: maybe, maybe not.

Question: Is this an extreme point solution? $\rightarrow$ NO!

- The feasible set lives in $\mathbb{R}^4$.
- An ext. pt. must satisfy 4 linearly indep. constraints as equality
- $\bar{x}$ only satisfies two: $3x_2 - x_3 = 5$ and $x_1 \ge 0$ constraint

Here is an ex. pt: $\hat{x}^T = [0, 5/3, 0, 2]$ with $c^T \hat{x} = -5/3 \leftarrow$ a better UB

A slight detour

The LP:

$$\begin{array}{rrrcl} z^* = & \min & 2x_1 - x_2 + 4x_3 & & \\ & \text{s.t.} & x_1 + x_2 + x_4 & \leq & 6 \\ & & 3x_2 - x_3 & = & 5 \\ & & x_3 + x_4 & \geq & 2 \\ & & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

A feasible solution:

$$\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}$$

with

$c^T \bar{x} = 2 \longleftarrow$ an UB on z^*

Question: Is this solution optimal? $\rightarrow$ Not sure: maybe, maybe not.

Question: Is this an extreme point solution? $\rightarrow$ NO!

- The feasible set lives in $\mathbb{R}^4$.
- An ext. pt. must satisfy 4 linearly indep. constraints as equality
- $\bar{x}$ only satisfies two: $3x_2 - x_3 = 5$ and $x_1 \geq 0$ constraint

Here is an ex. pt: $\hat{x}^T = [0, 5/3, 0, 2]$ with $c^T \hat{x} = -5/3 \leftarrow$ a better UB

A slight detour

The LP:

$$
\begin{aligned}
z^* = \min \quad & 2x_1 - x_2 + 4x_3 \\
\text{s.t.} \quad & x_1 + x_2 + x_4 \le 6 \\
& 3x_2 - x_3 = 5 \\
& x_3 + x_4 \ge 2 \\
& x_1, x_2, x_3, x_4 \ge 0
\end{aligned}
$$

A feasible solution:

$$
\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}
$$

with

$c^T\bar{x} = 2 \longleftarrow$ an UB on z^*

Question: Is this solution optimal? $\rightarrow$ Not sure: maybe, maybe not.

Question: Is this an extreme point solution? $\rightarrow$ NO!

- The feasible set lives in $\mathbb{R}^4$.
- An ext. pt. must satisfy 4 linearly indep. constraints as equality
- $\bar{x}$ only satisfies two: $3x_2 - x_3 = 5$ and $x_1 \ge 0$ constraint

Here is an ex. pt: $\hat{x}^T = [0, 5/3, 0, 2]$ with $c^T\hat{x} = -5/3 \leftarrow$ a better UB

A slight detour

The LP:

$$\begin{aligned} z^* \;=\; \min \quad & 2x_1 - x_2 + 4x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_4 \;\leq\; 6 \\ & 3x_2 - x_3 \;=\; 5 \\ & x_3 + x_4 \;\geq\; 2 \\ & x_1, x_2, x_3, x_4 \;\geq\; 0 \end{aligned}$$

A feasible solution:

$$\bar{x} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 2 \end{bmatrix} \begin{array}{l} \leftarrow \bar{x}_1 \\ \leftarrow \bar{x}_2 \\ \leftarrow \bar{x}_3 \\ \leftarrow \bar{x}_4 \end{array}$$

with

$c^T\bar{x} = 2 \longleftarrow$ an UB on z^*

Question: Is this solution optimal? $\rightarrow$ Not sure: maybe, maybe not.

Question: Is this an extreme point solution? $\rightarrow$ NO!

- The feasible set lives in $\mathbb{R}^4$.
- An ext. pt. must satisfy 4 linearly indep. constraints as equality
- $\bar{x}$ only satisfies two: $3x_2 - x_3 = 5$ and $x_1 \geq 0$ constraint

Here is an ex. pt: $\hat{x}^T = [0, 5/3, 0, 2]$ with $c^T\hat{x} = -5/3 \leftarrow$ a better UB

Lower bounds for minimization problems

- Again consider the optimization problem:

$$z^* = \min_{x \in X} \; f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- A lower bound $L \in \mathbb{R}$ is a number that can be certified to be less than or equal to z^*.

In other words: $\displaystyle \min_{x \in X} \; f(x) \geq L$

- We have to make sure that the best solution has to have an objective value larger than or equal to L.
- Which means that

$$f(x) \geq L \quad \text{for all} \;\; x \in X$$

- Finding lower bounds is usually complicated.
  - For LP problems, remember the weak duality theorem:
    (Feasible solutions to the dual give lower bounds for the primal LP.)

Lower bounds for minimization problems: LP Example

- Consider the following simple (feasible) LP with 2 variables:

$$
\begin{aligned}
z^* = \min \quad & 2x_1 + 6x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \geq 2 \\
& x_2 \geq 9.
\end{aligned}
$$

- We would like to find a lower bound L for z^*.
- L must be less than the objective value of **all** feasible solutions.
- Consider an arbitrary feasible point $\bar{x} = [\bar{x}_1, \bar{x}_2]$
- As $\bar{x}$ is feasible, we know that $\bar{x}_1 + 3\bar{x}_2 \geq 2$ and therefore

$$2\bar{x}_1 + 6\bar{x}_2 \geq 4$$

- As this holds for **all** feasible points, we have:

$$z^* \geq 4 \quad \leftarrow \quad L = 4 \text{ is a lower bound on } z^*$$

Lower bounds for minimization problems: LP Example

- Consider the following simple (feasible) LP with 2 variables:

$$\begin{aligned} z^* \;=\; \min \quad & 2x_1 + 6x_2 \\ \text{s.t.} \quad & x_1 + 3x_2 \geq 2 \\ & x_2 \geq 9. \end{aligned}$$

- We would like to find a lower bound L for z^*.
- L must be less than the objective value of **all** feasible solutions.
- Consider an arbitrary feasible point $\bar{x} = [\bar{x}_1, \bar{x}_2]$
- As $\bar{x}$ is feasible, we know that $\bar{x}_1 + 3\bar{x}_2 \geq 2$ and therefore

$$2\bar{x}_1 + 6\bar{x}_2 \geq 4$$

- As this holds for **all** feasible points, we have:

$$z^* \geq 4 \quad \leftarrow \quad L = 4 \text{ is a lower bound on } z^*$$

Lower bounds for minimization: Another LP example

- Let's try a different example:

$$\begin{aligned} z^* \;=\; \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 \\ & \phantom{x_1 + {}} x_2 \geq 1. \end{aligned}$$

- How can we express the objective function in terms of the constraints?
- We can do it by summing the constraints as follows:

$$\begin{array}{r} x_1 + x_2 \geq 4 \\ \underline{+\ \ 2\ \cdot\ (x_2 \geq 1)} \\ =\ x_1 + 3x_2 \geq 6 \end{array}$$

- As this holds for **all** feasible points, we have:

$$z^* \geq 6 \qquad \leftarrow \quad L = 6 \text{ is a lower bound on } z^*$$

Lower bounds for minimization: Another LP example

- Let's try a different example:

$$\begin{aligned} z^* \;=\; \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 4 \\ & \qquad\;\; x_2 \geq 1. \end{aligned}$$

- How can we express the objective function in terms of the constraints?
- We can do it by summing the constraints as follows:

$$\begin{array}{r} x_1 + x_2 \geq 4 \\ \underline{+ \;\; 2 \;\cdot\; (x_2 \geq 1)} \\ = \; x_1 + 3x_2 \geq 6 \end{array}$$

- As this holds for **all** feasible points, we have:

$$z^* \geq 6 \qquad \leftarrow \quad L = 6 \text{ is a lower bound on } z^*$$

Lower bounds for minimization: One last LP example

- Here is a more interesting example:

$$
\begin{aligned}
z^* = \min \quad & x_1 + 3x_2 \\
\text{s.t.} \quad & x_1 + x_2 \geq 2 \\
& x_2 \geq 1 \\
& x_1 - x_2 \geq 3.
\end{aligned}
$$

- How can we express the objective function in terms of the constraints?

- For any choice of $p_1, p_2, p_3 \geq 0$ we can write

$$p_1 \cdot (x_1 + x_2 \geq 2) + p_2 \cdot (x_2 \geq 1) + p_3 \cdot (x_1 - x_2 \geq 3)$$

$$\Rightarrow \quad \underbrace{(p_1 + p_3)}_{\text{if } \quad =1} x_1 + \underbrace{(p_1 + p_2 - p_3)}_{\text{and if } \quad =3} x_2 \geq \underbrace{2p_1 + p_2 + 3p_3}_{\text{then you obtain } \; L}$$

- The lower bound L we obtain depends on the choice of p:
  - $p_1 = 1$, $p_2 = 2$, $p_3 = 0$ gives us the lower bound $L = 4$.
  - $p_1 = 0$, $p_2 = 4$, $p_3 = 1$ gives us the lower bound $L = 7$.

Lower bounds for minimization: One last LP example

- Here is a more interesting example:

$$\begin{aligned} z^* = \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & \qquad x_2 \geq 1 \\ & x_1 - x_2 \geq 3. \end{aligned}$$

- How can we express the objective function in terms of the constraints?

- For any choice of $p_1, p_2, p_3 \geq 0$ we can write

$$p_1 \cdot (x_1 + x_2 \geq 2) + p_2 \cdot (x_2 \geq 1) + p_3 \cdot (x_1 - x_2 \geq 3)$$

$$\Rightarrow \quad \underbrace{(p_1 + p_3)}_{\text{if } =1} x_1 + \underbrace{(p_1 + p_2 - p_3)}_{\text{and if } =3} x_2 \;\geq\; \underbrace{2p_1 + p_2 + 3p_3}_{\text{then you obtain } L}$$

- The lower bound L we obtain depends on the choice of p:
  - $p_1 = 1$, $p_2 = 2$, $p_3 = 0$ gives us the lower bound $L = 4$.
  - $p_1 = 0$, $p_2 = 4$, $p_3 = 1$ gives us the lower bound $L = 7$.

LP Duality

- A natural question is: how can we obtain the best lower bound on z^*?

$$\begin{aligned} z^* \;=\; \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & \qquad\; x_2 \geq 1 \\ & x_1 - x_2 \geq 3. \end{aligned}$$

- Maximize $2p_1 + p_2 + 3p_3$ over the constraints that p should satisfy:

$$\begin{aligned} w^* \;=\; \max \quad & 2p_1 + p_2 + 3p_3 \\ \text{s.t.} \quad & p_1 + p_3 = 1 \\ & p_1 + p_2 - p_3 = 3 \\ & p_1, p_2, p_3 \geq 0. \end{aligned}$$

- This is the dual LP and w^* gives the best lower bound on z^*

Primal

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \end{aligned}$$

Dual

$$\begin{aligned} \text{maximize} \quad & b^T p \\ \text{subject to} \quad & A^T p = c \\ & p \geq 0. \end{aligned}$$

LP Duality

- A natural question is: how can we obtain the best lower bound on z^*?

$$\begin{aligned} z^* = \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & x_2 \geq 1 \\ & x_1 - x_2 \geq 3. \end{aligned}$$

- Maximize $2p_1 + p_2 + 3p_3$ over the constraints that p should satisfy:

$$\begin{aligned} w^* = \max \quad & 2p_1 + p_2 + 3p_3 \\ \text{s.t.} \quad & p_1 + p_3 = 1 \\ & p_1 + p_2 - p_3 = 3 \\ & p_1, p_2, p_3 \geq 0. \end{aligned}$$

- This is the dual LP and w^* gives the best lower bound on z^*

Primal

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \end{aligned}$$

Dual

$$\begin{aligned} \text{maximize} \quad & b^T p \\ \text{subject to} \quad & A^T p = c \\ & p \geq 0. \end{aligned}$$

Non-negativity constraints

- Example:

$$\begin{aligned} z^* = \min \quad & x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & \phantom{x_1 +{}} x_2 \geq 1 \\ & x_1 - x_2 \geq 3 \\ & \phantom{x_1 +{}} x_1 \geq 0 \\ & \phantom{x_1 +{}} x_2 \geq 0. \end{aligned}$$

- For any choice of $p_1, p_2, p_3, p_4, p_5 \geq 0$ we can write

$$p_1 \cdot (x_1 + x_2 \geq 2) + p_2 \cdot (x_2 \geq 1) + p_3 \cdot (x_1 - x_2 \geq 3) + p_4 \cdot (x_1 \geq 0) + p_5 \cdot (x_2 \geq 0)$$

$$\Rightarrow \quad \underbrace{(p_1 + p_3 + p_4)}_{\text{if } =1} x_1 + \underbrace{(p_1 + p_2 - p_3 + p_5)}_{\text{and if } =3} x_2 \geq \underbrace{2p_1 + p_2 + 3p_3}_{\text{then you obtain } L}$$

- The lower bound L we obtain depends on the choice of p:
- Find the best p using linear programming

- To find the best lower bound on z^*:

$$
\begin{aligned}
z^* \;=\; \min \quad & x_1 + 3x_2 \\
\text{s. t.} \quad & x_1 + x_2 \ge 2 \\
& x_2 \ge 1 \\
& x_1 - x_2 \ge 3 \\
& x_1 \ge 0 \\
& x_2 \ge 0.
\end{aligned}
$$

- Maximize $2p_1 + p_2 + 3p_3$ over the constraints that p should satisfy:

$$
\begin{aligned}
\max \quad & 2p_1 + p_2 + 3p_3 \\
\text{s. t.} \quad & p_1 + p_3 + p_4 = 1 \\
& p_1 + p_2 - p_3 + p_5 = 3 \\
& p_1, p_2, p_3, p_4, p_5 \ge 0.
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
\max \quad & 2p_1 + p_2 + 3p_3 \\
\text{s. t.} \quad & p_1 + p_3 \le 1 \\
& p_1 + p_2 - p_3 \le 3 \\
& p_1, p_2, p_3 \ge 0.
\end{aligned}
$$

(1/24/2024)

Recap: Upper bounds and lower for minimization problems

Given an optimization problem:

$$z^* = \min_{x \in X} \; f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- An upper bound $U \in \mathbb{R}$ is a number that satisfies:

  $$\min_{x \in X} \; f(x) \leq U$$

  It is perfectly OK if there are solutions $x' \in X$ such that $f(x') > U$

  We only need the condition to hold for the optimal solution $x^* \in X$.

  Any feasible point $\bar{x} \in X$ gives an upper bound $U = f(\bar{x})$

- A lower bound $L \in \mathbb{R}$ is a number that satisfies:

  $$\min_{x \in X} \; f(x) \geq L$$

  Which means that

  $$f(x) \geq L \quad \text{for all} \quad x \in X$$

  Finding lower bounds is usually complicated.

- Small upper bounds and large lower bounds are better.

Recap: Upper bounds and lower for minimization problems

Given an optimization problem:

$$z^* = \min_{x \in X} \; f(x) \qquad \text{where} \quad X \subseteq \mathbb{R}^n$$

- An upper bound $U \in \mathbb{R}$ is a number that satisfies:

  $$\min_{x \in X} \; f(x) \leq U$$

  It is perfectly OK if there are solutions $x' \in X$ such that $f(x') > U$

  We only need the condition to hold for the optimal solution $x^* \in X$.

  Any feasible point $\bar{x} \in X$ gives an upper bound $U = f(\bar{x})$

- A lower bound $L \in \mathbb{R}$ is a number that satisfies:

  $$\min_{x \in X} \; f(x) \geq L$$

  Which means that

  $$f(x) \geq L \quad \text{for all} \;\; x \in X$$

  Finding lower bounds is usually complicated.

- Small upper bounds and large lower bounds are better.

Recap: Lower bounds via duality

- For any LP, there is a corresponding dual LP.

Primal LP

$$\begin{aligned} \text{minimize} \quad & 4x_1 + 3x_2 \\ \text{subject to} \quad & 1x_1 + 1x_2 \geq 2 \\ & 0x_1 + 1x_2 \geq 1 \\ & 1x_1 - 1x_2 \geq 3, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Its Dual LP

$$\begin{aligned} \text{maximize} \quad & 2p_1 + 1p_2 + 3p_3 \\ \text{subject to} \quad & 1p_1 + 0p_2 + 1p_3 \leq 4 \\ & 1p_1 + 1p_2 - 1p_3 \leq 3 \\ & p_1, p_2, p_3 \geq 0. \end{aligned}$$

- More generally:

Primal

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \text{maximize} \quad & b^T p \\ \text{subject to} \quad & A^T p \leq c \\ & p \geq 0. \end{aligned}$$

Recap: Lower bounds via duality

- For any LP, there is a corresponding dual LP.

Primal LP

$$\begin{aligned} \text{minimize} \quad & 4x_1 + 3x_2 \\ \text{subject to} \quad & 1x_1 + 1x_2 \geq 2 \\ & 0x_1 + 1x_2 \geq 1 \\ & 1x_1 - 1x_2 \geq 3, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Its Dual LP

$$\begin{aligned} \text{maximize} \quad & 2p_1 + 1p_2 + 3p_3 \\ \text{subject to} \quad & 1p_1 + 0p_2 + 1p_3 \leq 4 \\ & 1p_1 + 1p_2 - 1p_3 \leq 3 \\ & p_1, p_2, p_3 \geq 0. \end{aligned}$$

- More generally:

Primal

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \text{maximize} \quad & b^T p \\ \text{subject to} \quad & A^T p \leq c \\ & p \geq 0. \end{aligned}$$

Weak Duality Thm: If $\bar{x}$ is primal feasible and $\bar{p}$ is dual feasible:

(min. objective) $\quad c^T \bar{x} \geq b^T \bar{p} \quad$ (max. objective)

Recap: Lower bounds via duality

- For any LP, there is a corresponding dual LP.

Primal LP

$$\begin{aligned} \text{minimize} \quad & 4x_1 + 3x_2 \\ \text{subject to} \quad & 1x_1 + 1x_2 \geq 2 \\ & 0x_1 + 1x_2 \geq 1 \\ & 1x_1 - 1x_2 \geq 3, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Its Dual LP

$$\begin{aligned} \text{maximize} \quad & 2p_1 + 1p_2 + 3p_3 \\ \text{subject to} \quad & 1p_1 + 0p_2 + 1p_3 \leq 4 \\ & 1p_1 + 1p_2 - 1p_3 \leq 3 \\ & p_1, p_2, p_3 \geq 0. \end{aligned}$$

- More generally:

Primal

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

Dual

$$\begin{aligned} \text{maximize} \quad & b^T p \\ \text{subject to} \quad & A^T p \leq c \\ & p \geq 0. \end{aligned}$$

Strong Duality Thm: If x^* is primal optimal and p^* is dual optimal:

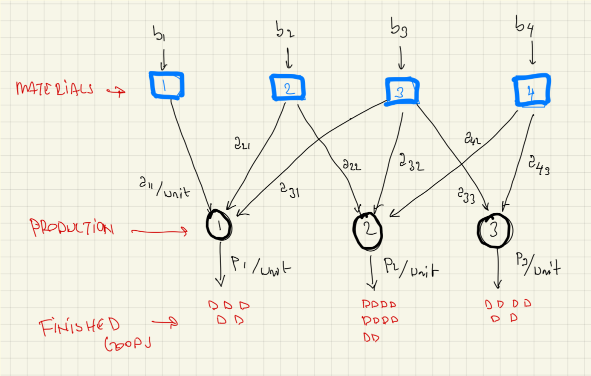
(min. objective) $\quad c^T x^* = b^T p^* \quad$ (max. objective)

LP notation example: Production planning

- A firm produces n different products using m different materials.
- Let $b_i \geq 0$, $i = 1, \ldots, m$, be the available amount of the ith material.
- The jth product, $j = 1, \ldots, n$, requires $a_{ij} \geq 0$ units of the ith material and results in a revenue of $p_j \geq 0$ per unit produced.
- Decide how much of each product to produce to maximize revenue.

LP notation example: Production planning

- A firm produces n different products using m different materials.
- Let $b_i \geq 0$, $i = 1, \ldots, m$, be the available amount of the ith material.
- The jth product, $j = 1, \ldots, n$, requires $a_{ij} \geq 0$ units of the ith material and results in a revenue of $p_j \geq 0$ per unit produced.
- Decide how much of each product to produce to maximize revenue.



LP notation example: Production planning

- A firm produces n different products using m different materials.
- Let $b_i \geq 0$, $i = 1, \ldots, m$, be the available amount of the ith material.
- The jth product, $j = 1, \ldots, n$, requires $a_{ij} \geq 0$ units of the ith material and results in a revenue of $p_j \geq 0$ per unit produced.
- Decide how much of each product to produce to maximize revenue.

- The decision variables x_j, denote how much jth product is produced.

LP notation example: Production planning

- A firm produces n different products using m different materials.
- Let $b_i \geq 0$, $i = 1, \ldots, m$, be the available amount of the ith material.
- The jth product, $j = 1, \ldots, n$, requires $a_{ij} \geq 0$ units of the ith material and results in a revenue of $p_j \geq 0$ per unit produced.
- Decide how much of each product to produce to maximize revenue.

- The decision variables x_j, denote how much jth product is produced.
- The problem can now be formulated as follows:

$$
\begin{array}{rl}
\text{maximize} & p_1 x_1 + p_2 x_2 + \cdots + p_n x_n \\
\text{subject to} & a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \leq b_1 \\
& a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \leq b_2 \\
& \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \\
& a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \leq b_m \\
& x_1, x_2, \cdots, x_n, \geq 0
\end{array}
$$

LP notation example: Production planning

- A firm produces n different products using m different raw materials.
- The decision variables x_j, denote how much jth product is produced.

$$
\begin{aligned}
\text{maximize} \quad & p_1 x_1 + \cdots + p_n x_n \\
\text{subject to} \quad & a_{i1} x_1 + \cdots + a_{in} x_n \le b_i \qquad i = 1, \ldots, m \\
& x_j \ge 0 \qquad j = 1, \ldots, n.
\end{aligned}
$$

LP problem using summations:

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} p_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1, \ldots m \\
& x_j \ge 0 \quad j = 1, \ldots, n.
\end{aligned}
$$

Written in matrix form:

$$
\begin{aligned}
\max \quad & p^T x \\
\text{s.t.} \quad & Ax \le b, \\
& x \ge 0
\end{aligned}
$$

$$p^T = [p_1,\ p_2,\ \cdots,\ p_n\ ]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \ b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(In this example we allow x variables to take fractional value, ex: $x_3 = 2.7$)

LP notation example: Production planning

- A firm produces n different products using m different raw materials.
- The decision variables x_j, denote how much jth product is produced.

$$\begin{array}{lll} \text{maximize} & p_1 x_1 + \cdots + p_n x_n & \\ \text{subject to} & a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, n. \end{array}$$

LP problem using summations:

$$\begin{array}{ll} \max & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, \ldots m \\ & x_j \geq 0 \quad j = 1, \ldots, n. \end{array}$$

Written in matrix form:

$$\begin{array}{ll} \max & p^T x \\ \text{s.t.} & Ax \leq b, \\ & x \geq 0 \end{array}$$

$$p^T = [p_1, \ p_2, \ \cdots \ , p_n \ ]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \ b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(In this example we allow x variables to take fractional value, ex: $x_3 = 2.7$)

LP notation example: Production planning

- A firm produces n different products using m different raw materials.
- The decision variables x_j, denote how much jth product is produced.

$$\begin{aligned} \text{maximize} \quad & p_1x_1 + \cdots + p_nx_n & \\ \text{subject to} \quad & a_{i1}x_1 + \cdots + a_{in}x_n \le b_i & i = 1, \ldots, m \\ & x_j \ge 0 & j = 1, \ldots, n. \end{aligned}$$

LP problem using summations:

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1, \ldots m \\ & x_j \ge 0 \quad j = 1, \ldots, n. \end{aligned}$$

Written in matrix form:

$$\begin{aligned} \max \quad & p^T x \\ \text{s.t.} \quad & Ax \le b, \\ & x \ge 0 \end{aligned}$$

$$p^T = [p_1,\ p_2,\ \cdots,\ p_n\ ]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(In this example we allow x variables to take fractional value, ex: $x_3 = 2.7$)

LP notation example: Production planning

- A firm produces n different products using m different raw materials.
- The decision variables x_j, denote how much jth product is produced.

$$\begin{array}{lll} \text{maximize} & p_1x_1 + \cdots + p_nx_n & \\ \text{subject to} & a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, n. \end{array}$$

LP problem using summations:

$$\begin{array}{ll} \max & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad i = 1, \ldots m \\ & x_j \geq 0 \quad j = 1, \ldots, n. \end{array}$$

Written in matrix form:

$$\begin{array}{rl} \max & p^T x \\ \text{s.t.} & Ax \leq b, \\ & x \geq 0 \end{array}$$

$$p^T = [p_1,\ p_2,\ \cdots,\ p_n\ ]$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

(In this example we allow x variables to take fractional value, ex: $x_3 = 2.7$)

Integer Programming

IP Example: Labor scheduling

- A hospital wants to make a weekly night shift (10pm-6am) schedule for its nurses.
- The demand for nurses for the night shift on day j is an integer d_j, for all $j \in \{Su, Mo, Tu, We, Th, Fr, Sa\}$.
- Every nurse works 5 consecutive days and takes the next 2 days off.
- Find the minimum number of nurses the hospital needs to hire.

Question: What are the decision variables?

IP Example: Labor scheduling

What are the decision variables?

- We could try using a decision variable y_j equal to the number of nurses that work on day j.
- However, with this definition we would not be able to capture the constraint that every nurse works 5 days in a row.
- We need to know the starting day of a nurse to model the problem correctly.
- We define x_j as the number of nurses starting their week on day $j \in \{Mo, Tu, We, Th, Fr, Sa, Su\}$.
- We can now write a constraint for every day of the week to make sure that the demand is satisfied.

IP Example: Labor scheduling

What are the decision variables?

- We could try using a decision variable y_j equal to the number of nurses that work on day j.
- However, with this definition we would not be able to capture the constraint that every nurse works 5 days in a row.
- We need to know the starting day of a nurse to model the problem correctly.
- We define x_j as the number of nurses starting their week on day $j \in \{Mo, Tu, We, Th, Fr, Sa, Su\}$.
- We can now write a constraint for every day of the week to make sure that the demand is satisfied.

IP Example: Labor scheduling

We then have the following problem formulation:

$$
\begin{array}{rllllllll}
\text{minimize} & x_{Su} & +x_{Mo}+x_{Tu} & +x_{We}+x_{Th} & +x_{Fr}+x_{Sa} & \\
\\
\text{subject to} & x_{Su} & & +x_{We}+x_{Th} & +x_{Fr}+x_{Sa} & \geq d_{Su} \\
& x_{Su} & +x_{Mo} & +x_{Th} & +x_{Fr}+x_{Sa} & \geq d_{Mo} \\
& x_{Su} & +x_{Mo}+x_{Tu} & & +x_{Fr}+x_{Sa} & \geq d_{Tu} \\
& x_{Su} & +x_{Mo}+x_{Tu} & +x_{We} & +x_{Sa} & \geq d_{We} \\
& x_{Su} & +x_{Mo}+x_{Tu} & +x_{We}+x_{Th} & & \geq d_{Th} \\
& & +x_{Mo}+x_{Tu} & +x_{We}+x_{Th} & +x_{Fr} & \geq d_{Fr} \\
& & +x_{Tu} & +x_{We}+x_{Th} & +x_{Fr}+x_{Sa} & \geq d_{Sa}
\end{array}
$$

$$x_j \geq 0 \text{ and } \underbrace{x_j \in \mathbb{Z},}_{integer} \quad j \in \{Mo,Tu,We,Th,Fr,Sa,Su\}$$

- This would be an LP except for the integrality constraints.

IP Example: Labor scheduling

We then have the following problem formulation:

$$
\begin{array}{lllllllll}
\text{minimize} & x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & & \\
\\
\text{subject to} & x_{Su} & & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Su} \\
 & x_{Su} & +x_{Mo} & \phantom{+x_{We}} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Mo} \\
 & x_{Su} & +x_{Mo} + x_{Tu} & & +x_{Fr} + x_{Sa} & \geq d_{Tu} \\
 & x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} & \phantom{+x_{Fr}} + x_{Sa} & \geq d_{We} \\
 & x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & & \geq d_{Th} \\
 & & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} & \geq d_{Fr} \\
 & & \phantom{+x_{Mo}} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Sa}
\end{array}
$$

$$x_j \geq 0 \;\; \text{and} \;\; \underbrace{x_j \in \mathbb{Z},}_{integer} \quad j \in \{Mo, Tu, We, Th, Fr, Sa, Su\}$$

- This would be an LP except for the integrality constraints.

Question:

What symbols do we use for numbers?

$\mathbb{R}$ Real numbers

$\mathbb{C}$ Complex numbers

$\mathbb{Q}$ Fractional numbers (Quotients)

$\mathbb{N}$ Natural numbers

$\mathbb{Z}$ Integers

Question: Why do we use $\mathbb{Z}$ for integers?

Answer: The use of the letter Z to denote the set of integers comes from the German word Zahlen ("numbers") and has been attributed to David Hilbert. [from Wikipedia]

Question:

What symbols do we use for numbers?

$\mathbb{R}$ Real numbers

$\mathbb{C}$ Complex numbers

$\mathbb{Q}$ Fractional numbers (Quotients)

$\mathbb{N}$ Natural numbers

$\mathbb{Z}$ Integers

Question: Why do we use $\mathbb{Z}$ for integers?

Answer: The use of the letter Z to denote the set of integers comes from the German word Zahlen ("numbers") and has been attributed to David Hilbert. [from Wikipedia]

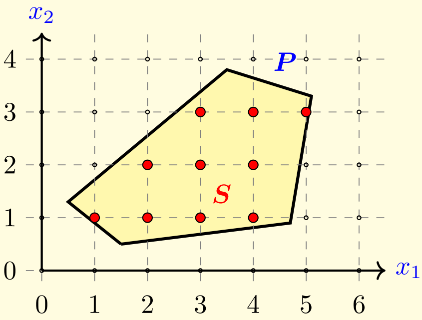
What is an Integer (Linear) Program?

A **pure integer program** (IP):

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \\ & x \ \text{integral (i.e., } x \in \mathbb{Z}^n\text{)}. \end{aligned}$$

Feasible set: $S = \{x \in \mathbb{Z}^n : Ax \geq b, x \geq 0\} = \textit{Polyhedron} \cap \mathbb{Z}^n$.

When $n = 2$



Remember the labor scheduling problem

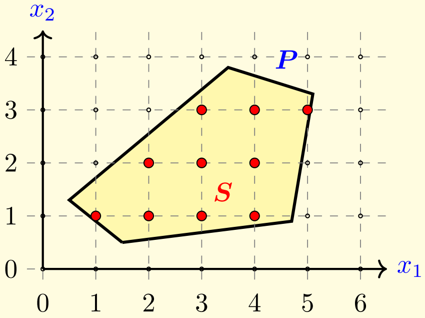
- If we ignore ("relax") the integrality constraints, we obtain the so-called LP relaxation of this problem.

$$
\begin{array}{rllllllll}
\text{minimize} & x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \\
\text{subject to} & x_{Su} & & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Su} \\
& x_{Su} & +x_{Mo} & \quad\;\; + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Mo} \\
& x_{Su} & +x_{Mo} + x_{Tu} & & +x_{Fr} + x_{Sa} & \geq d_{Tu} \\
& x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} & \quad\;\; + x_{Sa} & \geq d_{We} \\
& x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & & \geq d_{Th} \\
& & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} & \geq d_{Fr} \\
& & \quad\;\; + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Sa}
\end{array}
$$

$$x_j \geq 0 \;\; \cancel{\text{and } x_j \in \mathbb{Z}}, \qquad j \in \{Su, Mo, Tu, We, Th, Fr, Sa\}.$$

- All feasible solutions to the integer program are also feasible to the LP relaxation.

LP relaxations – minimization objective



- P is the feasible set for the LP relaxation (ignore integrality)
- S is the feasible set for the integer program:

$$S = P \cap \mathbb{Z}^n$$

Question: How would

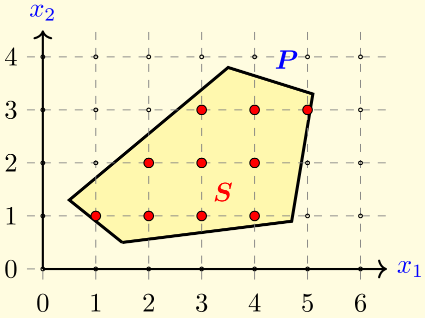
$$z_{IP} = \min c^T x \quad \text{s.t. } x \in S$$

compare to

$$z_{LP} = \min c^T x \quad \text{s.t. } x \in P \text{ ?}$$

Answer: $z_{IP} \geq z_{LP}$ $\leftarrow$ because $S \subseteq P$ and we are minimizing.

LP relaxations – minimization objective



- P is the feasible set for the LP relaxation (ignore integrality)
- S is the feasible set for the integer program:

$$S = P \cap \mathbb{Z}^n$$

Question: How would

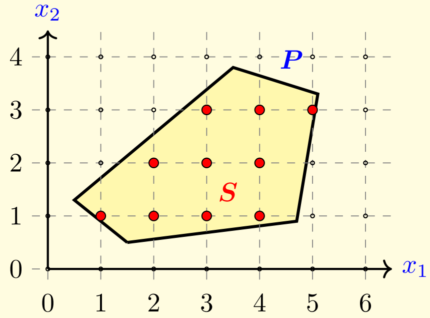
$$z_{IP} = \min c^T x \quad \text{s. t. } x \in S$$

compare to

$$z_{LP} = \min c^T x \quad \text{s. t. } x \in P \text{ ?}$$

Answer: $z_{IP} \geq z_{LP}$ $\leftarrow$ because $S \subseteq P$ and we are minimizing.

LP relaxations – **maximization** objective



- P is the feasible set for the LP relaxation
- S is the feasible set for the integer program:

$$S = P \cap \mathbb{Z}^n$$

Question: How would

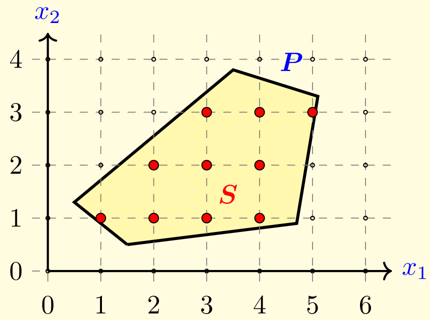
$$z_{IP} = \mathbf{max}\ c^T x \quad \text{s.t. } x \in S$$

compare to

$$z_{LP} = \mathbf{max}\ c^T x \quad \text{s.t. } x \in P\ ?$$

Answer: $z_{IP} \leq z_{LP}$ $\leftarrow$ because $S \subseteq P$ and we are maximizing.

LP relaxations – **maximization** objective



- P is the feasible set for the LP relaxation
- S is the feasible set for the integer program:

$$S = P \cap \mathbb{Z}^n$$

Question: How would

$$z_{IP} = \mathbf{max}\ c^T x \quad \text{s.t. } x \in S$$

compare to

$$z_{LP} = \mathbf{max}\ c^T x \quad \text{s.t. } x \in P\ ?$$

Answer: $z_{IP} \leq z_{LP}$ $\leftarrow$ because $S \subseteq P$ and we are maximizing.

Relaxations – lower bounds (minimization objective)

- More generally, consider the following 2 optimization problems:

$$z_S = \min f(x) \quad \text{s. t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s. t. } x \in B \qquad \text{(big set)}$$

where we optimize the same objective function over 2 different feasible sets S and B such that

$$S \subseteq B.$$

(B is called a **relaxation** of S as it contains more solutions.)

- Assume that both problems are feasible and have an optimal solution.

Claim: $z_S \geq z_B$.

Proof. Let $x^S \in S$ be an optimal solution for the first problem.

As $S \subseteq B$ we know that that $x^S \in B$. Therefore $f(x^S)$ is an upper bound on z_B:

$$z_S = \underbrace{f(x^S) \geq \min_{x \in B} f(x)}_{\text{because } x^S \in B} = z_B$$

- In the LP/IP setting this means that the opt. LP value gives a lower bound on the opt IP value (for minimization).

Relaxations – lower bounds (minimization objective)

- More generally, consider the following 2 optimization problems:

$$z_S = \min f(x) \quad \text{s.t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s.t. } x \in B \qquad \text{(big set)}$$

  where we optimize the same objective function over 2 different feasible sets S and B such that

$$S \subseteq B.$$

  (B is called a **relaxation** of S as it contains more solutions.)

- Assume that both problems are feasible and have an optimal solution.

  Claim: $z_S \geq z_B$.

  Proof. Let $x^S \in S$ be an optimal solution for the first problem.

  As $S \subseteq B$ we know that that $x^S \in B$. Therefore $f(x^S)$ is an upper bound on z_B:

$$z_S \; = \; \underbrace{f(x^S) \; \geq \min_{x \in B} f(x)}_{\text{because } x^S \in B} \; = z_B$$

- In the LP/IP setting this means that the opt. LP value gives a lower bound on the opt IP value (for minimization).

Relaxations – lower bounds (minimization objective)

- More generally, consider the following 2 optimization problems:

$$z_S = \min f(x) \quad \text{s.t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s.t. } x \in B \qquad \text{(big set)}$$

  where we optimize the same objective function over 2 different feasible sets S and B such that

$$S \subseteq B.$$

  (B is called a **relaxation** of S as it contains more solutions.)

- Assume that both problems are feasible and have an optimal solution.

  Claim: $z_S \geq z_B$.

  Proof. Let $x^S \in S$ be an optimal solution for the first problem.

  As $S \subseteq B$ we know that that $x^S \in B$. Therefore $f(x^S)$ is an upper bound on z_B:

$$z_S \; = \; \underbrace{f(x^S) \; \geq \min_{x \in B} f(x)}_{\text{because } x^S \in B} \; = z_B$$

- In the LP/IP setting this means that the opt. LP value gives a lower bound on the opt IP value (for minimization).

Relaxations – upper bounds (minimization objective)

- Once again, let

$$z_S = \min f(x) \quad \text{s.t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s.t. } x \in B \qquad \text{(big set)}$$

where B is a **relaxation** of S: $\quad S \subseteq B$.

- Assume that both problems are feasible and have an optimal solution.

Claim: If the optimal solution $x^B \in B$ for the second problem is also contained in S, then x^B is optimal for S and $z_S = z_B$.

Proof. We know that $z_S \geq z_B$ from previous slide. As $x^B \in S$,

$$z_B = \underbrace{f(x^B) \geq \min_{x \in S} f(x)}_{\text{because } x^B \in S} = z_S$$

and therefore $z_S \leq z_B \implies z_S = z_B$

- In the LP/IP setting this means that if the opt. solution to the LP happens to be integral, you are in luck!

Relaxations – upper bounds (minimization objective)

- Once again, let

$$z_S = \min f(x) \quad \text{s.t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s.t. } x \in B \qquad \text{(big set)}$$

where B is a **relaxation** of S: $\quad S \subseteq B$.

- Assume that both problems are feasible and have an optimal solution.

Claim: If the optimal solution $x^B \in B$ for the second problem is also contained in S, then x^B is optimal for S and $z_S = z_B$.

Proof. We know that $z_S \geq z_B$ from previous slide. As $x^B \in S$,

$$z_B = \underbrace{f(x^B) \geq \min_{x \in S} f(x)}_{\text{because } x^B \in S} = z_S$$

and therefore $z_S \leq z_B \implies z_S = z_B$

- In the LP/IP setting this means that if the opt. solution to the LP happens to be integral, you are in luck!

Relaxations – upper bounds (minimization objective)

- Once again, let

$$z_S = \min f(x) \quad \text{s. t. } x \in S \qquad \text{(small set)}$$

$$z_B = \min f(x) \quad \text{s. t. } x \in B \qquad \text{(big set)}$$

where B is a **relaxation** of S: $\qquad S \subseteq B$.

- Assume that both problems are feasible and have an optimal solution.

Claim: If the optimal solution $x^B \in B$ for the second problem is also contained in S, then x^B is optimal for S and $z_S = z_B$.

Proof. We know that $z_S \geq z_B$ from previous slide. As $x^B \in S$,

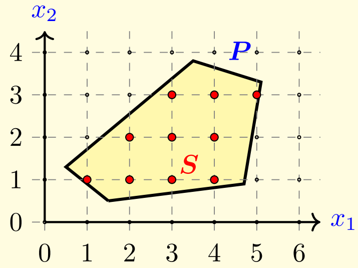
$$z_B = \underbrace{f(x^B) \geq \min_{x \in S} f(x)}_{\text{because } x^B \in S} = z_S$$

and therefore $z_S \leq z_B \quad \Longrightarrow \quad z_S = z_B$

- In the LP/IP setting this means that if the opt. solution to the LP happens to be integral, you are in luck!

LP vs. IP (minimization objective)

Integer program v.s. its linear programming relaxation:



$$z^{IP} = \min c^T x \text{ s.t. } x \in S \; (= P \cap \mathbb{Z}^n)$$

$$z^{LP} = \min c^T x \text{ s.t. } x \in P$$

where $S \subseteq P$.

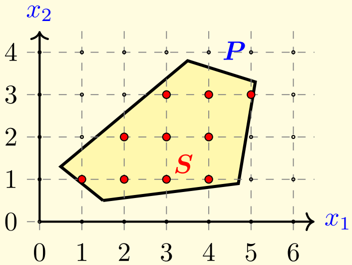
- As $S \subseteq P$, we always have $z^{IP} \geq z^{LP}$.
- If the optimal solution to the LP happens to be integral, then it is also an optimal solution to the original problem.
- If the optimal solution to the LP non-integral coordinates, then we have to do more work to solve the IP.

Question 1: Can the IP be feasible when the LP is infeasible?

Question 2: Can the IP be infeasible when the LP is feasible?

LP vs. IP (minimization objective)

Integer program v.s. its linear programming relaxation:



$$z^{IP} = \min c^T x \text{ s.t. } x \in S \ (= P \cap \mathbb{Z}^n)$$

$$z^{LP} = \min c^T x \text{ s.t. } x \in P$$

where $S \subseteq P$.

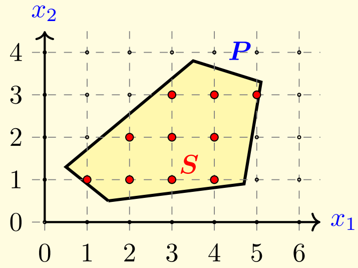
- As $S \subseteq P$, we always have $z^{IP} \geq z^{LP}$.
- If the optimal solution to the LP happens to be integral, then it is also an optimal solution to the original problem.
- If the optimal solution to the LP non-integral coordinates, then we have to do more work to solve the IP.

Question 1: Can the IP be feasible when the LP is infeasible?

Question 2: Can the IP be infeasible when the LP is feasible?

LP vs. IP (minimization objective)

Integer program v.s. its linear programming relaxation:



$$z^{IP} = \min c^T x \text{ s.t. } x \in S \ (= P \cap \mathbb{Z}^n)$$

$$z^{LP} = \min c^T x \text{ s.t. } x \in P$$

where $S \subseteq P$.

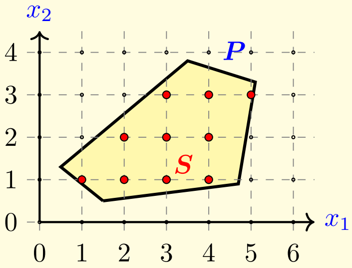
- As $S \subseteq P$, we always have $z^{IP} \geq z^{LP}$.
- If the optimal solution to the LP happens to be integral, then it is also an optimal solution to the original problem.
- If the optimal solution to the LP non-integral coordinates, then we have to do more work to solve the IP.

Question 1: Can the IP be feasible when the LP is infeasible?

Question 2: Can the IP be infeasible when the LP is feasible?

LP vs. IP (minimization objective)

Integer program v.s. its linear programming relaxation:



$$z^{IP} = \min c^T x \text{ s.t. } x \in S \ (= P \cap \mathbb{Z}^n)$$

$$z^{LP} = \min c^T x \text{ s.t. } x \in P$$

where $S \subseteq P$.

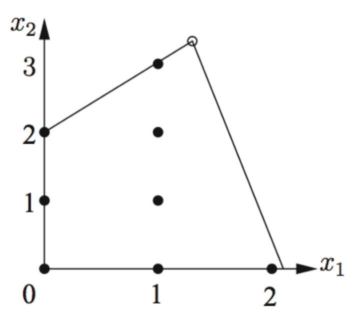
- As $S \subseteq P$, we always have $z^{IP} \geq z^{LP}$.
- If the optimal solution to the LP happens to be integral, then it is also an optimal solution to the original problem.
- If the optimal solution to the LP non-integral coordinates, then we have to do more work to solve the IP.

Question 1: Can the IP be feasible when the LP is infeasible?

Question 2: Can the IP be infeasible when the LP is feasible?

Example: Rounding LP solutions

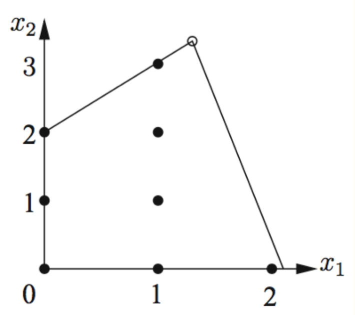
$$\begin{aligned} \max \quad & 5.5x_1 + 2.1x_2 \\ \text{s.t.} \quad & -x_1 + x_2 \le 2 \\ & 8x_1 + 2x_2 \le 17 \\ & x_1, x_2 \ge 0 \\ & x_1, x_2 \text{ integer} \end{aligned}$$



- The optimal IP solution is $(1, 3)$ with objective value 11.8.

Example: Rounding LP solutions

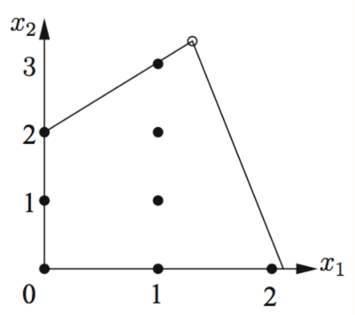
$$\begin{aligned} \max \quad & 5.5x_1 + 2.1x_2 \\ \text{s. t.} \quad & -x_1 + x_2 \leq 2 \\ & 8x_1 + 2x_2 \leq 17 \\ & x_1, x_2 \geq 0 \\ & \cancel{x_1, x_2 \text{ integer}} \end{aligned}$$



- The optimal IP solution is $(1, 3)$ with objective value 11.8.
- The optimal solution of the LP relaxation is $(1.3, 3.3)$, with objective value 14.08.
- Here the LP solution gives an upper bound on the optimal IP value as we are maximizing.

Example: Rounding LP solutions

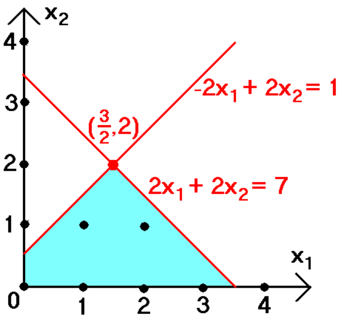
$$
\begin{aligned}
\max \quad & 5.5x_1 + 2.1x_2 \\
\text{s. t.} \quad & -x_1 + x_2 \leq 2 \\
& 8x_1 + 2x_2 \leq 17 \\
& x_1, x_2 \geq 0 \\
& \cancel{x_1, x_2 \text{ integer}}
\end{aligned}
$$



- The optimal IP solution is $(1, 3)$ with objective value 11.8.
- The optimal solution of the LP relaxation is $(1.3, 3.3)$, with objective value 14.08.
- Here the LP solution gives an upper bound on the optimal IP value as we are maximizing.
- It also looks like we can round $(1.3, 3.3)$ to the optimum solution $(1, 3)$, but. . .

Another example: Rounding LP solutions

$$
\begin{aligned}
\max \quad & 15x_2 \\
\text{s.t.} \quad & -2x_1 + 2x_2 \le 1 \\
& 2x_1 + 2x_2 \le 7 \\
& x_1, x_2 \ge 0 \\
& \cancel{x_1, x_2 \text{ integer}}
\end{aligned}
$$

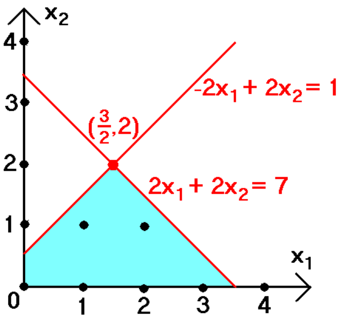


- Optimal IP solutions ares $(1, 1)$ and $(2, 1)$ with obj. value 15.
- The opt. sol. of the LP relaxation is $(3/2, 2)$, with obj. value 30.
- Rounding the fractional component either up or down leads to an infeasible solution.

(LP solution still gives an upper bound on the optimal IP value)

Another example: Rounding LP solutions

$$\begin{aligned} \max \quad & 15x_2 \\ \text{s.t.} \quad & -2x_1 + 2x_2 \leq 1 \\ & 2x_1 + 2x_2 \leq 7 \\ & x_1, x_2 \geq 0 \\ & \cancel{x_1, x_2 \text{ integer}} \end{aligned}$$



- Optimal IP solutions ares $(1, 1)$ and $(2, 1)$ with obj. value 15.
- The opt. sol. of the LP relaxation is $(3/2, 2)$, with obj. value 30.
- Rounding the fractional component either up or down leads to an infeasible solution.

  (LP solution still gives an upper bound on the optimal IP value)

One more example: Nurse scheduling

$$
\begin{array}{llllllllll}
\min & x_{Su} & +x_{Mo} & +\, x_{Tu} & +x_{We} & +\, x_{Th} & +x_{Fr} & +\, x_{Sa} & & \\
\text{s.t.} & x_{Su} & & & +x_{We} & +\, x_{Th} & +x_{Fr} & +\, x_{Sa} & \geq d_{Su} & = 2 \\
& x_{Su} & +x_{Mo} & & & +\, x_{Th} & +x_{Fr} & +\, x_{Sa} & \geq d_{Mo} & = 2 \\
& x_{Su} & +x_{Mo} & +\, x_{Tu} & & & +x_{Fr} & +\, x_{Sa} & \geq d_{Tu} & = 2 \\
& x_{Su} & +x_{Mo} & +\, x_{Tu} & +x_{We} & & & +\, x_{Sa} & \geq d_{We} & = 2 \\
& x_{Su} & +x_{Mo} & +\, x_{Tu} & +x_{We} & +\, x_{Th} & & & \geq d_{Th} & = 2 \\
& & +x_{Mo} & +\, x_{Tu} & +x_{We} & +\, x_{Th} & +x_{Fr} & & \geq d_{Fr} & = 2 \\
& & & +\, x_{Tu} & +x_{We} & +\, x_{Th} & +x_{Fr} & +\, x_{Sa} & \geq d_{Sa} & = 2
\end{array}
$$

$$x_j \geq 0 \ \text{ and } \ \underbrace{x_j \in \mathbb{Z}}_{integer}, \quad j \in \{Su, Mo, Tu, We, Th, Fr, Sa\}$$

- Optimal LP solution is $x_j = 2/5 \ \forall j$ with cost $14/5 = 2.8$.
- Round-up solution $x_j = \lceil 2/5 \rceil = 1 \ \forall j$ is always feasible for IP (why?) with cost 7.
- However $x_{Su} = x_{Tu} = x_{Th} = 1$ (remaining $x_j = 0$) is an optimal IP solution with cost 3!

One more example: Nurse scheduling

$$
\begin{array}{llllllllll}
\min & x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \\
\\
\text{s.t.} & x_{Su} & & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Su} = 2 \\
& x_{Su} & +x_{Mo} & + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Mo} = 2 \\
& x_{Su} & +x_{Mo} + x_{Tu} & & +x_{Fr} + x_{Sa} & \geq d_{Tu} = 2 \\
& x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} & + x_{Sa} & \geq d_{We} = 2 \\
& x_{Su} & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & & \geq d_{Th} = 2 \\
& & +x_{Mo} + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} & \geq d_{Fr} = 2 \\
& & + x_{Tu} & +x_{We} + x_{Th} & +x_{Fr} + x_{Sa} & \geq d_{Sa} = 2
\end{array}
$$

$$x_j \geq 0 \text{ and } \underbrace{x_j \in \mathbb{Z}}_{integer}, \quad j \in \{Su, Mo, Tu, We, Th, Fr, Sa\}$$

- Optimal LP solution is $x_j = 2/5 \ \forall j$ with cost $14/5 = 2.8$.
- Round-up solution $x_j = \lceil 2/5 \rceil = 1 \ \forall j$ is always feasible for IP (why?) with cost 7.
- However $x_{Su} = x_{Tu} = x_{Th} = 1$ (remaining $x_j = 0$) is an optimal IP solution with cost 3!

Can we just round the fractional LP solution?

- We cannot always round the solution of the LP relaxation to the optimal solution of the IP:

  1. Rounding to it a feasible integer solution might be impossible
  2. Moreover, the IP can be infeasible even though the LP is feasible

     Example: $P = \{x \in \mathbb{R}^2 : 0.6 \geq x_1 \geq 0.2,\ 0.75 \geq x_2 \geq 0.41\}$
  3. Even if rounding to an integer solution is possible, the opt. IP sol. can be arbitrarily far (different) from the opt. LP sol.
  4. Even if there is an optimal integer solution "nearby", finding it might require checking **exponentially many** rounding options (up/down) as we might need to consider:

     $$\lfloor x_j \rfloor \quad \text{and} \quad \lceil x_j \rceil \qquad \text{for all fractional components } x_j$$

     There can be up to 2^n possibilities.

Can we just round the fractional LP solution?

- We cannot always round the solution of the LP relaxation to the optimal solution of the IP:

  1. Rounding to it a feasible integer solution might be impossible
  2. Moreover, the IP can be infeasible even though the LP is feasible

     Example: $P = \{x \in \mathbb{R}^2 : 0.6 \geq x_1 \geq 0.2,\ 0.75 \geq x_2 \geq 0.41\}$
  3. Even if rounding to an integer solution is possible, the opt. IP sol. can be arbitrarily far (different) from the opt. LP sol.
  4. Even if there is an optimal integer solution "nearby", finding it might require checking **exponentially many** rounding options (up/down) as we might need to consider:

     $$\lfloor x_j \rfloor \quad \text{and} \quad \lceil x_j \rceil \qquad \text{for all fractional components } x_j$$

     There can be up to 2^n possibilities.

Solving IPs: Branch and Bound

How do we solve an IP?

$$z^{IP} = \min\ c^T x \quad \text{s.t.} \quad Ax \geq b,\ x \geq 0,\ x \in \mathbb{Z}^n \qquad (IP)$$

- Relaxing integrality gives an LP problem (easy to solve)
- Solving LP gives an lower bound z^{LP} (for minimization)
- The optimal solution x^{LP}(to LP may have fractional components, say, $x_j^{LP} \notin \mathbb{Z}$, (eg., $x_1^{LP} = 3.31$) $\implies x^{LP}$ not feasible for (IP).

How do we solve an IP?

$$z^{IP} = \min\ c^T x \quad \text{s.t.} \quad Ax \geq b,\ x \geq 0,\ x \in \mathbb{Z}^n \qquad (IP)$$

- Relaxing integrality gives an LP problem (easy to solve)
- Solving LP gives an lower bound z^{LP} (for minimization)
- The optimal solution x^{LP}(to LP may have fractional components, say, $x_j^{LP} \notin \mathbb{Z}$, (eg., $x_1^{LP} = 3.31$) $\implies x^{LP}$ not feasible for (IP).

Idea: Divide the solution set, and the IP into two new **subproblems** IP_1 and IP_2, with additional constraints

$$\text{IP}_1 : x_1 \leq \lfloor x_1^{LP} \rfloor = 3 \qquad \text{IP}_2 : x_1 \geq \lceil x_1^{LP} \rceil = 4$$

Instead of solving IP, we not solve both IP_1 and IP_2.

How do we solve an IP?

$$z^{IP} = \min\ c^T x \quad \text{s.t.} \quad Ax \geq b,\ x \geq 0,\ x \in \mathbb{Z}^n \qquad (IP)$$

- Relaxing integrality gives an LP problem (easy to solve)
- Solving LP gives an lower bound z^{LP} (for minimization)
- The optimal solution x^{LP}(to LP may have fractional components, say, $x_j^{LP} \notin \mathbb{Z}$, (eg., $x_1^{LP} = 3.31$) $\implies x^{LP}$ not feasible for (IP).

Idea: Divide the solution set, and the IP into two new **subproblems** IP_1 and IP_2, with additional constraints

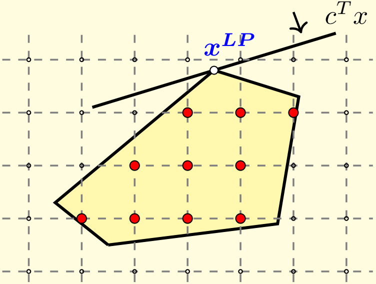
$$\text{IP}_1 : x_1 \leq \lfloor x_1^{LP} \rfloor = 3 \qquad \text{IP}_2 : x_1 \geq \lceil x_1^{LP} \rceil = 4$$

Instead of solving IP, we not solve both IP_1 and IP_2.

Note: **Any** feasible (optimal) solution of IP is either in IP_1 or in IP_2

Note: **No** feasible solution of LP_1 or LP_2 can have $x_1 = 3.31$

Partitioning step for integer programs



(IP)

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & \underbrace{Ax \geq b,\ x \geq 0}_{P=\{x\in\mathbb{R}^n : Ax\geq b, x\geq 0\}} & x \in \mathbb{Z}^n \end{array}$$

- Let $\boldsymbol{x^{LP}}$ be the opt. sol. to the LP relaxation.

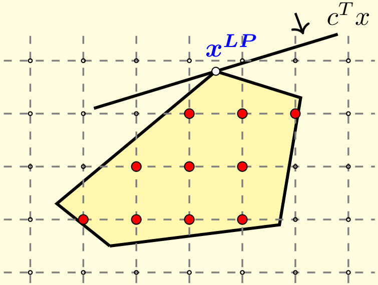
(IP$_1$) $\min\ c^T x\ \text{ s.t. }\ x \in P_1 \cap \mathbb{Z}^n$

$P_1 = \{x : Ax \geq b, x_1 \leq \lfloor x_1^{LP} \rfloor, x \geq 0\}$

(IP$_2$) $\min\ c^T x\ \text{ s.t. }\ x \in P_2 \cap \mathbb{Z}^n$

$P_2 = \{x : Ax \geq b, x_1 \geq \lceil x_1^{LP} \rceil, x \geq 0\}$

Partitioning step for integer programs

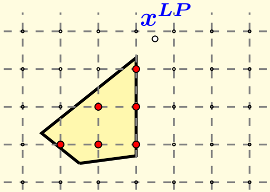


(IP)

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \underbrace{Ax \geq b,\ x \geq 0}_{P=\{x \in \mathbb{R}^n : Ax \geq b, x \geq 0\}} \quad x \in \mathbb{Z}^n \end{array}$$

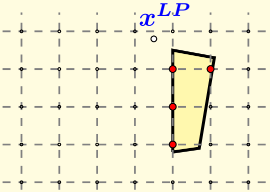
- Let x^{LP} be the opt. sol. to the LP relaxation.

(IP$_1$) $\min\ c^T x\ \text{s.t.}\ x \in P_1 \cap \mathbb{Z}^n$



$P_1 = \{x : Ax \geq b, x_1 \leq \lfloor x_1^{LP} \rfloor, x \geq 0\}$

(IP$_2$) $\min\ c^T x\ \text{s.t.}\ x \in P_2 \cap \mathbb{Z}^n$



$P_2 = \{x : Ax \geq b, x_1 \geq \lceil x_1^{LP} \rceil, x \geq 0\}$

Partitioning an IP problem into 2 subproblems

- Solve the LP relaxation:

$$z^{LP} \; = \; \min\{c^T x \; : \; x \in P\} = c^T x^{LP}$$

- Divide:

$$P_1 = P \cap \{x_1 \leq \lfloor x_1^{LP} \rfloor\} \text{ and } P_2 = P \cap \{x_1 \geq \lceil x_1^{LP} \rceil\}$$

- We have:

$$P \cap \mathbb{Z}^n = (P_1 \cap \mathbb{Z}^n) \cup (P_2 \cap \mathbb{Z}^n) \quad \longleftarrow \text{ integer points}$$

- Whereas:

$$P \supsetneq P_1 \cup P_2 \quad \longleftarrow \text{ fractional points}$$

Integer program (IP):

LP relaxation of IP:

$$z^{IP} = \min\{c^T x : x \in P \cap \mathbb{Z}^n\} \quad \geq \quad z^{LP} = \min\{c^T x : x \in P\}$$

$$z^{IP1} = \min\{c^T x : x \in P_1 \cap \mathbb{Z}^n\} \quad \geq \quad z^{LP1} = \min\{c^T x : x \in P_1\}$$

$$z^{IP2} = \min\{c^T x : x \in P_2 \cap \mathbb{Z}^n\} \quad \geq \quad z^{LP2} = \min\{c^T x : x \in P_2\}$$

$$z^{IP} \; = \; \min\{z^{IP1}, z^{IP2}\} \quad \geq \quad \min\{z^{LP1}, z^{LP2}\} \; \geq \; z^{LP}$$

Partitioning an IP problem into 2 subproblems

- Solve the LP relaxation:

$$z^{LP} \;=\; \min\{c^T x \;:\; x \in P\} = c^T x^{LP}$$

- Divide:

$$P_1 = P \cap \{x_1 \leq \lfloor x_1^{LP} \rfloor\} \text{ and } P_2 = P \cap \{x_1 \geq \lceil x_1^{LP} \rceil\}$$

- We have:

$$P \cap \mathbb{Z}^n = (P_1 \cap \mathbb{Z}^n) \cup (P_2 \cap \mathbb{Z}^n) \quad \longleftarrow \text{ integer points}$$

- Whereas:

$$P \supsetneq P_1 \cup P_2 \quad \longleftarrow \text{ fractional points}$$

Integer program (IP):		LP relaxation of IP:
$z^{IP} = \min\{c^T x : x \in P \cap \mathbb{Z}^n\}$	$\geq$	$z^{LP} = \min\{c^T x : x \in P\}$
$z^{IP1} = \min\{c^T x : x \in P_1 \cap \mathbb{Z}^n\}$	$\geq$	$z^{LP1} = \min\{c^T x : x \in P_1\}$
$z^{IP2} = \min\{c^T x : x \in P_2 \cap \mathbb{Z}^n\}$	$\geq$	$z^{LP2} = \min\{c^T x : x \in P_2\}$
$z^{IP} \;=\; \min\{z^{IP1}, z^{IP2}\}$	$\geq$	$\min\{z^{LP1}, z^{LP2}\} \;\geq\; z^{LP}$

More generally: Divide and conquer principle

Consider

$$z^* = \min\{f(x) : x \in S\}$$

If a collection of disjoint sets $\{S_1, S_2, \ldots, S_k\}$ satisfy

$$S = S_1 \cup S_2 \cup \ldots \cup S_k$$

then $\{S_1, S_2, \ldots, S_k\}$ is called a partition of S

Let $z^i = \min\{f(x) : x \in S_i\}$, and $z^i \geq z^i_{LB}$ $\longleftarrow$ a lower bound

Observation 1:

$$z^* = \min\{z^1, z^2, z^3, \ldots, z^k\}$$

Observation 2:

$$z^* \leq \min\{z^1, z^3, z^8, \ldots\}$$ $\longleftarrow$ some z^is are missing here

Observation 3:

$$z^* \geq \min\{z^1_{LB}, z^2_{LB}, \ldots, z^k_{LB}\}.$$

(In branch and bound, we dynamically decide how to partition of S.)

More generally: Divide and conquer principle

Consider

$$z^* = \min \{ f(x) : x \in S \}$$

If a collection of disjoint sets $\{S_1, S_2, \ldots, S_k\}$ satisfy

$$S = S_1 \cup S_2 \cup \ldots \cup S_k$$

then $\{S_1, S_2, \ldots, S_k\}$ is called a partition of S

Let $z^i = \min\{f(x) : x \in S_i\}$, and $z^i \geq z^i_{LB}$ $\longleftarrow$ a lower bound

Observation 1:

$$z^* = \min\{z^1, z^2, z^3, \ldots, z^k\}$$

Observation 2:

$$z^* \leq \min\{z^1, z^3, z^8, \ldots\}$$ $\longleftarrow$ some z^is are missing here

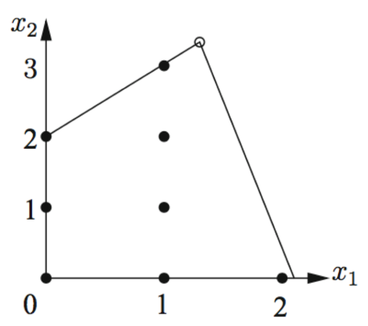
Observation 3:

$$z^* \geq \min\{z^1_{LB}, z^2_{LB}, \ldots, z^k_{LB}\}.$$

(In branch and bound, we dynamically decide how to partition of S.)

Example

$$\begin{array}{lll} \text{(IP)} & \min & x_1 - 2x_2 \\ & \text{s.t.} & -x_1 + x_2 \leq 2 \\ & & 8x_1 + 2x_2 \leq 17 \\ & & x_1, x_2 \geq 0 \\ & & x_1, x_2 \text{ integer} \end{array}$$



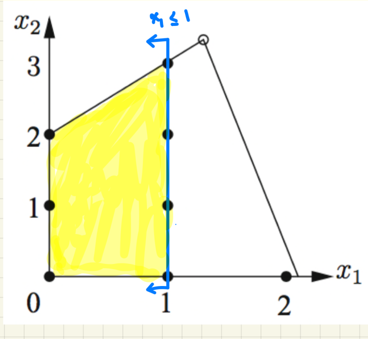
- Optimal solution to LP is $x^{\mathsf{LP}} = (1.3, 3.3)^T$ with objective $z^{\mathsf{LP}} = -5.3$
- We now have a lower bound for the IP: $\boldsymbol{L} = -5.3$
- We can branch on either x_1 or x_2
- Choose $x_1 \Rightarrow$ Create 2 subproblems by adding the constraints:

$$(i)\ x_1 \leq 1 \ \text{ and } \ (ii)\ \ x_1 \geq 2$$

to the IP.

Example

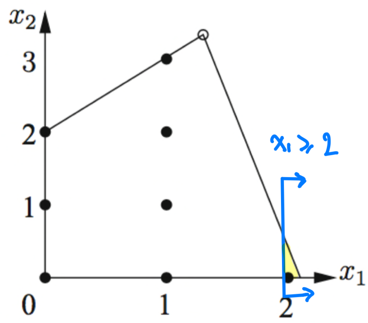
$$\begin{array}{lll} \text{(IP1)} & \min & x_1 - 2x_2 \\ & \text{s.t.} & -x_1 + x_2 \leq 2 \\ & & 8x_1 + 2x_2 \leq 17 \\ & & x_1 \leq 1 \\ & & x_1, x_2 \geq 0 \\ & & x_1, x_2 \text{ integer} \end{array}$$



- Optimal solution to LP1 is $x^{\text{LP1}} = (1,3)^T$ with $z^{\text{LP1}} = -5$
- As the LP solution is integral, we solved IP1 to optimality: $z^{\text{IP1}} = -5$
  (We do not need to explore this subproblem anymore.)
- We now have an upper bound for the IP: $\boldsymbol{U} = -5$
- Let's go back to the other subproblem IP2.

Example

$$
\begin{aligned}
\text{(IP2)} \quad \min \quad & x_1 - 2x_2 \\
\text{s.t.} \quad & -x_1 + x_2 \le 2 \\
& 8x_1 + 2x_2 \le 17 \\
& x_1 \ge 2 \\
& x_1, x_2 \ge 0 \\
& x_1, x_2 \text{ integer}
\end{aligned}
$$



- Optimal solution to LP2 is $x^{\mathsf{LP2}} = (2, 0.5)^T$ with $z^{\mathsf{LP2}} = 1$
- Notice that

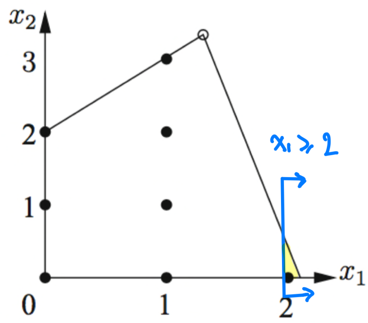
$$z^{\mathsf{IP2}} \ \ge \ z^{\mathsf{LP2}} \ = \ 1 \ > \ -5 \ = \ U$$

- There cannot be better integer solutions in this subproblem.
- We are done:

$$z^{\mathsf{IP}} = -5$$

Example

$$
\begin{aligned}
\text{(IP2)}\quad \min\quad & x_1 - 2x_2 \\
\text{s.t.}\quad & -x_1 + x_2 \le 2 \\
& 8x_1 + 2x_2 \le 17 \\
& x_1 \ge 2 \\
& x_1, x_2 \ge 0 \\
& x_1, x_2 \text{ integer}
\end{aligned}
$$



- Optimal solution to LP2 is $x^{\mathsf{LP2}} = (2, 0.5)^T$ with $z^{\mathsf{LP2}} = 1$
- Notice that

$$z^{\mathsf{IP2}} \ \ge\ z^{\mathsf{LP2}} \ =\ 1 \ >\ -5 \ =\ \boldsymbol{U}$$

- There cannot be better integer solutions in this subproblem.
- We are done:

$$z^{\mathsf{IP}} = -5$$

1/29/2024

Recap: Divide and conquer principle

Consider

$$z^* \;=\; \min\left\{f(x) \,:\, x \in S\right\}$$

If a collection of disjoint sets $\{S_1, S_2, \ldots, S_k\}$ satisfy

$$S = S_1 \cup S_2 \cup \ldots \cup S_k$$

then $\{S_1, S_2, \ldots, S_k\}$ is called a partition of S

Let $z^i = \min\{f(x) \,:\, x \in S_i\}$, and $\quad z^i \;\geq\; z^i_{LB} \quad \longleftarrow$ a lower bound

Observation 1:

$$z^* = \min\{z^1, z^2, z^3, \ldots, z^k\}$$

Observation 2:

$$z^* \leq \min\{z^1, z^3, z^8, \ldots\} \quad \longleftarrow \text{ some } z^i\text{s are missing here}$$

Observation 3:

$$z^* \;\geq\; \min\{z^1_{LB}, z^2_{LB}, \ldots, z^k_{LB}\}.$$

(In branch and bound, we dynamically decide how to partition of S.)

Consider a generic IP: $z^* = \min\left\{c^T x \, : \, x \in S\right\}$ where $S = P \cap \mathbb{Z}^n$

Partitioning S:

[0] S

[1] $S = S_1 \cup S_2$

$S_1 = S_3 \cup S_4$

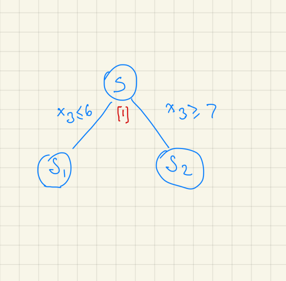
[2] $S = S_3 \cup S_4 \cup S_2$

$S_2 = S_5 \cup S_6$

[3] $S = S_3 \cup S_4 \cup S_5 \cup S_6$

$S_5 = S_7 \cup S_8$

[4] $S = S_3 \cup S_4 \cup S_7 \cup \cancel{S_8} \cup S_6$



Consider a generic IP: $z^* = \min\left\{c^T x \,:\, x \in S\right\}$ where $S = P \cap \mathbb{Z}^n$

Partitioning S:

[0] S

[1] $S = S_1 \cup S_2$

$$S_1 = S_3 \cup S_4$$

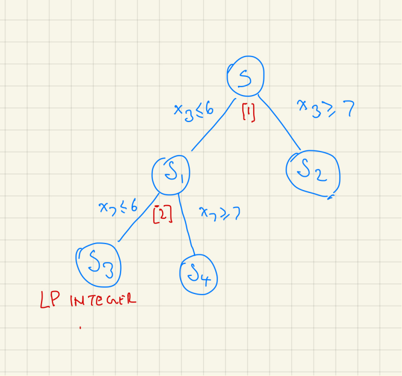
[2] $S = S_3 \cup S_4 \cup S_2$

$$S_2 = S_5 \cup S_6$$

[3] $S = S_3 \cup S_4 \cup S_5 \cup S_6$

$$S_5 = S_7 \cup S_8$$

[4] $S = S_3 \cup S_4 \cup S_7 \cup \cancel{S_8} \cup S_6$



Consider a generic IP: $z^* = \min\{c^T x : x \in S\}$ where $S = P \cap \mathbb{Z}^n$

Partitioning S:

[0] S

[1] $S = S_1 \cup S_2$

$$S_1 = S_3 \cup S_4$$

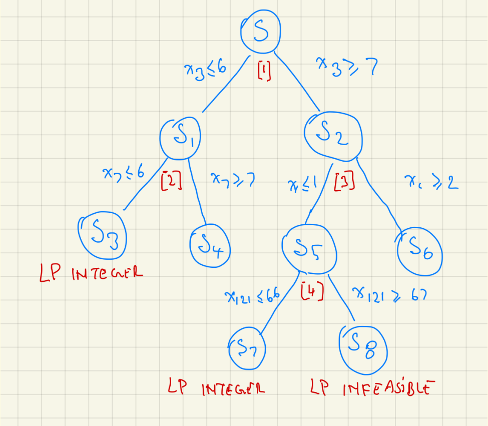
[2] $S = S_3 \cup S_4 \cup S_2$

$$S_2 = S_5 \cup S_6$$

[3] $S = S_3 \cup S_4 \cup S_5 \cup S_6$

$$S_5 = S_7 \cup S_8$$

[4] $S = S_3 \cup S_4 \cup S_7 \cup \cancel{S_8} \cup S_6$



Consider a generic IP: $z^* \;=\; \min\left\{c^T x \;:\; x \in S\right\}$ where $S = P \cap \mathbb{Z}^n$

Partitioning S:

$[0] \quad S$

$[1] \quad S = S_1 \cup S_2$

$$S_1 = S_3 \cup S_4$$

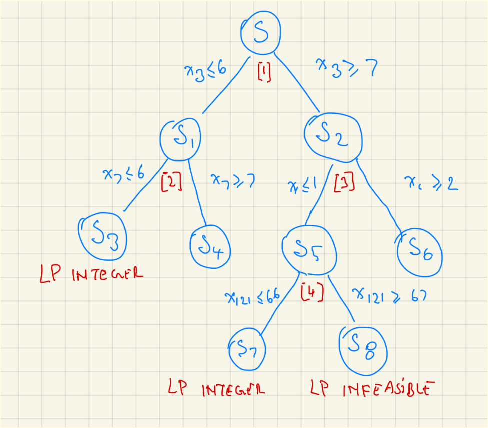
$[2] \quad S = S_3 \cup S_4 \cup S_2$

$$S_2 = S_5 \cup S_6$$

$[3] \quad S = S_3 \cup S_4 \cup S_5 \cup S_6$

$$S_5 = S_7 \cup S_8$$

$[4] \quad S = S_3 \cup S_4 \cup S_7 \cup \cancel{S_8} \cup S_6$



$$[1] \quad z^* \;=\; \min\{z^1, z^2\}$$

$$[2] \quad z^* \;=\; \min\{z^2, z^3, z^4\} \qquad \longleftarrow \qquad z^* \le \; \boldsymbol{U} = z^3$$

(IP^3 solved)

$$[3] \quad z^* \;=\; \min\{z^3, z^4, z^5, z^6\}$$

$$[4] \quad z^* \;=\; \min\{z^3, z^4, z^6, z^7, \cancel{z^8}\} \qquad \longleftarrow \qquad z^* \le \; \boldsymbol{U} = \min\left\{z^3, z^7\right\}$$

Consider a generic IP: $z^* = \min\left\{c^T x : x \in S\right\}$ where $S = P \cap \mathbb{Z}^n$

Partitioning S:

[0] S

[1] $S = S_1 \cup S_2$

$$S_1 = S_3 \cup S_4$$

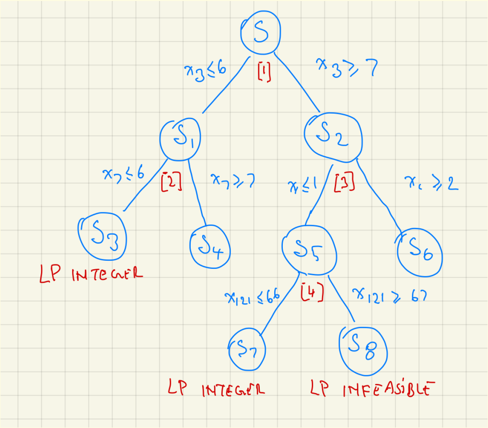
[2] $S = S_3 \cup S_4 \cup S_2$

$$S_2 = S_5 \cup S_6$$

[3] $S = S_3 \cup S_4 \cup S_5 \cup S_6$

$$S_5 = S_7 \cup S_8$$

[4] $S = S_3 \cup S_4 \cup S_7 \cup \cancel{S_8} \cup S_6$



$$z^* \geq \boldsymbol{L} = \min\{\cancel{z^3_{LB}}, z^4_{LB}, z^6_{LB}, \cancel{z^7_{LB}}, \cancel{z^8_{LB}}\}$$

$$z^* \leq \boldsymbol{U} = \min\{z^3, z^7\} \longleftarrow$$ because we solved IP^3 and IP^7

When $\boldsymbol{L} = \boldsymbol{U}$ we have solved the IP.

When we solve the LP relaxation of a subproblem

- When we solve the LP relaxation of IP_1 (call it LP_1)
  - If LP_1 is infeasible, then IP_1 is also infeasible.
  - If the optimal solution to LP_1 is integral then IP_1 is solved to optimality and $z^{\mathsf{IP}_1} = z^{\mathsf{LP}_1}$

    [As we have a feasible integer solution at hand, z^{IP_1} is an upper bound on IP (minimization)]
  - If the LP solution is **not** integral, then we can again divide IP_1 into two new subproblems IP_3 and IP_4.

- Repeating this process, we create subproblems IP_k for $k = 1, \ldots$
- We have to further divide any subproblem IP_k unless its LP relaxation LP_k returns an integral solution or it is infeasible.
- Notice that at each step we might replace one IP/LP with two!
- This might lead to exponential blow up!

  (However, this is still the best way to solve general IPs.)

When we solve the LP relaxation of a subproblem

- When we solve the LP relaxation of IP_1 (call it LP_1)
  - If LP_1 is infeasible, then IP_1 is also infeasible.
  - If the optimal solution to LP_1 is integral then IP_1 is solved to optimality and $z^{\mathsf{IP}_1} = z^{\mathsf{LP}_1}$

    [As we have a feasible integer solution at hand, z^{IP_1} is an upper bound on IP (minimization)]
  - If the LP solution is **not** integral, then we can again divide IP_1 into two new subproblems IP_3 and IP_4.

- Repeating this process, we create subproblems IP_k for $k = 1, \ldots$
- We have to further divide any subproblem IP_k unless its LP relaxation LP_k returns an integral solution or it is infeasible.
- Notice that at each step we might replace one IP/LP with two!
- This might lead to exponential blow up!

  (However, this is still the best way to solve general IPs.)

Pruning subprobems

- After solving its LP relaxation, we do not partition a subproblem IP_k into 2 new subproblems in one of these cases:
  - If its LP relaxation is infeasible.
  - If the LP solution is integral (in this case we solved IP_k to optimality and found a new lower bound $\boldsymbol{L} = z^{\mathsf{IP}_k}$ for IP)

- Let's now assume that we have already found an integral solution earlier, giving us an upper bound $\boldsymbol{U}$.
  - What happens if $z^{\mathsf{LP}_k} \geq \boldsymbol{U}$?
  - IP_k cannot contain a better solution to the IP

$$\underbrace{z^{\mathsf{IP}_k} \geq z^{\mathsf{LP}_k}}_{\text{relaxation}} \geq \underbrace{\boldsymbol{U} \geq z^*}_{\text{upper bound}}$$

  - Therefore, we do care about IP_k and we prune it (it cannot contain a better solution so we do not explore it further)

Pruning subprobems

- After solving its LP relaxation, we do not partition a subproblem IP_k into 2 new subproblems in one of these cases:
  - If its LP relaxation is infeasible.
  - If the LP solution is integral (in this case we solved IP_k to optimality and found a new lower bound $\boldsymbol{L} = z^{\text{IP}_k}$ for IP)

- Let's now assume that we have already found an integral solution earlier, giving us an upper bound $\boldsymbol{U}$.
  - What happens if $z^{\text{LP}_k} \geq \boldsymbol{U}$?
  - IP_k cannot contain a better solution to the IP

$$\underbrace{z^{\text{IP}_k} \geq z^{\text{LP}_k}}_{\text{relaxation}} \geq \underbrace{\boldsymbol{U} \geq z^*}_{\text{upper bound}}$$

  - Therefore, we do care about IP_k and we prune it (it cannot contain a better solution so we do not explore it further)

Branch&Bound for minimization problems

(IP) $z^* = \min\left\{c^T x : x \in P \cap \mathbb{Z}^n\right\}$

- Set the list of problems to solve to $\mathcal{L} \leftarrow \{\text{IP}\}$ and set $\boldsymbol{U} = +\infty$.
- **While** $\mathcal{L} \neq \emptyset$
  - Pick a subproblem IP$'$ from $\mathcal{L}$ and set $\mathcal{L} = \mathcal{L} \setminus \{\text{IP}'\}$
  - Relax IP$'$ $\rightarrow$ solve LP$'$ $\rightarrow$ obtain solution $x^{\text{LP}'}$ with obj. $z^{\text{LP}'}$
  - **If** LP$'$ is infeasible, **break**
  - **If** $x^{\text{LP}'}$ is integral, update $\boldsymbol{U} \leftarrow \min\{\boldsymbol{U}, z^{\text{LP}'}\}$, **break**
  - **If** the optimal value $z^{\text{LP}'} \geq \boldsymbol{U}$, **break**
  - Choose a fractional x_j i.e., $x_j^{\text{LP}'} \notin \mathbb{Z}$

    Create IP$''$ with additional constraint: $x_j \leq \lfloor x_j^{\text{LP}'} \rfloor$

    Create IP$'''$ with additional constraint: $x_j \geq \lceil x_j^{\text{LP}'} \rceil$

    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\text{IP}'', \text{IP}'''\}$

Branch&Bound for **maximization** problems

(IP) $z^* = \max\left\{c^T x : x \in P \cap \mathbb{Z}^n\right\}$

- Set the list of problems to solve to $\mathcal{L} \leftarrow \{\mathsf{IP}\}$ and set $\boldsymbol{L} = -\infty$.
- **While** $\mathcal{L} \neq \emptyset$
  - Pick a subproblem IP′ from $\mathcal{L}$ and set $\mathcal{L} = \mathcal{L} \setminus \{\mathsf{IP}'\}$
  - Relax IP′ $\rightarrow$ solve LP′ $\rightarrow$ obtain solution $x^{\mathsf{LP}'}$ with obj. $z^{\mathsf{LP}'}$
  - **If** LP′ is infeasible, **break**
  - **If** $x^{\mathsf{LP}'}$ is integral, update $\boldsymbol{L} \leftarrow \max\{\boldsymbol{L}, z^{\mathsf{LP}'}\}$, **break**
  - **If** the optimal value $z^{\mathsf{LP}'} \leq \boldsymbol{L}$, **break**
  - Choose a fractional component $x_j^{\mathsf{LP}'} \notin \mathbb{Z}$

    Create IP″ with additional constraint $x_j \leq \lfloor x_j^{\mathsf{LP}'} \rfloor$

    Create IP‴ with additional constraint $x_j \geq \lceil x_j^{\mathsf{LP}'} \rceil$

    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathsf{IP}'', \mathsf{IP}'''\}$

For maximization problems

(IP) $z^* = \max\left\{c^T x : x \in P \cap \mathbb{Z}^n\right\}$

B&B summary:

- $\mathcal{L} = \{IP^2, IP^5, IP^7, IP^8, IP^9, \ldots\}$ contains the subproblems that have the potential to contain the optimal solution to the IP.
- The best LP objective value of the subproblems in $\mathcal{L}$ give an upper bound on z^*

$$z^* \;\le\; \max\{z^{LP^2}, z^{LP^5}, z^{LP^7}, z^{LP^8}, z^{LP^9}, \ldots\} \quad \longleftarrow \; \boldsymbol{U}$$

- Whenever we encounter an integer solution we update

$$\boldsymbol{L} \leftarrow \max\{\boldsymbol{L}, z^{\mathsf{LP}'}\}$$

  (we also remember the best integral solution so far and call it the incumbent)

- We stop when $\mathcal{L} = \emptyset$. The incumbent is the optimal solution to IP with value $\boldsymbol{L}$.
- If we terminate early (i.e. $\mathcal{L} \neq \emptyset$) then we know: $\boldsymbol{U} \geq z^* \geq \boldsymbol{L}$.

One last example

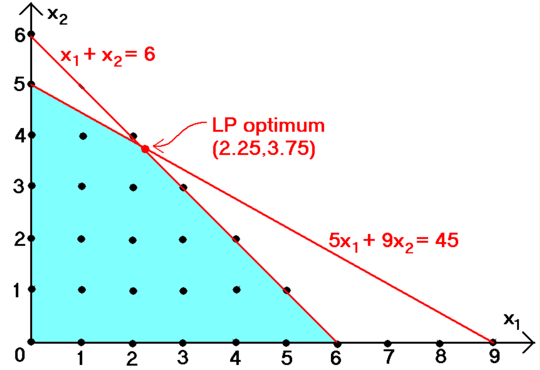
Example (maximization)

.

(IP)

$$\begin{array}{llllll} \max & 5x_1 & + & 8x_2 & & \\ \text{s.t.} & x_1 & + & x_2 & \le & 6 \\ & 5x_1 & + & 9x_2 & \le & 45 \end{array}$$

$x_1, x_2 \ge 0$, and integer



- The optimal solution of the LP relaxation is $x^* = [2.25, 3.75]^T$,
- The optimal LP objective value is $z^{LP} = 41.25$.
- Therefore we know that optimal IP value is at most $U = 41.25$

  (Note: This actually means that $U = 41 = \lfloor 41.25 \rfloor$, why?)
- We branch on x_2 : $x_2 \le 3$ or $x_2 \ge 4$

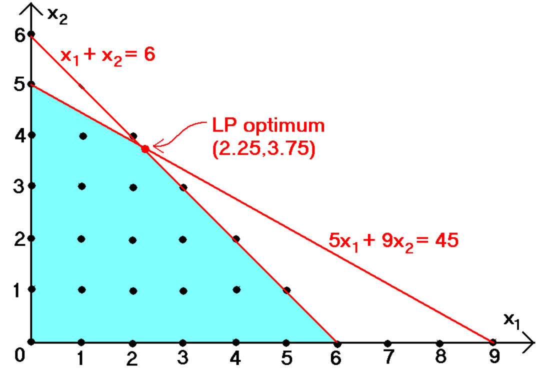
Example (maximization)

.

(IP)

$$\begin{array}{lrcrcl} \max & 5x_1 & + & 8x_2 & & \\ \text{s.t.} & x_1 & + & x_2 & \leq & 6 \\ & 5x_1 & + & 9x_2 & \leq & 45 \end{array}$$

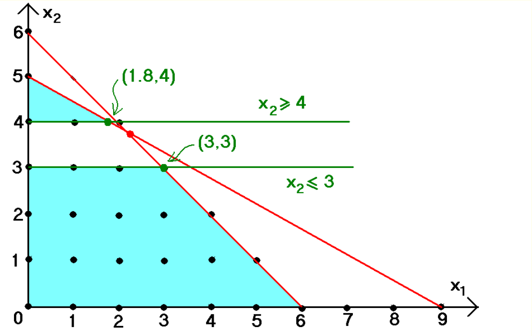
$x_1, x_2 \geq 0,$ and integer



- The optimal solution of the LP relaxation is $x^* = [2.25, 3.75]^T$,
- The optimal LP objective value is $z^{LP} = 41.25$.
- Therefore we know that optimal IP value is at most $U = 41.25$

  (Note: This actually means that $U = 41 = \lfloor 41.25 \rfloor$, why?)
- We branch on x_2 : $\quad x_2 \leq 3$ or $x_2 \geq 4$

Create 2 subproblems



(IP1)

$$\begin{array}{lrcrcl} \max & 5x_1 & + & 8x_2 & & \\ \text{s. t.} & x_1 & + & x_2 & \leq & 6 \\ & 5x_1 & + & 9x_2 & \leq & 45 \\ & & & x_2 & \leq & 3 \end{array}$$

$x_1, x_2 \geq 0,$ and integer

(IP2)

$$\begin{array}{lrcrcl} \max & 5x_1 & + & 8x_2 & & \\ \text{s. t.} & x_1 & + & x_2 & \leq & 6 \\ & 5x_1 & + & 9x_2 & \leq & 45 \\ & & & x_2 & \geq & 4 \end{array}$$

$x_1, x_2 \geq 0,$ and integer

Solve the LP relaxation of the subproblems

$$(\mathsf{IP}_1) \quad \max \quad 5x_1 + 8x_2 \quad \text{s.t.} \quad x \in P, \quad x_2 \leq 3, \quad x_1, x_2 \text{ integer}$$

$$(\mathsf{IP}_2) \quad \max \quad 5x_1 + 8x_2 \quad \text{s.t.} \quad x \in P, \quad x_2 \geq 4, \quad x_1, x_2 \text{ integer}$$

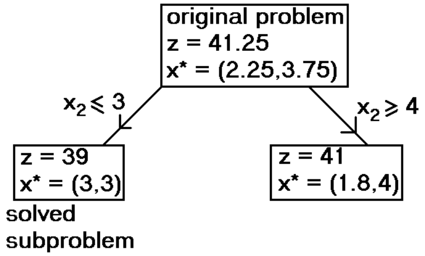
LP optimal solution:

$x^{LP1} = [3, 3]^T$, with $z^{LP1} = 39$

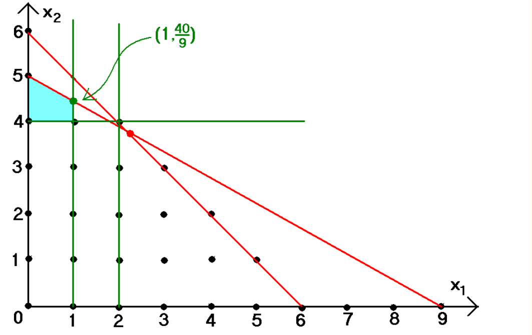
LP optimal solution:

$x^{LP2} = [1.8, 4]^T$, with $z^{LP2} = 41$

- IP_1 is solved to optimality. We have now a lower bound of $\boldsymbol{L} = 39$.
- We need to explore (divide) IP_2 further: branch on x_1.



Create 2 more subproblems



(IP$_3$) $\max \quad 5x_1 + 8x_2$

s. t. $\quad x \in P, \; x_2 \geq 4 \;\; x_1 \leq 1$

x_1, x_2 integer

LP optimal solution:

$x^{LP3} = [1, 4\frac{4}{9}]^T$ with $z^{LP1} = 40\frac{5}{9}$

(IP$_4$) $\max \quad 5x_1 + 8x_2$

s. t. $\quad x \in P, \; x_2 \geq 4 \;\; x_1 \geq 2$

x_1, x_2 integer

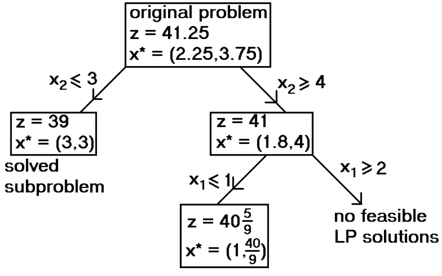
LP infeasible:

Subproblem IP$_4$ is also infeasible.

After solving x^{LP3} and x^{LP4},

this is the current B&B tree $\rightarrow$.

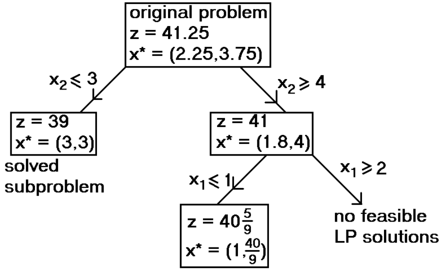
- We have a lower bound of $\boldsymbol{L} = 39$ for the IP
- We have improved the IP upper bound to $\boldsymbol{U} = 40\frac{5}{9}$ $\rightarrow 40$



After solving x^{LP3} and x^{LP4},

this is the current B&B tree $\rightarrow$.

- We have a lower bound of $\boldsymbol{L} = 39$ for the IP
- We have improved the IP upper bound to $\boldsymbol{U} = 40\frac{5}{9}$ $\rightarrow 40$



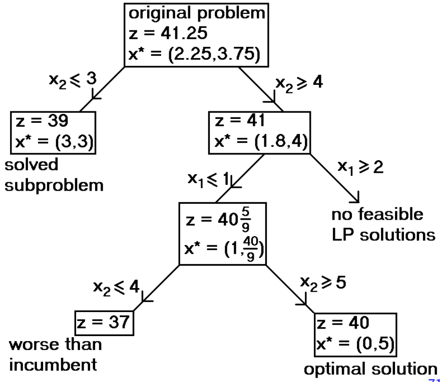
Exploring IP_3 further we create 2 new subproblems branching on x_2

- IP_5 has an additional constraint

$$x_2 \leq 4$$

- IP_6 has an additional constraint

$$x_2 \geq 5$$



Solving the knapsack problem

The 0-1 Knapsack problem

- You are going on a camping trip
- You have a knapsack that can carry a maximum weight $b > 0$.
- There are n different items that you could take.
- Each item of type i has weight $a_i > 0$.
- Each item of type i has value $c_i > 0$.
- You want to load the knapsack with some of these items
- Which items should you pack?
(Without exceeding the knapsack capacity)

The 0-1 knapsack problem

- As we can pack only one unit of each item we will use binary variables.
- The $0, 1$ **knapsack set** K **is:**

$$K := \left\{ x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b \right\}.$$

- The $0, 1$ **knapsack problem:**

$$\max\{c^T x : x \in K\}.$$

In other words,

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leq b \\ & x_i \in \{0,1\} \ \text{ for all } \ i = 1, \ldots, n. \end{aligned}$$

Why don't we just enumerate possible solutions?

- In the 0-1 Knapsack Problem, what we want is to :
  - Choose a subset $S \subseteq I$ of possible items $I = \{1, \ldots, n\}$
  - Make sure they fit: $\sum_{i \in S} a_i \leq b$
  - Maximize: reward $= \sum_{i \in S} c_i$

Why don't we simply numerate all possible subsets of I, consider the ones that weigh at most b, pick the best among them.

How long will it take to do this using the fastest supercomputer:

n	Solutions to check	Time
3	8	0
10	1024	0
50	2^{50}	2 sec
110		

Why don't we just enumerate possible solutions?

- In the 0-1 Knapsack Problem, what we want is to :
  - Choose a subset $S \subseteq I$ of possible items $I = \{1, \ldots, n\}$
  - Make sure they fit: $\sum_{i \in S} a_i \leq b$
  - Maximize: reward $= \sum_{i \in S} c_i$

Why don't we simply numerate all possible subsets of I, consider the ones that weigh at most b, pick the best among them.

How long will it take to do this using the fastest supercomputer:

n	Solutions to check	Time
3	8	0
10	1024	0
50	2^{50}	2 sec
110	2^{110}	Take a guess

Why don't we just enumerate possible solutions?

- In the 0-1 Knapsack Problem, what we want is to :
  - Choose a subset $S \subseteq I$ of possible items $I = \{1, \ldots, n\}$
  - Make sure they fit: $\sum_{i \in S} a_i \leq b$
  - Maximize: reward $= \sum_{i \in S} c_i$

Why don't we simply numerate all possible subsets of I, consider the ones that weigh at most b, pick the best among them.

How long will it take to do this using the fastest supercomputer:

n	Solutions to check	Time
3	8	0
10	1024	0
50	2^{50}	2 sec
110	2^{110}	69 billion years*

*That's four times the age of the universe as we know it!

(remember, I has 2^n distinct subsets.)

Solving The Knapsack Problem

- We will solve the following knapsack problem using branch-and-bound:

$$\begin{array}{rrrrrrl} z^{\mathsf{IP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\ \text{s. t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\ & x_1, & x_2, & x_3, & x_4, & x_5 & \in \{0,1\} \end{array}$$

- We start with solving its LP relaxation

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s. t.} \quad & \sum_{i=1}^{n} a_i x_i \leq b \\ & 1 \geq x_i \geq 0 \quad \cancel{x_i \in \{0,1\}} \quad \text{for all} \quad i = 1, \ldots, n. \end{aligned}$$

- The LP relaxation is solved using the greedy algorithm:

$$
\begin{array}{rllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
\text{s. t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9, \quad 1 \geq x_i \geq 0\ \forall i
\end{array}
$$

- We will next look at properties of an optimal solution of this LP.
- Let $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$ be the optimal solution to LP
- Can we have $\bar{x}_4 > \bar{x}_5$?
  - Notice that x_5 gives *more bang for the buck:*

$$
\frac{c_5}{a_5} = \frac{10}{2} > \frac{c_4}{a_4} = \frac{5}{4}
$$

  - If $\bar{x}_4 > \bar{x}_5$, using $1 \geq x_i \geq 0$ for all x_i, we know that

$$
1 \geq \bar{x}_4 > \bar{x}_5 \geq 0 \implies (i)\ 1 > \bar{x}_5, \quad \text{and} \qquad (ii)\ \bar{x}_4 > 0
$$

  - Now consider a new solution x' obtained by by decreasing $\bar{x}_4$ by a tiny number $\delta > 0$ and increasing $\bar{x}_5$ twice as much:

$$
x' = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4 - \delta, \bar{x}_5 + 2\delta)
$$

$$
\begin{array}{rrrrrrl}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9, \quad 1 \geq x_i \geq 0 \ \forall i
\end{array}
$$

- We will next look at properties of an optimal solution of this LP.
- Let $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$ be the optimal solution to LP
- Can we have $\bar{x}_4 > \bar{x}_5$?
  - Notice that x_5 gives *more bang for the buck:*

$$\frac{c_5}{a_5} = \frac{10}{2} > \frac{c_4}{a_4} = \frac{5}{4}$$

  - If $\bar{x}_4 > \bar{x}_5$, using $1 \geq x_i \geq 0$ for all x_i, we know that

$$1 \geq \bar{x}_4 > \bar{x}_5 \geq 0 \implies (i)\ 1 > \bar{x}_5, \quad \text{and} \qquad (ii)\ \bar{x}_4 > 0$$

  - Now consider a new solution x' obtained by by decreasing $\bar{x}_4$ by a tiny number $\delta > 0$ and increasing $\bar{x}_5$ twice as much:

$$x' = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4 - \delta, \bar{x}_5 + 2\delta)$$

$$
\begin{array}{llllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9, & 1 \geq x_i \geq 0 \ \forall i
\end{array}
$$

- If $\bar{x}_4 > \bar{x}_5$, then for some small $\delta > 0$, consider

$$x' = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4 - \delta, \bar{x}_5 + 2\delta)$$

- As $a_4/a_5 = 4/2 = 2$, the constraint is still satisfied:

$$\sum a_i x'_i = \sum a_i \bar{x}_i - 4\delta + 2 \cdot 2\delta = \sum a_i \bar{x}_i \leq b$$

- The new solution x' has a strictly better objective value for any $\delta > 0$:

$$\sum c_i x'_i = \sum c_i \bar{x}_i - 5\delta + 20\delta = \sum c_i \bar{x}_i + 15\delta > \sum c_i \bar{x}_i$$

- We also need to make sure that $1 \geq x' \geq 0$: How large can δ be?
  - $x'_4 = \bar{x}_4 - \delta \in [0,1] \implies \bar{x}_4 \geq \delta$
  - $x'_5 = \bar{x}_5 + 2\delta \in [0,1] \implies \bar{x}_5 + 2\delta \leq 1$

$\implies$ Pick $\delta = \min\left\{\bar{x}_4, \frac{1}{2}(1 - \bar{x}_5)\right\} > 0$ to obtain a better LP solution.

$$
\begin{array}{rllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \le 9, \quad 1 \ge x_i \ge 0\ \forall i
\end{array}
$$

- If $\bar{x}_4 > \bar{x}_5$, then for some small $\delta > 0$, consider

$$x' = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4 - \delta, \bar{x}_5 + 2\delta)$$

- As $a_4/a_5 = 4/2 = 2$, the constraint is still satisfied:

$$\sum a_i x_i' = \sum a_i \bar{x}_i - 4\delta + 2 \cdot 2\delta = \sum a_i \bar{x}_i \le b$$

- The new solution x' has a strictly better objective value for any $\delta > 0$:

$$\sum c_i x_i' = \sum c_i \bar{x}_i - 5\delta + 20\delta = \sum c_i \bar{x}_i + 15\delta > \sum c_i \bar{x}_i$$

- We also need to make sure that $1 \ge x' \ge 0$: How large can δ be?
  - $x_4' = \bar{x}_4 - \delta \in [0,1] \implies \bar{x}_4 \ge \delta$
  - $x_5' = \bar{x}_5 + 2\delta \in [0,1] \implies \bar{x}_5 + 2\delta \le 1$

$\implies$ Pick $\delta = \min\left\{\bar{x}_4, \frac{1}{2}(1 - \bar{x}_5)\right\} > 0$ to obtain a better LP solution.

$$
\begin{array}{llllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \le 9, & 1 \ge x_i \ge 0\ \forall i
\end{array}
$$

- We therefore established that if

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$$

is an optimal solution to LP then we cannot have $\bar{x}_4 > \bar{x}_5$.

- Then, in any optimal solution we must have $\bar{x}_5 \ge \bar{x}_4$.

[because $10/2 > 5/4$ meaning that x_5 gives more bang for the buck]

- How about $\bar{x}_4$ and $\bar{x}_3$?
- With the same reasoning optimal solution must have $\bar{x}_4 \ge \bar{x}_3$.

[because $5/4 > 4/5$]

- And $\bar{x}_3 \ge \bar{x}_2$, and $\bar{x}_2 \ge \bar{x}_1$.

You will always prefer items that give *more bang for the buck* for the LP.

$$
\begin{array}{rllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9, \quad 1 \geq x_i \geq 0\ \forall i
\end{array}
$$

- We therefore established that if

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$$

is an optimal solution to LP then we cannot have $\bar{x}_4 > \bar{x}_5$.

- Then, in any optimal solution we must have $\bar{x}_5 \geq \bar{x}_4$.

  [because $10/2 > 5/4$ meaning that x_5 gives more bang for the buck]

- How about $\bar{x}_4$ and $\bar{x}_3$?
- With the same reasoning optimal solution must have $\bar{x}_4 \geq \bar{x}_3$.

  [because $5/4 > 4/5$]

- And $\bar{x}_3 \geq \bar{x}_2$, and $\bar{x}_2 \geq \bar{x}_1$.

You will always prefer items that give *more bang for the buck* for the LP.

$$
\begin{array}{rllllll}
z^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
\text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9, \quad 1 \geq x_i \geq 0 \ \forall i
\end{array}
$$

- We therefore established that if

$$\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$$

  is an optimal solution to LP then we cannot have $\bar{x}_4 > \bar{x}_5$.

- Then, in any optimal solution we must have $\bar{x}_5 \geq \bar{x}_4$.

  [because $10/2 > 5/4$ meaning that x_5 gives more bang for the buck]

- How about $\bar{x}_4$ and $\bar{x}_3$?

- With the same reasoning optimal solution must have $\bar{x}_4 \geq \bar{x}_3$.

  [because $5/4 > 4/5$]

- And $\bar{x}_3 \geq \bar{x}_2$, and $\bar{x}_2 \geq \bar{x}_1$.

  You will always prefer items that give *more bang for the buck* for the LP.

Solving the 0-1 knapsack problem with branch and bound

- To solve the LP relaxation of the Knapsack problem:

$$\begin{array}{rrrrrrrl} z^{\mathsf{IP}} = & \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\ & \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\ & & x_1, & x_2, & x_3, & x_4, & x_5 & \in \{0,1\} \end{array}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{2}{8} \leq \frac{1}{2} \leq \frac{4}{5} \leq \frac{5}{4} \leq \frac{10}{2} \leftarrow \text{most desirable}$$

- Therefore, in the optimal solution to LP, we have:

$$1 \geq x_5^{LP} \geq x_4^{LP} \geq x_3^{LP} \geq x_2^{LP} \geq x_1^{LP} \geq 0$$

Solving the 0-1 knapsack problem with branch and bound

- To solve the LP relaxation of the Knapsack problem:

$$\begin{array}{rllllll} z^{\mathsf{IP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\ \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\ & x_1, & x_2, & x_3, & x_4, & x_5 & \in \{0,1\} \end{array}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{2}{8} \leq \frac{1}{2} \leq \frac{4}{5} \leq \frac{5}{4} \leq \frac{10}{2} \leftarrow \text{most desirable}$$

- Therefore, in the optimal solution to LP, we have:

$$1 \geq x_5^{LP} \geq x_4^{LP} \geq x_3^{LP} \geq x_2^{LP} \geq x_1^{LP} \geq 0$$

- Moreover if $x_4^{LP} = x_5^{LP}$ then they must either both 0, or 1.

  [otherwise, you can again decrease x_4 by δ and increase x_5 by 2δ to improve the objective.]

- Therefore, you will fill your knapsack (fractionally) with more profitable (meaning, larger c_i/a_i) items first.

Solving the 0-1 knapsack problem with branch and bound

- To solve the LP relaxation of the Knapsack problem:

$$\begin{array}{rrrrrrrl} z^{\mathsf{IP}} = & \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\ & \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\ & & x_1, & x_2, & x_3, & x_4, & x_5 & \in \{0,1\} \end{array}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{2}{8} \leq \frac{1}{2} \leq \frac{4}{5} \leq \frac{5}{4} \leq \frac{10}{2} \leftarrow \text{most desirable}$$

- Therefore, in the optimal solution to LP, we have:

$$1 \geq x_5^{LP} \geq x_4^{LP} \geq x_3^{LP} \geq x_2^{LP} \geq x_1^{LP} \geq 0$$

- The LP relaxation is solved using the greedy algorithm:
  - Set the remaining budget $B \leftarrow 9$
  - **For** $j = 5, 4, 3, 2, 1$ (Sorted from best to worst reward/weight ratio.)

    if $a_i \leq B$, **then** set $x_i^{LP} \leftarrow 1$ and $B \leftarrow B - a_i$

    if $a_i \leq B$, **else** set $x_i^{LP} \leftarrow B/a_i$; **stop**

Solve the LP relaxation

Solve the LP relaxation using the greedy algorithm:

$$
\begin{array}{rrrrrrrl}
z^{\mathsf{LP}} = & \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
& \text{s. t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \in [0,1]
\end{array}
$$

- Sort the items in decreasing order of reward/weight ratio:

$$5, 4, 3, 2, 1$$

- Set the current budget $B = 9$
- **for** $i = 5, 4, 3, 2, 1$ (In order of best to worst reward/weight ratio.)

$[i = 5]$ $2 = a_5 \leq B = 9$, therefore we set $x_5^{LP} = 1$ and $B = 9 - 2 = 7$

$[i = 4]$ $4 = a_4 \leq B = 7$, therefore we set $x_4^{LP} = 1$ and $B = 7 - 4 = 3$

$[i = 3]$ $5 = a_3 \not\leq B = 3$, therefore we set $x_3^{LP} = B/a_3 = 3/5$; **stop**

$$x^{\mathsf{LP}} = (0, 0, 3/5, 1, 1)$$

Solve the LP relaxation

- Solve the LP relaxation using the greedy algorithm:

$$
\begin{array}{rrrrrrrl}
z^{\mathsf{LP}} = & \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
& \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \in [0,1]
\end{array}
$$

- Solution: $x^{\mathsf{LP}} = (0, 0, 3/5, 1, 1) \quad z^{\mathsf{LP}} = 17\frac{2}{5}$
- z^{LP} gives an upper bound for IP: $\boldsymbol{U} = 17\frac{2}{5}$.
- We do not have a lower bound (need an integral solution) (do we have one?)
- Split the problem into two sub-problems:
- $x_3^{\mathsf{LP}} = 3/5$ is the only fractional variable
  - IP_1: $x_3 \leq 0 \implies x_3 = 0$ (because $1 \geq x_3 \geq 0$)
  - IP_2: $x_3 \geq 1 \implies x_3 = 1$ (because $1 \geq x_3 \geq 0$)

Solve the LP relaxation

- Solve the LP relaxation using the greedy algorithm:

$$\begin{array}{rrrrrrrl} z^{\mathsf{LP}} = & \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\ & \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \leq 9 \\ & & x_1, & x_2, & x_3, & x_4, & x_5 & \in [0,1] \end{array}$$

- Solution: $x^{\mathsf{LP}} = (0, 0, 3/5, 1, 1) \quad z^{\mathsf{LP}} = 17\frac{2}{5}$
- z^{LP} gives an upper bound for IP: $\boldsymbol{U} = 17\frac{2}{5}$.
- We do not have a lower bound (need an integral solution) (do we have one?)
- Split the problem into two sub-problems:
- $x_3^{\mathsf{LP}} = 3/5$ is the only fractional variable
  - IP_1: $x_3 \leq 0 \implies x_3 = 0$ (because $1 \geq x_3 \geq 0$)
  - IP_2: $x_3 \geq 1 \implies x_3 = 1$ (because $1 \geq x_3 \geq 0$)

Consider subproblem IP_1

Solve LP_1 with $x_3 = 0$ (we will consider IP_2 later)

$$
\begin{array}{rrrrrrrrl}
z^{\mathsf{LP1}} & = & \max & 2x_1 & +1x_2 & +\ 0 & +5x_4 & +10x_5 & \\
 & & \text{s. t.} & 8x_1 & +2x_2 & +\ 0 & +4x_4 & +2x_5 & \leq 9 \\
 & & & x_1, & x_2, & & x_4, & x_5 & \in [0,1]
\end{array}
$$

- Solve LP_1 using the greedy algorithm.
- Solution: $x^{\mathsf{LP1}} = (1/8, 1, 0, 1, 1) \quad z^{\mathsf{LP1}} = 16\frac{1}{4}$
- x_1^{LP1} is fractional
- We will need to split the problem IP_1 into two sub-problems:
  - IP_3 would have an additional constraint $x_1 = 0$
  - IP_4 would have an additional constraint: $x_1 = 1$

Consider subproblem IP_1

Solve LP_1 with $x_3 = 0$ (we will consider IP_2 later)

$$
\begin{array}{rlrrrrrl}
z^{\mathsf{LP1}} = & \max & 2x_1 & +1x_2 & +\,0 & +5x_4 & +10x_5 & \\
& \text{s.t.} & 8x_1 & +2x_2 & +\,0 & +4x_4 & +2x_5 & \leq 9 \\
& & x_1, & x_2, & & x_4, & x_5 & \in [0,1]
\end{array}
$$

- Solve LP_1 using the greedy algorithm.
- Solution: $x^{\mathsf{LP1}} = (1/8, 1, 0, 1, 1) \quad z^{\mathsf{LP1}} = 16\frac{1}{4}$
- x_1^{LP1} is fractional
- We will need to split the problem IP_1 into two sub-problems:
  - IP_3 would have an additional constraint $x_1 = 0$
  - IP_4 would have an additional constraint: $x_1 = 1$

Consider subproblem IP_2

Solve LP_2 problem with $x_3 = 1$

$$\begin{array}{rcllllll} z_2^{\text{LP}} & = & \max & 2x_1 & +1x_2 & +\,4 & +5x_4 & +10x_5 & \\ & & \text{s.t.} & 8x_1 & +2x_2 & +\,5 & +4x_4 & +2x_5 & \leq 9 \\ & & & x_1, & x_2, & & x_4, & x_5 & \in [0,1] \end{array}$$

- Solve LP_2 (with budget $9 - 5 = 4$) using the greedy algorithm.
- Solution: $x^{\text{LP2}} = (0, 0, 1, 1/2, 1) \quad z^{\text{LP2}} = 16\frac{1}{2}$
- x_4^{LP2} is fractional
- We need to split the problem IP_2 into two sub-problems:
  - IP_5: $x_4 = 0$
  - IP_6: $x_4 = 1$

Consider subproblem IP_2

Solve LP_2 problem with $x_3 = 1$

$$
\begin{array}{rcrrrrrl}
z_2^{\mathsf{LP}} = \max & 2x_1 & +1x_2 & +\ 4 & +5x_4 & +10x_5 & \\
\text{s.t.} & 8x_1 & +2x_2 & +\ 5 & +4x_4 & +2x_5 & \leq 9 \\
& x_1, & x_2, & & x_4, & x_5 & \in [0,1]
\end{array}
$$

- Solve LP_2 (with budget $9 - 5 = 4$) using the greedy algorithm.
- Solution: $x^{\mathsf{LP2}} = (0, 0, 1, 1/2, 1) \quad z^{\mathsf{LP2}} = 16\frac{1}{2}$
- x_4^{LP2} is fractional
- We need to split the problem IP_2 into two sub-problems:
  - IP_5: $x_4 = 0$
  - IP_6: $x_4 = 1$

Knapsack B&B

$$
\begin{array}{llllllll}
\text{(IP)} & z^{\mathsf{IP}} = \max & 2x_1 & +1x_2 & +4x_3 & +5x_4 & +10x_5 & \\
& \text{s.t.} & 8x_1 & +2x_2 & +5x_3 & +4x_4 & +2x_5 & \le 9 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \in \{0,1\}
\end{array}
$$

LP solution: $x^{\mathsf{LP}} = (0, 0, 3/5, 1, 1), \quad z^{\mathsf{LP}} = 17\frac{2}{5}$

(IP$_1$) $[x_3 = 0]$ $\quad x^{\mathsf{LP1}} = (1/8, 1, 0, 1, 1) \quad z^{\mathsf{LP1}} = 16\frac{1}{4}$

(IP$_3$) $[x_3 = 0, x_1 = 0]$ $\quad x^{\mathsf{LP3}} = (0, 1, 0, 1, 1),\ z^{\mathsf{LP3}} = z^{\mathsf{IP3}} = 16 \ \leftarrow\ \boldsymbol{L}$

(IP$_4$) $[x_3 = 0, x_1 = 1]$ $\quad x^{\mathsf{LP4}} = (1, 0, 0, 0, 1/2), \quad z^{\mathsf{LP4}} = 7 < \boldsymbol{L}$

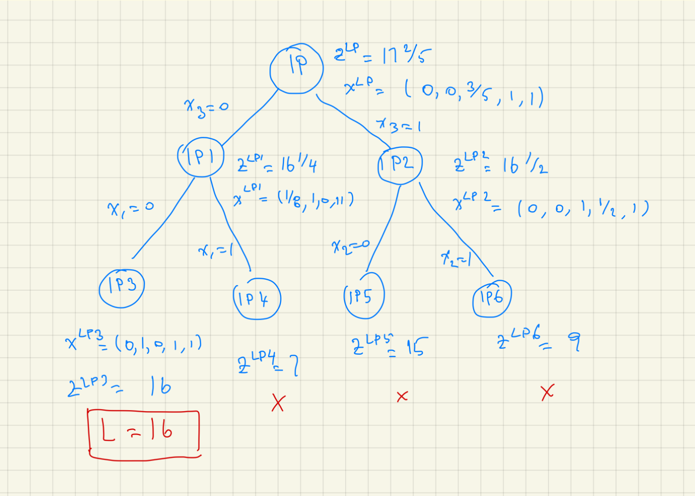
(IP$_2$) $x_3 = 1$ $\quad x^{\mathsf{LP2}} = (0, 0, 1, 1/2, 1) \quad w^2 = 16\frac{1}{2}$

(IP$_5$) $[x_3 = 1, x_4 = 0]$ $\quad x^{\mathsf{LP5}} = (0, 1, 1, 0, 1), \quad z^{\mathsf{LP5}} = z^{\mathsf{IP5}} = 15 < \boldsymbol{L}$

(IP$_6$) $[x_3 = 1, x_4 = 1]$ $\quad x^{\mathsf{LP6}} = (0, 0, 1, 1, 0), \quad z^{\mathsf{LP6}} = z^{\mathsf{IP6}} = 9 < \boldsymbol{L}$

The B&B tree

$$(\text{IP}) \quad z^{\text{IP}} = \max \; 2x_1 + 1x_2 + 4x_3 + 5x_4 + 10x_5$$
$$\text{s.t.} \; 8x_1 + 2x_2 + 5x_3 + 4x_4 + 2x_5 \le 9$$
$$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$$



A small detour: What is the "knap" in knapsack?

- In English knap means
  - Crest of a hill (summit), if it is a noun
  - To break with a quick blow, if it is a verb.
- But then, why do we call a knapsack a knapsack?
- Possible answers from different languages:
  (this part is not completely factual)
  - It comes from Arabic, where it means treasure.
  - It comes from German, where it means to bite
  - It comes from Dutch, where it means laundry.

  Question: Which one is it?

A small detour: What is the "knap" in knapsack?

- In English knap means
  - Crest of a hill (summit), if it is a noun
  - To break with a quick blow, if it is a verb.
- But then, why do we call a knapsack a knapsack?
- Possible answers from different languages:
  (this part is not completely factual)
  - It comes from Arabic, where it means treasure.
  - It comes from German, where it means to bite
  - It comes from Dutch, where it means laundry.

  Question: Which one is it?

  Answer: Knapsack comes from the German *knappen*, "to bite," and some experts believe that the name evolved from the fact that soldiers carried food in their knapsacks.

Integer Knapsack

(1/31/2024)

The integer knapsack problem

- You are going on a camping trip
- You have a knapsack that can carry a maximum weight $b > 0$.
- There are n different items that you could take.
- Each item of type i has weight $a_i > 0$.
- Each item of type i has value $c_i > 0$.
- You can pack up to u_i copies of item of type i.
- You want to load the knapsack with items (possibly several items of the same type).
- Which how many copies of each item should you pack?
  (Without exceeding the knapsack capacity)

IP formulation of the ~~0-1~~ integer knapsack problem

- Now we can pack multiple copies of the same item in the *knap*sack,
- Let variable $x_i \in \mathbb{Z}$ represent the number of items of type i that you pack.
- The **knapsack set** S contains all feasible loads:

$$S := \Big\{ x \in \mathbb{Z}^n : \sum_{i=1}^{n} a_i x_i \leq b,\ u \ \geq x \geq 0 \Big\}.$$

- The **integer knapsack problem is:**

$$\max \Big\{ \sum_{i=1}^{n} c_i x_i : x \in S \Big\}.$$

In other words,

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i \\ \text{s. t.} \quad & \sum_{i=1}^{n} a_i x_i \leq b \end{aligned}$$

$u_i \geq x_i \geq 0$ and integer for all $i = 1, \ldots, n$

Solving the integer knapsack problem with B&B

- To solve the LP relaxation of the Knapsack problem:

$$
\begin{array}{rlllll}
z^{\mathsf{IP}} = \max & 1x_1 & +6x_2 & +4x_3 & +5x_4 & \\
\text{s.t.} & 4x_1 & +10x_2 & +5x_3 & +4x_4 & \leq 29
\end{array}
$$

$$3 \geq x_1 \geq 0,\ 2 \geq x_2 \geq 0,\ 1 \geq x_3 \geq 0,\ 2 \geq x_4 \geq 0 \text{ and integer}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{1}{4} \leq \frac{6}{10} \leq \frac{4}{5} \leq \frac{5}{4} \leftarrow \text{most desirable}$$

- In the optimal LP solution more desirable items would be picked first (up to their upper bounds) before less desirable items.

Solving the integer knapsack problem with B&B

- To solve the LP relaxation of the Knapsack problem:

$$\begin{array}{rrrrrrrr} z^{\mathsf{IP}} & = & \max & 1x_1 & +\,6x_2 & +4x_3 & +5x_4 & \\ & & \text{s.t.} & 4x_1 & +10x_2 & +5x_3 & +4x_4 & \leq\ 29 \end{array}$$

$$3 \geq x_1 \geq 0,\ 2 \geq x_2 \geq 0,\ 1 \geq x_3 \geq 0,\ 2 \geq x_4 \geq 0 \text{ and integer}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{1}{4} \leq \frac{6}{10} \leq \frac{4}{5} \leq \frac{5}{4} \leftarrow \text{most desirable}$$

- In the optimal LP solution more desirable items would be picked first (up to their upper bounds) before less desirable items.
- Meaning
  - If $x_4 < 2$, then $x_3 = 0$
  - If $x_3 < 1$, then $x_2 = 0$
  - If $x_2 < 2$, then $x_1 = 0$

Solving the integer knapsack problem with B&B

- To solve the LP relaxation of the Knapsack problem:

$$
\begin{array}{rrllll}
z^{\mathsf{IP}} = & \max & 1x_1 & +\,6x_2 & +4x_3 & +5x_4 \\
& \text{s.t.} & 4x_1 & +10x_2 & +5x_3 & +4x_4 \quad \leq \; 29
\end{array}
$$

$$3 \geq x_1 \geq 0,\; 2 \geq x_2 \geq 0,\; 1 \geq x_3 \geq 0,\; 2 \geq x_4 \geq 0 \text{ and integer}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \;\rightarrow\; \frac{1}{4} \;\leq\; \frac{6}{10} \;\leq\; \frac{4}{5} \;\leq\; \frac{5}{4} \;\leftarrow\; \text{most desirable}$$

- The LP relaxation is solved using the greedy algorithm:
  - Set the remaining budget $B \leftarrow 29$
  - **for** $i = 4, 3, 2, 1$ (Sorted from best to worst reward/weight ratio.)

    if $a_i u_i \leq B$, **then** set $x_i^* \leftarrow u_i$ and $B \leftarrow B - a_i u_i$

    if $a_i u_i \leq B$, **else** set $x_i^* \leftarrow B/a_i$; **stop**

Solving the integer knapsack problem with B&B

- To solve the LP relaxation of the Knapsack problem:

$$\begin{array}{rlllllll} z^{\mathsf{IP}} = & \max & 1x_1 & +6x_2 & +4x_3 & +5x_4 & & \\ & \text{s.t.} & 4x_1 & +10x_2 & +5x_3 & +4x_4 & \leq & 29 \end{array}$$

$$3 \geq x_1 \geq 0,\ 2 \geq x_2 \geq 0,\ 1 \geq x_3 \geq 0,\ 2 \geq x_4 \geq 0 \text{ and integer}$$

- Look at the reward/weight ratio (c_i/a_i) of each item and sort the items:

$$\text{least desirable} \rightarrow \frac{1}{4} \leq \frac{6}{10} \leq \frac{4}{5} \leq \frac{5}{4} \leftarrow \text{most desirable}$$

- **For** $i = 4, 3, 2, 1$ (In order of best to worst reward/weight ratio.)

$[j = 4]$ $8 = a_4 u_4 \leq B = 29$, $\Rightarrow$ set $x_4^{LP} = 2$ and $B = 29 - 8 = 21$

$[j = 3]$ $5 = a_3 u_3 \leq B = 21$, $\Rightarrow$ set $x_3^{LP} = 1$; and $B = 21 - 5 = 16$

$[j = 3]$ $20 = a_2 u_2 \not\leq B = 16$, $\Rightarrow$ set $x_2^{LP} = B/a_2 = 16/10$; **Stop**

$$x^{\mathsf{LP}} = (0,\ 1.6,\ 1,\ 2) \quad \text{and} \quad z^{\mathsf{LP}} = 6(1.4) + 4(1) + 5(2) = 22.4 \leftarrow \boldsymbol{U}$$

(rest will be in next homework)

Comparing formulations

Comparing formulations

What can you say about the following 3 sets:

$$K^1 = \left\{ \; x \in \{0,1\}^3 \;\; : 3x_1 + 4x_2 + 5x_3 \leq 6 \right\}$$

$$K^2 = \left\{ \; x \in \{0,1\}^3 : \begin{array}{r} x_1 + x_2 \leq 1 \\ x_1 + x_3 \leq 1 \\ x_2 + x_3 \leq 1 \end{array} \right\}$$

$$K^3 = \left\{ \; x \in \{0,1\}^3 \;\; : x_1 + x_2 + x_3 \leq 1 \right\}$$

Are they the same or are they different?

Comparing formulations

What can you say about the following 3 sets:

$$K^1 = \left\{ \ x \in \{0,1\}^3 \ : 3x_1 + 4x_2 + 5x_3 \leq 6 \right\}$$

$$K^2 = \left\{ \ x \in \{0,1\}^3 : \begin{array}{r} x_1 + x_2 \leq 1 \\ x_1 + x_3 \leq 1 \\ x_2 + x_3 \leq 1 \end{array} \right\}$$

$$K^3 = \left\{ \ x \in \{0,1\}^3 \ : x_1 + x_2 + x_3 \leq 1 \right\}$$

Are they the same or are they different?

They are the same:

$$K^1 = K^2 = K^3 = \Big\{(0,0,0), (1,0,0), (0,1,0), (0,0,1)\Big\}$$

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \left\{ x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \le 6,\ 1 \ge x_i \ge 0\ \forall i \right\}$$

$$P^2 = \left\{ x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \le 1, & 1 \ge x_1 \ge 0 \\ x_1 + x_3 \le 1, & 1 \ge x_2 \ge 0 \\ x_2 + x_3 \le 1, & 1 \ge x_3 \ge 0 \end{array} \right\}$$

$$P^3 = \left\{ x \in \mathbb{R}^3 : x_1 + x_2 + x_3 \le 1,\ 1 \ge x_i \ge 0\ \forall i \right\}$$

Claim 1: $P^1 \supsetneq P^2$:

Proof. If $\bar{x} \in P^2$ then

$$\left.\begin{array}{rcl} \bar{x}_1 + \bar{x}_2 & \le & 1 \\ 2\bar{x}_1 + 2\bar{x}_3 & \le & 2 \\ 3\bar{x}_2 + 3\bar{x}_3 & \le & 3 \end{array}\right\} \implies 3\bar{x}_1 + 4\bar{x}_2 + 5\bar{x}_3 \le 6 \implies \bar{x} \in P^1$$

Did we prove the claim? No. We also need to argue that $P^1 \neq P^2$

We have $x' = (1,\ 3/4,\ 0) \in P^1$ but $x' \notin P^2$ $\square$

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \{\ x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \leq 6,\ 1 \geq x_i \geq 0\ \forall i\}$$

$$P^2 = \left\{\ x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \leq 1, & 1 \geq x_1 \geq 0 \\ x_1 + x_3 \leq 1, & 1 \geq x_2 \geq 0 \\ x_2 + x_3 \leq 1, & 1 \geq x_3 \geq 0 \end{array} \right\}$$

$$P^3 = \{\ x \in \mathbb{R}^3\ : x_1 + x_2 + x_3 \leq 1,\ 1 \geq x_i \geq 0\ \forall i\}$$

Claim 1: $P^1 \supsetneq P^2$:

Proof. If $\bar{x} \in P^2$ then

$$\left.\begin{array}{rcl} \bar{x}_1 + \bar{x}_2 & \leq & 1 \\ 2\bar{x}_1 + 2\bar{x}_3 & \leq & 2 \\ 3\bar{x}_2 + 3\bar{x}_3 & \leq & 3 \end{array}\right\} \implies 3\bar{x}_1 + 4\bar{x}_2 + 5\bar{x}_3 \leq 6 \implies \bar{x} \in P^1$$

Did we prove the claim? No. We also need to argue that $P^1 \neq P^2$

We have $x' = (1,\ 3/4,\ 0) \in P^1$ but $x' \notin P^2$ $\square$

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \{\ x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \leq 6,\ 1 \geq x_i \geq 0\ \forall i\}$$

$$P^2 = \left\{\ x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \leq 1, & 1 \geq x_1 \geq 0 \\ x_1 + x_3 \leq 1, & 1 \geq x_2 \geq 0 \\ x_2 + x_3 \leq 1, & 1 \geq x_3 \geq 0 \end{array} \right\}$$

$$P^3 = \{\ x \in \mathbb{R}^3\ : x_1 + x_2 + x_3 \leq 1,\ 1 \geq x_i \geq 0\ \forall i\}$$

Claim 1: $P^1 \supsetneq P^2$:

Proof. If $\bar{x} \in P^2$ then

$$\left.\begin{array}{rcl} \bar{x}_1 + \bar{x}_2 & \leq & 1 \\ 2\bar{x}_1 + 2\bar{x}_3 & \leq & 2 \\ 3\bar{x}_2 + 3\bar{x}_3 & \leq & 3 \end{array}\right\} \implies 3\bar{x}_1 + 4\bar{x}_2 + 5\bar{x}_3 \leq 6 \implies \bar{x} \in P^1$$

Did we prove the claim? No. We also need to argue that $P^1 \neq P^2$

We have $x' = (1,\ 3/4,\ 0) \in P^1$ but $x' \notin P^2$ □

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \left\{\ x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \leq 6,\ 1 \geq x_i \geq 0\ \forall i \right\}$$

$$P^2 = \left\{\ x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \leq 1, & 1 \geq x_1 \geq 0 \\ x_1 + x_3 \leq 1, & 1 \geq x_2 \geq 0 \\ x_2 + x_3 \leq 1, & 1 \geq x_3 \geq 0 \end{array} \right\}$$

$$P^3 = \left\{\ x \in \mathbb{R}^3\ : x_1 + x_2 + x_3 \leq 1,\ 1 \geq x_i \geq 0\ \forall i \right\}$$

Claim 1: $P^1 \supsetneq P^2$:

Proof. If $\bar{x} \in P^2$ then

$$\left.\begin{array}{rcl} \bar{x}_1 + \bar{x}_2 & \leq & 1 \\ 2\bar{x}_1 + 2\bar{x}_3 & \leq & 2 \\ 3\bar{x}_2 + 3\bar{x}_3 & \leq & 3 \end{array}\right\} \implies 3\bar{x}_1 + 4\bar{x}_2 + 5\bar{x}_3 \leq 6 \implies \bar{x} \in P^1$$

Did we prove the claim? No. We also need to argue that $P^1 \neq P^2$

We have $x' = (1,\ 3/4,\ 0) \in P^1$ but $x' \notin P^2$ □

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \left\{ \, x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \leq 6, \; 1 \geq x_i \geq 0 \;\forall i \right\}$$

$$P^2 = \left\{ \; x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \leq 1, & 1 \geq x_1 \geq 0 \\ x_1 + x_3 \leq 1, & 1 \geq x_2 \geq 0 \\ x_2 + x_3 \leq 1, & 1 \geq x_3 \geq 0 \end{array} \right\}$$

$$P^3 = \left\{ \, x \in \mathbb{R}^3 \; : x_1 + x_2 + x_3 \leq 1, \; 1 \geq x_i \geq 0 \;\forall i \right\}$$

Claim 1: $P^1 \supsetneq P^2$:

Proof. If $\bar{x} \in P^2$ then

$$\left. \begin{array}{rcl} \bar{x}_1 + \bar{x}_2 & \leq & 1 \\ 2\bar{x}_1 + 2\bar{x}_3 & \leq & 2 \\ 3\bar{x}_2 + 3\bar{x}_3 & \leq & 3 \end{array} \right\} \implies 3\bar{x}_1 + 4\bar{x}_2 + 5\bar{x}_3 \leq 6 \implies \bar{x} \in P^1$$

Did we prove the claim? No. We also need to argue that $P^1 \neq P^2$

We have $x' = (1, \; 3/4, \; 0) \in P^1$ but $x' \notin P^2$ $\square$

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \left\{\ x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \le 6,\ 1 \ge x_i \ge 0\ \forall i\right\}$$

$$P^2 = \left\{\ x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \le 1, & 1 \ge x_1 \ge 0 \\ x_1 + x_3 \le 1, & 1 \ge x_2 \ge 0 \\ x_2 + x_3 \le 1, & 1 \ge x_3 \ge 0 \end{array} \right\}$$

$$P^3 = \left\{\ x \in \mathbb{R}^3\ : x_1 + x_2 + x_3 \le 1,\ 1 \ge x_i \ge 0\ \forall i\right\}$$

Claim 2: $P^2 \supsetneq P^3$.

Proof. If $\bar{x} \in P^3$ then $\bar{x}_i \ge 0\ \forall i$ and

$$\bar{x}_1 + \bar{x}_2 + \bar{x}_3 \le 1 \implies \bar{x}_i + \bar{x}_j \le 1\ \forall i,j \implies \bar{x} \in P^2$$

To complete the proof:

Also notice that $x'' = (1/2, 1/2, 1/2) \in P^2$ but $x'' \notin P^3$ $\square$

Comparing formulations

Now lets look at the polyhedra (relaxations) defining them:

$$P^1 = \left\{ \; x \in \mathbb{R}^3 : 3x_1 + 4x_2 + 5x_3 \leq 6, \; 1 \geq x_i \geq 0 \; \forall i \right\}$$

$$P^2 = \left\{ \; x \in \mathbb{R}^3 : \begin{array}{ll} x_1 + x_2 \leq 1, & 1 \geq x_1 \geq 0 \\ x_1 + x_3 \leq 1, & 1 \geq x_2 \geq 0 \\ x_2 + x_3 \leq 1, & 1 \geq x_3 \geq 0 \end{array} \right\}$$

$$P^3 = \left\{ \; x \in \mathbb{R}^3 \; : x_1 + x_2 + x_3 \leq 1, \; 1 \geq x_i \geq 0 \; \forall i \right\}$$

Claim 2: $P^2 \supsetneq P^3$.

Proof. If $\bar{x} \in P^3$ then $\bar{x}_i \geq 0 \; \forall i$ and

$$\bar{x}_1 + \bar{x}_2 + \bar{x}_3 \leq 1 \implies \bar{x}_i + \bar{x}_j \leq 1 \; \forall i, j \implies \bar{x} \in P^2$$

To complete the proof:

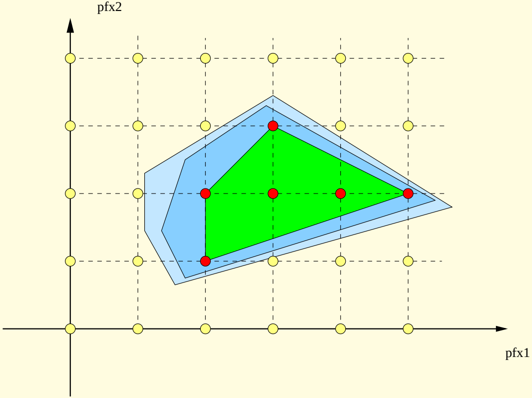
Also notice that $x'' = (1/2, 1/2, 1/2) \in P^2$ but $x'' \notin P^3$ $\square$

Geometrically

There are different ways to formulate the same integer program:

$$P^1 \supsetneq P^2 \supsetneq P^3 \quad \text{where as} \quad K^1 = K^2 = K^3$$

where $K^i = P^i \cap \mathbb{Z}^n$ for $i = 1, 2, 3$



Two formulations of the same IP problem

$$
(\mathsf{IP}_1) \qquad\qquad (\mathsf{IP}_2)
$$

$$
\left.\begin{aligned}
z^{\mathsf{IP1}} = \max \quad & 7x_1 + 8x_2 + 9x_3 \\
\text{s.t.} \quad & x_1 + x_2 \le 1 \\
& x_1 + x_3 \le 1 \\
& x_2 + x_3 \le 1 \\
& x_1, x_2, x_3 \in \{0,1\}
\end{aligned}\right\} \equiv \left\{\begin{aligned}
z^{\mathsf{IP2}} = \max \quad & 7x_1 + 8x_2 + 9x_3 \\
\text{s.t.} \quad & x_1 + x_2 + x_3 \le 1 \\
& x_1, x_2, x_3 \in \{0,1\}
\end{aligned}\right.
$$

Optimal solutions to LP relaxations:

LP_1: $x^{\mathsf{LP1}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, $\quad z^{\mathsf{LP1}} = 12 \implies 12 \ge z^{\mathsf{IP}_1} = z^{\mathsf{IP2}}$

LP_2: $x^{\mathsf{LP2}} = (0, 0, 1)$, $\quad z^{\mathsf{LP2}} = 9 \implies 9 \ge z^{\mathsf{IP1}} = z^{\mathsf{IP2}}$

x^{LP2} is **feasible** for IP_1 and IP_2! $\implies z^{\mathsf{IP1}} = z^{\mathsf{IP2}} \ge 9$

$\Rightarrow$ $z^{\mathsf{IP1}} = z^{\mathsf{IP2}} = 9$, and IP_2 is a better model than IP_1 as

$$P_2 \subset P_1$$

Two formulations of the same IP problem

$$
(\mathsf{IP}_1) \qquad\qquad\qquad\qquad (\mathsf{IP}_2)
$$

$$
\left.\begin{aligned}
z^{\mathsf{IP1}} = \max \quad & 7x_1 + 8x_2 + 9x_3 \\
\text{s.t.} \quad & x_1 + x_2 \le 1 \\
& x_1 + x_3 \le 1 \\
& x_2 + x_3 \le 1 \\
& x_1, x_2, x_3 \in \{0,1\}
\end{aligned}\right\} \equiv \left\{\begin{aligned}
z^{\mathsf{IP2}} = \max \quad & 7x_1 + 8x_2 + 9x_3 \\
\text{s.t.} \quad & x_1 + x_2 + x_3 \le 1 \\
& x_1, x_2, x_3 \in \{0,1\}
\end{aligned}\right.
$$

Optimal solutions to LP relaxations:

LP_1: $x^{\mathsf{LP1}} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}), \quad z^{\mathsf{LP1}} = 12 \implies 12 \ge z^{\mathsf{IP}_1} = z^{\mathsf{IP2}}$

LP_2: $x^{\mathsf{LP2}} = (0,0,1), \quad z^{\mathsf{LP2}} = 9 \implies 9 \ge z^{\mathsf{IP1}} = z^{\mathsf{IP2}}$

x^{LP2} is **feasible** for IP_1 and IP_2! $\implies z^{\mathsf{IP1}} = z^{\mathsf{IP2}} \ge 9$

$\Rightarrow$ $z^{\mathsf{IP1}} = z^{\mathsf{IP2}} = 9$, and IP_2 is a better model than IP_1 as

$$P_2 \subset P_1$$

Modeling with Binary Variables

Binary variables

- Help model yes/no decisions: $x_i \in \{0, 1\}$
  - $x_i = 0$ if the decision is "no",
  - $x_i = 1$ if it is "yes"
- They can also model logical operators. When 3 cousins go out to have ice cream, if their grandma says:
  - Mario or Luigi can have ice cream, but **not both**:

    $$x_{\text{Mario}} + x_{\text{Luigi}} \leq 1$$

  - **At least one** among Mario and Luigi will have ice cream:

    $$x_{\text{Mario}} + x_{\text{Luigi}} \geq 1$$

  - **If** Mario has ice cream, **then** Giovanni will have one too:

    $$x_{\text{Mario}} \leq x_{\text{Giovanni}}$$

  - Luigi gets ice cream **if and only if** Giovanni does not get any:

    $$x_{\text{Luigi}} = 1 - x_{\text{Giovanni}}$$

Facility Location Problem

Uncapacitated Facility Location Problem

- A set J of retailers has to be served by a set of plants, yet to be built.
- There is a set I of potential sites for plants, and there is
  - a cost f_i for building plant $i \in I$
  - a (transportation) cost c_{ij} from plant $i \in I$ to retailer $j \in J$
- Each retailer $j \in J$ will be served by exactly one plant $i \in I$.
- Choose a subset S of I such that the total cost is minimized.

Variables:

- $x_i \in \{0,1\}$ for $i \in I$ indicates if plant is built

$$x_i = \begin{cases} 1 & \text{if plant } i \text{ is built} \\ 0 & \text{otherwise} \end{cases}$$

- $y_{ij} \in \{0,1\}$ indicates if retailer $j \in J$ is assigned to $i \in I$:

$$y_{ij} = \begin{cases} 1 & \text{if } i \text{ serves retailer } j \\ 0 & \text{otherwise} \end{cases}$$

Uncapacitated Facility Location Problem

- A set J of retailers has to be served by a set of plants, yet to be built.
- There is a set I of potential sites for plants, and there is
  - a cost f_i for building plant $i \in I$
  - a (transportation) cost c_{ij} from plant $i \in I$ to retailer $j \in J$
- Each retailer $j \in J$ will be served by exactly one plant $i \in I$.
- Choose a subset S of I such that the total cost is minimized.

Variables:

- $x_i \in \{0, 1\}$ for $i \in I$ indicates if plant is built

$$x_i = \begin{cases} 1 & \text{if plant } i \text{ is built} \\ 0 & \text{otherwise} \end{cases}$$

- $y_{ij} \in \{0, 1\}$ indicates if retailer $j \in J$ is assigned to $i \in I$:

$$y_{ij} = \begin{cases} 1 & \text{if } i \text{ serves retailer } j \\ 0 & \text{otherwise} \end{cases}$$

Uncapacitated Facility Location Problem

Objective function:

$$\sum_{i \in I} f_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}$$

Constraints:

- Each retailer $j \in J$, should be assigned to one plant:

$$\sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J$$

- A retailer $j \in J$ can be assigned to plant $i \in I$ only if it is built:

$$y_{ij} \leq x_i \quad \forall i \in I, j \in J$$

or, alternatively:

$$\sum_{j \in J} y_{ij} \leq |J| x_i \qquad \forall i \in I$$

Uncapacitated Facility Location Problem

Objective function:

$$\sum_{i \in I} f_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}$$

Constraints:

- Each retailer $j \in J$, should be assigned to one plant:

$$\sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J$$

- A retailer $j \in J$ can be assigned to plant $i \in I$ only if it is built:

$$y_{ij} \leq x_i \quad \forall i \in I, j \in J$$

or, alternatively:

$$\sum_{j \in J} y_{ij} \leq |J| x_i \qquad \forall i \in I$$

Two correct formulations for the facility location problem

$$\left.\begin{array}{ll} & \text{IP}_1 \\ \min & \sum_{i\in I} f_i x_i + \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij} \\ \text{s.t.} & \sum_{i\in I} y_{ij} = 1 \qquad \forall j \in J \\ & y_{ij} \le x_i \quad \forall i \in I, j \in J \\ & x_i, y_{ij} \in \{0,1\} \end{array}\right\} \equiv \left\{\begin{array}{ll} & \text{IP}_2 \\ \min & \sum_{i\in I} f_i x_i + \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij} \\ \text{s.t.} & \sum_{i\in I} y_{ij} = 1 \qquad \forall j \in J \\ & \sum_{j\in J} y_{ij} \le |J| x_i \quad \forall i \in I \\ & x_i, y_{ij} \in \{0,1\} \end{array}\right.$$

- Which one to use?
- Consider a feasible solution $(\bar{x}, \bar{y})$ to LP$_1$. for each $i \in I$, we have

$$\bar{y}_{ij} \le \bar{x}_i, \ \forall j \in J \implies \sum_{j\in J} \bar{y}_{ij} \le |J|\bar{x}_i$$

  Implying that, $(\bar{x}, \bar{y})$ is a feasible solution to LP$_2$. Therefore $P_1 \subset P_2$
- When $|I| = 150$, $|J| = 50$, IP$_2$ takes an hour, IP$_1$ takes seconds to solve.

Two correct formulations for the facility location problem

$$\left.\begin{array}{ll} & \mathsf{IP}_1 \\ \min & \sum_{i\in I} f_i x_i + \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij} \\ \text{s.t.} & \sum_{i\in I} y_{ij} = 1 \qquad \forall j \in J \\ & y_{ij} \le x_i \quad \forall i \in I, j \in J \\ & x_i, y_{ij} \in \{0,1\} \end{array}\right\} \equiv \left\{\begin{array}{ll} & \mathsf{IP}_2 \\ \min & \sum_{i\in I} f_i x_i + \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij} \\ \text{s.t.} & \sum_{i\in I} y_{ij} = 1 \qquad \forall j \in J \\ & \sum_{j\in J} y_{ij} \le |J| x_i \quad \forall i \in I \\ & x_i, y_{ij} \in \{0,1\} \end{array}\right.$$

- Which one to use?
- Consider a feasible solution $(\bar{x}, \bar{y})$ to LP_1. for each $i \in I$, we have

$$\bar{y}_{ij} \le \bar{x}_i,\ \forall j \in J \implies \sum_{j\in J} \bar{y}_{ij} \le |J|\bar{x}_i$$

  Implying that, $(\bar{x}, \bar{y})$ is a feasible solution to LP_2. Therefore $P_1 \subset P_2$

- When $|I| = 150$, $|J| = 50$, IP_2 takes an hour, IP_1 takes seconds to solve.

Two correct formulations for the facility location problem

$$
\left.
\begin{array}{ll}
& \mathsf{IP}_1 \\
\min & \displaystyle\sum_{i \in I} f_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} \\
\text{s.t.} & \displaystyle\sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J \\
& y_{ij} \leq x_i \quad \forall i \in I, j \in J \\
& x_i, y_{ij} \in \{0,1\}
\end{array}
\right\}
\equiv
\left\{
\begin{array}{ll}
& \mathsf{IP}_2 \\
\min & \displaystyle\sum_{i \in I} f_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} \\
\text{s.t.} & \displaystyle\sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J \\
& \displaystyle\sum_{j \in J} y_{ij} \leq |J| x_i \quad \forall i \in I \\
& x_i, y_{ij} \in \{0,1\}
\end{array}
\right.
$$

- Which one to use?
- Consider a feasible solution $(\bar{x}, \bar{y})$ to LP_1. for each $i \in I$, we have

$$\bar{y}_{ij} \leq \bar{x}_i, \ \forall j \in J \implies \sum_{j \in J} \bar{y}_{ij} \leq |J| \bar{x}_i$$

  Implying that, $(\bar{x}, \bar{y})$ is a feasible solution to LP_2. Therefore $P_1 \subset P_2$
- When $|I| = 150$, $|J| = 50$, IP_2 takes an hour, IP_1 takes seconds to solve.

Size of formulations: Counting non-zero entries of constraint matrix

IP$_1$

Non-zeroes:

$$\min \quad \sum_{i \in I} f_i x_i + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}$$

$\longleftarrow$ objective has $|I| + |I| \cdot |J|$ nonzeroes (not important)

$$\text{s.t.} \quad \sum_{i \in I} y_{ij} = 1 \qquad \forall j \in J$$

$\longleftarrow$ Each one has $|I|$ nonzeroes $\implies |I| \cdot |J|$

$$y_{ij} \le x_i \quad \forall i \in I, j \in J$$

$\longleftarrow$ Each one has 2 nonzeroes $\implies 2|I| \cdot |J|$

$$x_i \,, y_{ij} \in \{0,1\} \; \forall i \in I, j \in J$$

$\longleftarrow$ There are $|I|$ many x variables and $|I| \cdot |J|$ many y variables.

- When $|I| = 150$, $|J| = 50$, the nonzero entries of the constraint matrix is

$$3|I| \cdot |J| = 3 \cdot 150 \cdot 50 = 22,500$$

- It has $|I| + |I| \cdot |J| = 150 + 150 \cdot 50 = 7600$ binary variables
- Generally, formulations with fewer nonzeroes and fewer integer variables solve faster but the strength of the LP formulation matters most.

Size of formulations: Counting non-zero entries of constraint matrix

IP$_1$	**Non-zeroes**:
$\min \quad \sum_{i\in I} f_i x_i + \sum_{i\in I}\sum_{j\in J} c_{ij} y_{ij}$	$\longleftarrow$ objective has $\|I\| + \|I\|\cdot\|J\|$ nonzeroes (not important)
$\text{s.t.} \quad \sum_{i\in I} y_{ij} = 1 \qquad \forall j \in J$	$\longleftarrow$ Each one has $\|I\|$ nonzeroes $\implies \|I\|\cdot\|J\|$
$y_{ij} \le x_i \quad \forall i\in I, j\in J$	$\longleftarrow$ Each one has 2 nonzeroes $\implies 2\|I\|\cdot\|J\|$
$x_i\ , y_{ij} \in \{0,1\}\ \forall i \in I, j\in J$	$\longleftarrow$ There are $\|I\|$ many x variables and $\|I\|\cdot\|J\|$ many y variables.

- When $|I| = 150$, $|J| = 50$, the nonzero entries of the constraint matrix is

$$3|I|\cdot|J| = 3\cdot 150\cdot 50 = 22,500$$

- It has $|I| + |I|\cdot|J| = 150 + 150\cdot 50 = 7600$ binary variables
- Generally, formulations with fewer nonzeroes and fewer integer variables solve faster but the strength of the LP formulation matters most.

A Detour to Machine Learning and a Clustering Problem

Supervised vs. unsupervised learning tasks

- In **supervised** ML each data point $x = [x_1, x_2, \ldots]$ has a label(s) y and the ML task is to guess y correctly. i.e. Find a function f such that

$$f(x) \approx y$$

  - **Classification**: if y takes discrete values i.e. $y \in \{good, bad, ugly\}$
  - **Regression**: if y takes continuous values i.e. $y \in [0, 8] \quad$ (height)

- In **unsupervised** ML there are no labels and goal is to infer a structure present within the data points. Example:

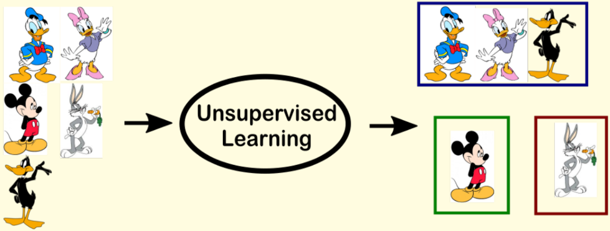
Supervised vs. unsupervised learning tasks

- In **supervised** ML each data point $x = [x_1, x_2, \ldots]$ has a label(s) y and the ML task is to guess y correctly. i.e. Find a function f such that

$$f(x) \approx y$$

  - **Classification**: if y takes discrete values i.e. $y \in \{good, bad, ugly\}$
  - **Regression**: if y takes continuous values i.e. $y \in [0, 8]$ (height)

- In **unsupervised** ML there are no labels and goal is to infer a structure present within the data points. Example:



An unsupervised ML task: Clustering

- Clustering is an ML problem that involves the grouping of data points.
  - Data points in the same cluster should be "similar"
  - Data points in different clusters should be "dissimilar".
- Each data point is represented by its features $x \in \mathbb{R}^n$
- Given points $x^i, x^j \in \mathbb{R}^n$, how to measure similarity?
  - For example we can use the Euclidean distance between the points

$$d(i,j) = \left\| x^i - x^j \right\|_2 = \sqrt{\sum_{k=1}^{n} (x_k^i - x_k^j)^2}$$

  - Points with smaller distance can be considered more similar.
- Depending on the application there might be different ways to measure similarity (distance).
- Given a collection of points and distances between them we can now define the clustering problem formally.

An unsupervised ML task: Clustering

- Clustering is an ML problem that involves the grouping of data points.
  - Data points in the same cluster should be "similar"
  - Data points in different clusters should be "dissimilar".
- Each data point is represented by its features $x \in \mathbb{R}^n$
- Given points $x^i, x^j \in \mathbb{R}^n$, how to measure similarity?
  - For example we can use the Euclidean distance between the points

$$d(i,j) = \left\|x^i - x^j\right\|_2 = \sqrt{\sum_{k=1}^{n}(x_k^i - x_k^j)^2}$$

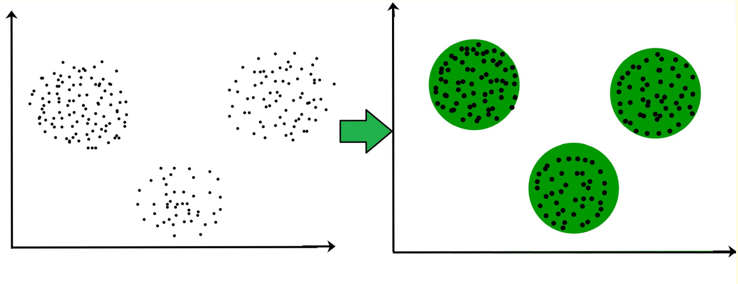
  - Points with smaller distance can be considered more similar.
- Depending on the application there might be different ways to measure similarity (distance).
- Given a collection of points and distances between them we can now define the clustering problem formally.

Clustering problem:

Given an integer $k > 1$ and a collection of points $X = \{x^1, x^2, \dots\}$ together with distances between pairs of these points. Partition the points in X into k clusters $C_1, \dots, C_k$, such that minimum distance between pairs of points in different clusters:

$$\min_{i \in C_p,\, j \in C_q,\, p \neq q} d(i,j)$$

is maximized.



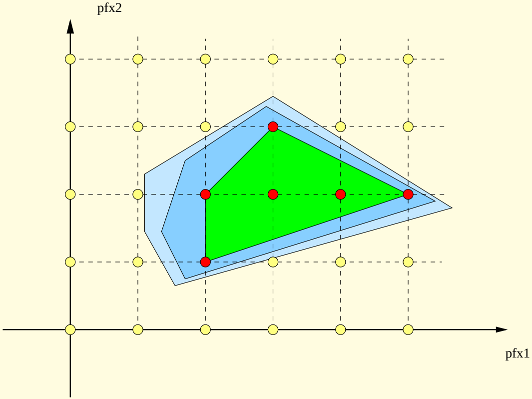
Optional Weekend Homework :)

- Winnie the Pooh has 9 pots with honey whose weights are 1, 2, ..., 9 lbs.
- Each pot has its weight written on it.
- In addition, someone put a small (1 oz) piece of cheese into one of the pots.

  Question: How can Winnie, in two weighings on a balance scale (with no additional weights), find out which pot contains the cheese?

(2/5/2024)

Recap: Formulating IPs



There are different ways to formulate the same integer program:

$$P^1 \supsetneqq P^2 \supsetneqq P^3 \text{ where as } K^1 = K^2 = K^3$$

where $K^i = P^i \cap \mathbb{Z}^n$ for $i = 1, 2, 3$

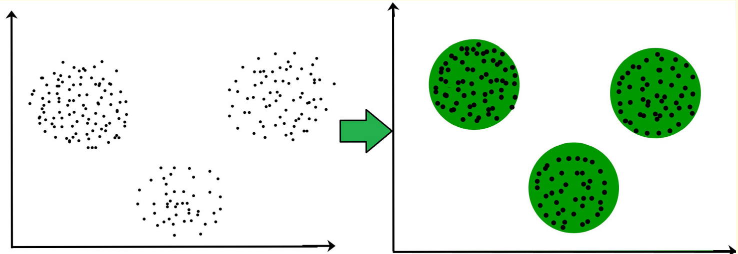
Recap: Clustering problem

Given: An integer $k > 1$ and a collection of points $X = \{x^1, x^2, \ldots\}$ together with distances between pairs of these points.

Goal: Partition X into k clusters $C_1, \ldots, C_k$, such that minimum distance between pairs of points in different clusters:

$$\min_{i \in C_p,\, j \in C_q,\, p \neq q} d(i, j)$$

is maximized. ($d(i, j)$ measures the distance between points i and j)



Lloyd's algorithm: K-Means clustering

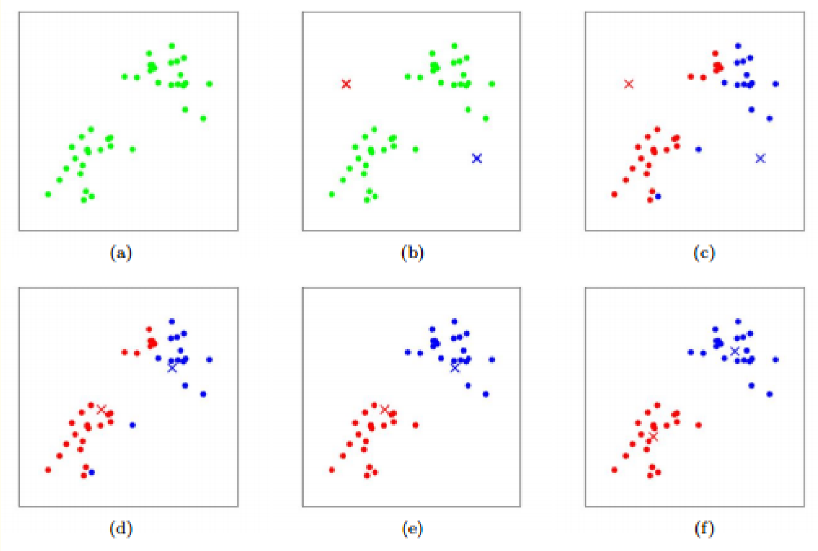
Clustering problem: Partition X into clusters $C_1, \ldots, C_k$ so as to maximize minimum distance between clusters :

$$d^* = \max_{C_1 \ldots, C_k \text{ is a partition}} \left(\min_{i \in C_p,\, j \in C_q,\, p \neq q} d(i,j) \right)$$

- K-Means is the most popular clustering algorithm (it is a heuristic).
  1. Randomly pick k seed points (one for each cluster).
  2. Assign points to the closest seed to form the clusters.
  3. Change the seed points to a "central" point in each cluster
  4. Repeat until clusters do not change much.
  5. Return the best solution found during the search
- Easy to understand and implement.
- It is a good heuristic for the clustering problem (practical performance).

K-Means example

Data points Initial seeds A clustering



Clustering when $d(i, j) \in \{0, 1\}$

(points are either similar or dissimilar)

Clustering Problem

Consider the following clustering problem:

- There are n objects $N = \{1, \ldots, n\}$.
- Any pair of objects $i, j \in N$ is either similar or dissimilar

  [$d(i, j)$ is either 0 (if similar) or 1 (if dissimilar)]
- We are given a set D that contains pairs of dissimilar objects (the other pairs are similar)
- We want to cluster the objects in exactly k clusters so that each cluster $C_1, \ldots, C_k$ consists of items that are mostly similar to each other. $K = \{1, \ldots, k\}$
- In addition, each cluster must contain at least ℓ objects.

- Model this as an IP where we the objective is to minimizing the total number of pairs of dissimilar objects put in the same cluster.

Clustering problem

Input:

- n objects numbered $1, 2, \ldots, n$
- Desired number of clusters k, and a lower bound ℓ on the number of objects in a cluster
- A set D of pairs of dissimilar objects

  (i,e. $\{i, j\} \in D$ means that objects i and j are dissimilar)

Output:

- A partitioning of the objects into cluster $C_1, C_2, \ldots, C_k$

  Partitioning means:

  (i) $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, 2, \ldots, n\}$, and

  (ii) $C_s \cap C_t = \varnothing$ for all $s \neq t$.

Goal:

- Minimize the total number of pairs $\{i, j\}$ where i and j are clustered in the same cluster, but are dissimilar (meaning, $\{i, j\} \in D$)

Clustering Problem

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(only for } i < j\text{)}$$

IP Formulation:

$$\begin{array}{llll} \min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & \longleftarrow & \text{dissimilar pairs in the same cluster} \\ \text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\ & \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\ & x_{ijs} \in \{0,1\} & \forall i < j \in N, \forall s \in K & \\ & y_{is} \in \{0,1\} & \forall i \in N, \forall s \in K & \end{array}$$

Clustering Problem

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(only for } i < j\text{)}$$

IP Formulation:

$$\min \quad \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} \quad \longleftarrow \quad \text{dissimilar pairs in the same cluster}$$

$$\text{s.t.} \quad \sum_{s \in K} y_{is} = 1 \qquad \forall i \in N \quad \longleftarrow \quad \text{objects}$$

$$\sum_{i \in N} y_{is} \geq \ell \qquad \forall s \in K \quad \longleftarrow \quad \text{clusters}$$

$$x_{ijs} \in \{0,1\} \qquad \forall i < j \in N, \forall s \in K$$

$$y_{is} \in \{0,1\} \qquad \forall i \in N, \forall s \in K$$

Clustering Problem

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

(only for $i < j$)

IP Formulation:

$$\min \quad \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} \quad \longleftarrow \quad \text{dissimilar pairs in the same cluster}$$

$$\text{s.t.} \quad \sum_{s \in K} y_{is} = 1 \qquad \forall i \in N \quad \longleftarrow \quad \text{objects}$$

$$\sum_{i \in N} y_{is} \geq \ell \qquad \forall s \in K \quad \longleftarrow \quad \text{clusters}$$

How do we say: $x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$

$$x_{ijs} \in \{0, 1\} \qquad \forall i < j \in N \ , \forall s \in K$$

$$y_{is} \in \{0, 1\} \qquad \forall i \in N \ , \forall s \in K$$

- How do we say:

$$x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

using the fact that x and y take $\{0,1\}$ values?

- First idea: We can write

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

and

$$x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})$$

  - When both $y_{is} = 1$ and $y_{js} = 1$ the we have

$$x_{ijs} \geq 1 + 1 - 1 = 1 \quad \text{and} \quad x_{ijs} \leq \tfrac{1}{2}(1+1) = 1 \implies x_{ijs} = 1$$

  - If not, then we must have $y_{is} + y_{js} \leq 1$ and

$$\underbrace{x_{ijs} \geq y_{is} + y_{js} - 1}_{x_{ijs} \geq 0} \quad \text{and} \quad \underbrace{x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})}_{x_{ijs} \leq 1/2} \implies x_{ijs} = 0$$

  - It works! (because $x_{ijs} \in \{0,1\}$)

- How do we say:

$$x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

  using the fact that x and y take $\{0,1\}$ values?

- First idea: We can write

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

  and

$$x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})$$

  – When both $y_{is} = 1$ and $y_{js} = 1$ the we have

$$x_{ijs} \geq 1 + 1 - 1 = 1 \quad \text{and} \quad x_{ijs} \leq \frac{1}{2}(1+1) = 1 \implies x_{ijs} = 1$$

  – If not, then we must have $y_{is} + y_{js} \leq 1$ and

$$\underbrace{x_{ijs} \geq y_{is} + y_{js} - 1}_{x_{ijs} \geq 0} \quad \text{and} \quad \underbrace{x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})}_{x_{ijs} \leq 1/2} \implies x_{ijs} = 0$$

  – It works! (because $x_{ijs} \in \{0,1\}$)

- How do we say:

$$x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

using the fact that x and y take $\{0, 1\}$ values?

- First idea: We can write

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

and

$$x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})$$

  - When both $y_{is} = 1$ and $y_{js} = 1$ the we have

$$x_{ijs} \geq 1 + 1 - 1 = 1 \quad \text{and} \quad x_{ijs} \leq \tfrac{1}{2}(1 + 1) = 1 \implies x_{ijs} = 1$$

  - If not, then we must have $y_{is} + y_{js} \leq 1$ and

$$\underbrace{x_{ijs} \geq y_{is} + y_{js} - 1}_{x_{ijs} \geq 0} \quad \text{and} \quad \underbrace{x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})}_{x_{ijs} \leq 1/2} \implies x_{ijs} = 0$$

  - It works! (because $x_{ijs} \in \{0, 1\}$)

Clustering Problem: Formulation 0

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(only for } i < j\text{)}$$

IP Formulation:

$$\begin{array}{lrll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \,, s \in K & \\
& x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js}) & \forall i < j \in N \,, s \in K & \\
& x_{ijs} \in \{0, 1\} & \forall i < j \in N \,, \forall s \in K & \\
& y_{is} \in \{0, 1\} & \forall i \in N \,, \forall s \in K &
\end{array}$$

- We want to say:

$$x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

  using the fact that x and y take $\{0, 1\}$ values?

- This is same as

$$x_{ijs} = y_{is} y_{js}$$

- But we cannot write this in a linear integer program.

- So instead we wrote 2 linear inequalities

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

  and

$$x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})$$

- Question: Can we do better?

Taking a step back: Multiplying binary variables

- Let $y_1 \in \{0, 1\}$ and $y_2 \in \{0, 1\}$ be two binary variables.
- Assume we are interested in their product $y_1 \cdot y_2$.
- How can we express their product $x = y_1 y_2$ using linear inequalities?

$$x = \begin{cases} 1 & \text{if } y_1 = 1 \text{ and } y_2 = 1 \\ 0 & \text{otherwise.} \end{cases}$$

- Consider the following inequalities:

$$\begin{aligned} x &\leq y_1 \\ x &\leq y_2 \\ x &\geq 0 \\ x &\geq y_1 + y_2 - 1 \end{aligned}$$

Claim

If $y_1, y_2 \in \{0, 1\}$ and x satisfies the constraints above, then

$$x = y_1 y_2$$

Claim

If x, y_1, y_2 satisfy the McCormick constraints

$$
\begin{aligned}
x &\le y_1, \\
x &\le y_2, \\
x &\ge y_1 + y_2 - 1, \\
x &\ge 0,
\end{aligned}
$$

and

$$y_1, y_2 \in \{0, 1\}$$

then $x = y_1 y_2$.

Proof :

$\boldsymbol{y}_1$	$\boldsymbol{y}_2$	constraints		$\boldsymbol{x}$
0	0	$x \le y_1,$	$x \ge 0$	0
0	1	$x \le y_1,$	$x \ge 0$	0
1	0	$x \le y_2,$	$x \ge 0$	0
1	1	$x \ge y_1 + y_2 - 1,$	$x \le y_1$	1

Back to Formulation 0 for the Clustering Problem

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases}$$

(Notice that we want x_{ijs} variable to be equal to $y_{is} \cdot y_{js}$)

IP Formulation:

$$\begin{array}{lrll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N, s \in K & \\
& x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js}) & \forall i < j \in N, s \in K & \\
& x_{ijs} \in \{0,1\} & \forall i < j \in N, \forall s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N, \forall s \in K &
\end{array}$$

Clustering Problem: Formulation 1

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases}$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N\,, s \in K & \\
& x_{ijs} \leq y_{is} & \forall i < j \in N\,, s \in K & \\
& x_{ijs} \leq y_{js} & \forall i < j \in N\,, s \in K & \\
& x_{ijs} \in \{0,1\} & \forall i < j \in N\,, s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N\,, s \in K &
\end{array}$$

Comparing formulations 0 and 1

- Size of the formulation for $n = 40$ objects and $k = 3$ clusters:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620

  – y_{is} variables: $40 \times 3 = 120$

  – x_{ijs} variables: $\binom{40}{2} \times 3 = 780 \times 3 = 2340$

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds

  – Form. 1 needs much fewer nodes to solve the IP.

  – Surprisingly, Form. 1 is $\approx$ 20% faster per B&B node (# of LPs)

  – In both formulations x and y variables are declared binary

  – Form.0 has 2 constraints for each x_{ijs} variable, Form.1 has 3.

Comparing formulations 0 and 1

- Size of the formulation for $n = 40$ objects and $k = 3$ clusters:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620

  - y_{is} variables: $40 \times 3 = 120$
  - x_{ijs} variables: $\binom{40}{2} \times 3 = 780 \times 3 = 2340$

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds

  - Form. 1 needs much fewer nodes to solve the IP.
  - Surprisingly, Form. 1 is ≈ 20% faster per B&B node (# of LPs)
  - In both formulations x and y variables are declared binary
  - Form.0 has 2 constraints for each x_{ijs} variable, Form.1 has 3.

Why fewer nodes?

LP relaxation of these two formulations look like:

$$
\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \text{ objects} \\
 & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \text{ clusters} \\
 & x_{ijs} \text{ constraints} & \forall i < j \in N\ , s \in K & \\
 & 1 \geq x_{ijs} \geq 0 & 1 \geq y_{is} \geq 0 &
\end{array}
$$

Formulation 0:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

$$x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})$$

Formulation 1:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

$$x_{ijs} \leq y_{is}$$

$$x_{ijs} \leq y_{js}$$

- Now consider a feasible solution to the LP relaxation of F1.

$$\underbrace{(x_{ijs} \leq y_{is}) \quad \text{AND} \quad (x_{ijs} \leq y_{is})}_{\text{solution feasible for F1}} \quad \Rightarrow \quad \underbrace{\left(x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})\right)}_{\text{also feasible for F0}}$$

- Therefore, Formulation 1 is better as its LP feasible region is smaller.

Why fewer nodes?

LP relaxation of these two formulations look like:

$$\begin{array}{lrll} \min & \sum_{\{i,j\}\in D}\sum_{s\in K} x_{ijs} & & \\ \text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \text{ objects} \\ & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \text{ clusters} \\ & x_{ijs} \text{ constraints} & \forall i < j \in N\ , s \in K & \\ & 1 \geq x_{ijs} \geq 0 & 1 \geq y_{is} \geq 0 & \end{array}$$

Formulation 0:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$
$$x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})$$

Formulation 1:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$
$$x_{ijs} \leq y_{is}$$
$$x_{ijs} \leq y_{js}$$

- Now consider a feasible solution to the LP relaxation of F1.

$$\underbrace{(x_{ijs} \leq y_{is}) \quad \text{AND} \quad (x_{ijs} \leq y_{is})}_{\text{solution feasible for F1}} \quad \Rightarrow \quad \underbrace{(x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})}_{\text{also feasible for F0}}$$

- Therefore, Formulation 1 is better as its LP feasible region is smaller.

Why fewer nodes?

LP relaxation of these two formulations look like:

$$
\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
 & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
 & x_{ijs} \text{ constraints} & \forall i < j \in N, s \in K & \\
 & 1 \geq x_{ijs} \geq 0 \quad 1 \geq y_{is} \geq 0 & &
\end{array}
$$

Formulation 0:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

$$x_{ijs} \leq \frac{1}{2}(y_{is} + y_{js})$$

Formulation 1:

$$x_{ijs} \geq y_{is} + y_{js} - 1$$

$$x_{ijs} \leq y_{is}$$

$$x_{ijs} \leq y_{js}$$

- Now consider a feasible solution to the LP relaxation of F1.

$$\underbrace{(x_{ijs} \leq y_{is}) \quad \text{AND} \quad (x_{ijs} \leq y_{is})}_{\text{solution feasible for F1}} \quad \Rightarrow \quad \underbrace{(x_{ijs} \leq \tfrac{1}{2}(y_{is} + y_{js})}_{\text{also feasible for F0}}$$

- Therefore, Formulation 1 is better as its LP feasible region is smaller.

Observation 0: If x, y_1, y_2 satisfy the constraints

$$x \geq y_1 + y_2 - 1$$

$$x \leq \frac{1}{2}(y_1 + y_2)$$

$$y_1, y_2 \in \{0,1\} \text{ and } x \in \{0,1\}$$

then $x = y_1 y_2$. (i.e., $x = 1$ only when both $y_1 = 1$ and $y_2 = 1$)

Without $x \in \{0,1\}$, the point $\underbrace{(1, 0, \frac{1}{2})}_{(y_1, y_2, x)}$ is feasible to the system above.

Observation 1: If x, y_1, y_2 satisfy the constraints

$$x \leq y_1, \quad x \leq y_2,$$

$$x \geq 0, \quad x \geq y_1 + y_2 - 1,$$

$$y_1, y_2 \in \{0,1\}$$

then $x = y_1 y_2$. (Notice that x variable is not declared to be binary)

Observation 0: If x, y_1, y_2 satisfy the constraints

$$x \geq y_1 + y_2 - 1$$

$$x \leq \frac{1}{2}(y_1 + y_2)$$

$$y_1, y_2 \in \{0,1\} \text{ and } x \in \{0,1\}$$

then $x = y_1 y_2$. (i.e., $x = 1$ only when both $y_1 = 1$ and $y_2 = 1$)

Without $x \in \{0,1\}$, the point $\underbrace{(1, 0, \frac{1}{2})}_{(y_1, y_2, x)}$ is feasible to the system above.

Observation 1: If x, y_1, y_2 satisfy the constraints

$$x \leq y_1, \quad x \leq y_2,$$

$$x \geq 0, \quad x \geq y_1 + y_2 - 1,$$

$$y_1, y_2 \in \{0,1\}$$

then $x = y_1 y_2$. (Notice that x variable is not declared to be binary)

Clustering Problem: Formulation 1^+

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases}$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \ , s \in K & \\
& x_{ijs} \leq y_{is}, \ x_{ijs} \leq y_{js} & \forall i < j \in N \ , s \in K & \\
& \cancel{x_{ijs} \in \{0,1\}} \ \ 1 \geq x_{ijs} \geq 0 & \forall i < j \in N \ , s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N \ , s \in K &
\end{array}$$

Comparing formulations 0, 1 and 1^+

- Size of the formulation for $n = 40$ objects and $k = 3$ clusters:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620

  – y_{is} variables: $40 \times 3 = 120$ $\longleftarrow$ binary
  – x_{ijs} variables: $\binom{40}{2} \times 3 = 780 \times 3 = 2340$ $\longleftarrow$ continuous in F 1^+

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds

  – Formulation 1^+ is 5x faster than Formulation 1, that's pretty good!

Comparing formulations 0, 1 and 1^+

- Size of the formulation for $n = 40$ objects and $k = 3$ clusters:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620

  - y_{is} variables: $40 \times 3 = 120$ $\longleftarrow$ binary
  - x_{ijs} variables: $\binom{40}{2} \times 3 = 780 \times 3 = 2340$ $\longleftarrow$ continuous in F 1^+

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds

  - Formulation 1^+ is 5x faster than Formulation 1, that's pretty good!

Lets look at Formulation 1^+ again

(y_{is} : item i is in cluster C_s, x_{ijs} : both i and j are in cluster C_s)

Formulation 1^+ :

$$
\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \ , s \in K & \\
& x_{ijs} \leq y_{is}, \ x_{ijs} \leq y_{js} & \forall i < j \in N \ , s \in K & \\
& \cancel{x_{ijs} \in \{0,1\}} \ \ 1 \geq x_{ijs} \geq 0 & \forall i < j \in N \ , s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N \ , s \in K &
\end{array}
$$

Question: Even if we did not have the constraints $x_{ijs} \leq y_{is}, \ x_{ijs} \leq y_{js}$, would $x_{ijs} = 1$ in an optimal sol. if either $y_{is} = 0$ or $y_{js} = 0$?

Lets look at Formulation 1^+ again

(y_{is} : item i is in cluster C_s, x_{ijs} : both i and j are in cluster C_s)

Formulation 1^+ :

$$
\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \text{ objects} \\
 & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \text{ clusters} \\
 & x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N\ , s \in K & \\
 & x_{ijs} \leq y_{is},\ x_{ijs} \leq y_{js} & \forall i < j \in N\ , s \in K & \\
 & \cancel{x_{ijs} \in \{0,1\}}\ \ 1 \geq x_{ijs} \geq 0 & \forall i < j \in N\ , s \in K & \\
 & y_{is} \in \{0,1\} & \forall i \in N\ , s \in K &
\end{array}
$$

Question: Even if we did not have the constraints $x_{ijs} \leq y_{is},\ x_{ijs} \leq y_{js}$, would $x_{ijs} = 1$ in an optimal sol. if either $y_{is} = 0$ or $y_{js} = 0$?

Clustering Problem: Formulation 1^{++}

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases}$$

IP Formulation:

$$\begin{array}{lrll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
 & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
 & x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N\,, s \in K & \\
 & \cancel{x_{ijs} \leq y_{is}},\ \cancel{x_{ijs} \leq y_{js}} & \cancel{\forall i < j \in N}\,, \cancel{s \in K} & \\
 & x_{ijs} \geq 0 & \forall i < j \in N\,, s \in K & \\
 & y_{is} \in \{0,1\} & \forall i \in N\,, s \in K &
\end{array}$$

Note: This formulation allows $x_{ijs} = 1$ even when items i and j are in different clusters but this would never happen in an opt. solution.

Comparing the Formulations 0, 1, 1^+ and 1^{++}

- Size of the formulations:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds

  – Form. 1^{++} enumerates more nodes but the speed up is 6x.

  – LPs are now much easier to solve (fewer constraints and non-zeroes).

Comparing the Formulations 0, 1, 1^+ and 1^{++}

- Size of the formulations:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds

  - Form. 1^{++} enumerates more nodes but the speed up is 6x.
  - LPs are now much easier to solve (fewer constraints and non-zeroes).

Clustering Problem: Formulation 2

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if objects } i \text{ and } j \text{ are put in the same cluster} \\ 0 & \text{otherwise.} \end{cases}$$

(only for $i < j$)

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} z_{ij} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters}
\end{array}$$

How to say $z_{ij} = 1$ when both $y_{is}, y_{js} = 1$ for some $s \in K$

$$\begin{array}{rl}
z_{ij} \geq 0 & \forall i < j \in N \\
y_{is} \in \{0, 1\} & \forall i \in N \,, s \in K
\end{array}$$

Clustering Problem: Formulation 2

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if objects } i \text{ and } j \text{ are put in the same cluster} \\ 0 & \text{otherwise.} \qquad \text{(only for } i < j\text{)} \end{cases}$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} z_{ij} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& z_{ij} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \,, s \in K & \\
& z_{ij} \geq 0 & \forall i < j \in N & \\
& y_{is} \in \{0, 1\} & \forall i \in N \,, s \in K &
\end{array}$$

Comparing the formulations

- Size of the formulation:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^{+}	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260
Formulation 2	900	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^{+}	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds
Formulation 2	4,392	369,175	3.98 seconds

(Formulation 2 is 35% faster than Formulation 1^{++})

Comparing the formulations

- Size of the formulation:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260
Formulation 2	900	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds
Formulation 2	4,392	369,175	3.98 seconds

(Formulation 2 is 35% faster than Formulation 1^{++})

Gurobi Logs, SVM, Production Planning

(2/7/2024)

Recap: A clustering problem

Input:

- n objects numbered $1, 2, \ldots, n$
- Desired number of clusters k, and a lower bound ℓ on the number of objects in a cluster
- A set D of pairs of dissimilar objects

Output:

- A partitioning of the objects into cluster $C_1, C_2, \ldots, C_k$

  Partitioning means:

  (i) $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, 2, \ldots, n\}$, and

  (ii) $C_s \cap C_t = \varnothing$ for all $s \neq t$.

Goal:

- Minimize the total number of pairs $\{i, j\}$ where i and j are clustered in the same cluster, but are dissimilar (meaning, $\{i, j\} \in D$)

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(only for } i < j\text{)}$$

IP Formulation:

$$\min \quad \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} \quad \longleftarrow \quad \text{dissimilar pairs in the same cluster}$$

$$\text{s.t.} \quad \sum_{s \in K} y_{is} = 1 \qquad \forall i \in N \quad \longleftarrow \quad \text{objects}$$

$$\sum_{i \in N} y_{is} \geq \ell \qquad \forall s \in K \quad \longleftarrow \quad \text{clusters}$$

How do we say: $x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$

i.e. We want $x_{ijs} = y_{is} y_{js}$

$$x_{ijs} \in \{0, 1\} \qquad \forall i < j \in N \,, \forall s \in K$$

$$y_{is} \in \{0, 1\} \qquad \forall i \in N \,, \forall s \in K$$

How to multiply binary variables

Claim

If x, y_1, y_2 satisfy the McCormick constraints

$$\begin{aligned} x &\le y_1, \\ x &\le y_2, \\ x &\ge 0, \\ x &\ge y_1 + y_2 - 1, \\ y_1, y_2 &\in \{0, 1\} \end{aligned}$$

then $x = y_1 y_2$. Note: x is not declared to be binary

Proof : Any point satisfying the above constraints has $y_1, y_2 \in \{0, 1\}$.

- If either y_1 or y_2 is 0, then the first 3 constraints imply that $x = 0$.
- The only remaining case is when both y_1 or y_2 is 1. In this case, the first constraint and the last one imply that $x = 1$.

Clustering Problem: Formulation 1^+

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad (\text{ for } i < j)$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \ , s \in K & \\
& x_{ijs} \leq y_{is}, \ x_{ijs} \leq y_{js} & \forall i < j \in N \ , s \in K & \\
& \cancel{x_{ijs} \in \{0,1\}} \ \ x_{ijs} \geq 0 & \forall i < j \in N \ , s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N \ , s \in K &
\end{array}$$

Clustering Problem: Formulation 1^{++}

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad (\text{ for } i < j)$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N, s \in K & \\
& \cancel{x_{ijs} \leq y_{is}}, \ \cancel{x_{ijs} \leq y_{js}} & \cancel{\forall i < j \in N}, \cancel{s \in K} & \\
& x_{ijs} \geq 0 & \forall i < j \in N, s \in K & \\
& y_{is} \in \{0, 1\} & \forall i \in N, s \in K &
\end{array}$$

Note: In an optimal solution, x_{ijs} is not guaranteed to be 1 when $y_{is} = 0$ or $y_{js} = 0$, but that's OK (obj function).

Clustering Problem: Formulation 2

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if objects } i \text{ and } j \text{ are put in the same cluster} \\ 0 & \text{otherwise.} \qquad \text{(for } i < j\text{)} \end{cases}$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} z_{ij} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \text{ objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \text{ clusters} \\
& z_{ij} \geq y_{is} + y_{js} - 1 & \forall i < j \in N, \forall s \in K & \\
& z_{ij} \geq 0 & \forall i < j \in N & \\
& y_{is} \in \{0, 1\} & \forall i \in N, \ \forall s \in K &
\end{array}$$

Comparing the formulations

- Size of the formulation:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260
Formulation 2	900	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds
Formulation 2	4,392	369,175	3.98 seconds

Gurobi Output for Formulation 2

```
Gurobi log file for last model:
================================

900 variables, all binary
2383 constraints, all linear; 7260 nonzeros
40 equality constraints
2343 inequality constraints
1 linear objective; 226 nonzeros.

Gurobi 9.1.1: outlev=1
threads=4
Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (linux64)
Thread count: 32 physical cores, 64 logical processors, using up
Optimize a model with 2383 rows, 900 columns and 7260 nonzeros
Model fingerprint: 0xae721739
Variable types: 0 continuous, 900 integer (900 binary)
Coefficient statistics:
Matrix range     [1e+00, 1e+00]
Objective range  [1e+00, 1e+00]
Bounds range     [1e+00, 1e+00]
RHS range        [1e+00, 1e+01]
```

(continued....)

```
Found heuristic solution: objective 81.0000000
Presolve removed 1662 rows and 554 columns
Presolve time: 0.00s
Presolved: 721 rows, 346 columns, 2274 nonzeros
Variable types: 0 continuous, 346 integer (346 binary)

Root relaxation: objective 0.000000e+00, 161 iterations, 0.00 se

* |  Nodes |   Current Node   |  Objective Bounds  |   Work
  |Exp Unex| Obj Depth IntInf|Incumbent BestBd Gap|It/Node Time

0    0       0.0   0    58     81.00  0.00  100%     -    0s
H    0       0                 34.00  0.00  100%     -    0s
0    0       0.0   0    85     34.00  0.00  100%     -    0s
0    0       1.5   0   127     34.00  1.50   95.6%   -    0s
0    0       1.5   0   124     34.00  1.50   95.6%   -    0s
0    2       1.5   0   121     34.00  1.50   95.6%   -    0s
*  271  239       17           32.00  9.04   71.7%  103   0s
H  494  297                    29.00 10.35   64.3%   93.5 0s
```

```
(continued....)

H  630  351                    28.00 11.21   59.9%   94.5 0s
*  633  335        18          27.00 11.21   58.5%   94.3 0s
H  691  316                    25.00 12.30   50.8%   95.0 0s
H  974  354                    24.00 13.91   42.0%   95.5 1s

Explored 4392 nodes (369175 simplex iterations) in 3.98 seconds

Optimal solution found (tolerance 1.00e-04)
Best objective 2.400e+01, best bound 2.400e+01, gap 0.0000%

369175 simplex iterations
4392 branch-and-cut nodes

Cutting planes:
Gomory: 3
MIR: 7
Zero half: 26
RLT: 128
BQP: 60
```

Solving IPs: computation time

- Consider the following LP formulation

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & A^1 x \geq b^1, \\ & A^2 x = b^2, \\ & x \geq 0 \end{aligned}$$

- The non-zeroes of this formulation is the number of nonzero entries in the matrices A^1 and A^2.
  - LPs are solved using either simplex or interior point algorithms,
  - In both cases one has to solve (many, many) linear equations
  - Computational burden per LP iteration typically grows with the number of non-zero entries of the constraint matrices A^1 and A^2
  - It also grows with the number of rows of A^1 and A^2.
- IP solution time depends on the number of B&B nodes and the LP solution time at each node.

Supervised binary classification

Another ML Example : Supervised binary classification

- We are given m objects and a description of their features.
- For the jth object let $a^j \in \mathbb{R}^n$ denote the associated feature vector.

  Example: $a^j \in \mathbb{R}^3$ could be: (measured in some scale)

  - a_1^j indicates the ellipticity of the object,
  - a_2^j: the length of its stem,
  - a_3^j is its color (in grayscale).

- Each objects belong to one of two classes.

  For example: It is the image of an apple or an orange

- We are interested in designing a classifier which, given a new object, will figure out the class that it belongs to.
- There are many ways of approaching this problem.

  (Ex: Decision trees, random forests, logistic regression, etc.)

Another ML Example : Supervised binary classification

- We are given m objects and a description of their features.
- For the jth object let $a^j \in \mathbb{R}^n$ denote the associated feature vector.

  Example: $a^j \in \mathbb{R}^3$ could be: (measured in some scale)

  - a_1^j indicates the ellipticity of the object,
  - a_2^j: the length of its stem,
  - a_3^j is its color (in grayscale).
- Each objects belong to one of two classes.

  For example: It is the image of an apple or an orange
- We are interested in designing a classifier which, given a new object, will figure out the class that it belongs to.
- There are many ways of approaching this problem.

  (Ex: Decision trees, random forests, logistic regression, etc.)

Linear classifiers for binary classification

- A linear classifier is defined by an n-dimensional coefficient vector $w \in \mathbb{R}^n$ and a number w_0.
- Given an object with feature vector $a \in \mathbb{R}^n$, the classifier declares it to be an apple if

$$\sum_{i=1}^{n} w_i a_i \geq w_0,$$

and an orange if

$$\sum_{i=1}^{n} w_i a_i < w_0.$$

- In words, a linear classifier makes decisions on the basis of a linear combination of the features of the object.
- Our objective is to use known objects to design a "good" linear classifier.

(Train on the known objects to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that would lead to a good linear classifier that you can use on new objects)

Linear classifiers for binary classification

- A linear classifier is defined by an n-dimensional coefficient vector $w \in \mathbb{R}^n$ and a number w_0.
- Given an object with feature vector $a \in \mathbb{R}^n$, the classifier declares it to be an apple if

$$\sum_{i=1}^{n} w_i a_i \geq w_0,$$

  and an orange if

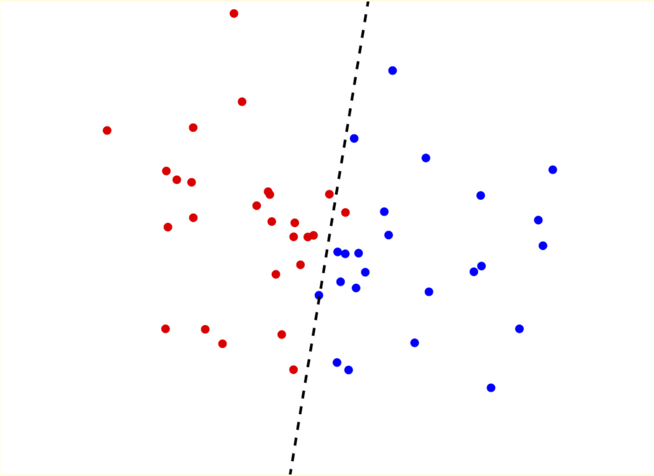
$$\sum_{i=1}^{n} w_i a_i < w_0.$$

- In words, a linear classifier makes decisions on the basis of a linear combination of the features of the object.
- Our objective is to use known objects to design a "good" linear classifier.

  (Train on the known objects to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that would lead to a good linear classifier that you can use on new objects)

Linear classifiers for binary classification

A linear classifier in $\mathbb{R}^2$:



- The coordinates of a point corresponds to its features.

Linear classifiers for binary classification

- How to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$?
- A reasonable approach would be to pick w and w_0 so that the classifier gives correct answer for all known objects (samples).
- Let $S = S^1 \cup S^2$ be the set of known samples with $S^2 = S \setminus S^1$
  - Let S^1 be the set of objects of type 1 (apples), and,
  - Let S^2 be the set of objects of type 2 (oranges)
- We are then looking for some $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that will satisfy:

$$
\begin{aligned}
w^T a^j &\geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)} \\
w^T a^j &< w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}
\end{aligned}
$$

- Note that the second set of constraints involves a strict inequality and therefore we cannot model it using linear programming.

Linear classifiers for binary classification

- How to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$?
- A reasonable approach would be to pick w and w_0 so that the classifier gives correct answer for all known objects (samples).
- Let $S = S^1 \cup S^2$ be the set of known samples with $S^2 = S \setminus S^1$
  - Let S^1 be the set of objects of type 1 (apples), and,
  - Let S^2 be the set of objects of type 2 (oranges)
- We are then looking for some $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that will satisfy:

$$
\begin{aligned}
w^T a^j &\geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)} \\
w^T a^j &< w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}
\end{aligned}
$$

- Note that the second set of constraints involves a strict inequality and therefore we cannot model it using linear programming.

How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \; = \; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0)$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\bar{w}^T a^j \geq \bar{w}_0, \qquad j \in S^1$$

$$\bar{w}^T a^j \leq \bar{w}_0 - 1, \qquad j \in S^2.$$

- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.
  (Note: w and w_0 can have negative entries)

How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \;=\; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0)$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\begin{aligned} \bar{w}^T a^j &\geq \bar{w}_0, & j \in S^1 \\ \bar{w}^T a^j &\leq \bar{w}_0 - 1, & j \in S^2. \end{aligned}$$

- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.

  (Note: w and w_0 can have negative entries)

How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \;=\; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0)$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

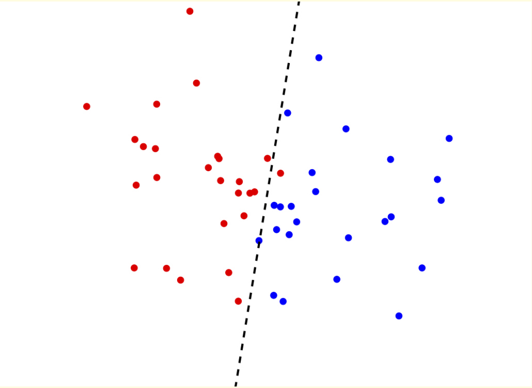
$$\begin{aligned} \bar{w}^T a^j &\geq \bar{w}_0, & j &\in S^1 \\ \bar{w}^T a^j &\leq \bar{w}_0 - 1, & j &\in S^2. \end{aligned}$$

- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.
  (Note: w and w_0 can have negative entries)

Support vector machine problem

- In practice a perfect linear classifier usually does not exist.
  i.e. the following system would be infeasible:

$$
\begin{aligned}
w^T a^j &\geq w_0, & j &\in S^1 \\
w^T a^j &\leq w_0 - 1, & j &\in S^2.
\end{aligned}
$$



Support vector machine problem

- In practice a perfect linear classifier usually does not exist.
  i.e. the following system would be infeasible:

$$
\begin{aligned}
w^T a^j &\geq w_0, && j \in S^1 \\
w^T a^j &\leq w_0 - 1, && j \in S^2.
\end{aligned}
$$

- In this case, we look for a solution that minimizes total error:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in S} \delta_j \\
\text{subject to} \quad & w^T a^j + \delta_j \geq w_0, && j \in S^1 \\
& w^T a^j - \delta_j \leq w_0 - 1, && j \in S^2 \\
& \delta_j \geq 0 && j \in S^1 \cup S^2
\end{aligned}
$$

  where the variable δ_j measures the classification error of sample j.
- If all $\delta_j = 0$ in the optimal solution, then we have a perfect classifier.

Minimizing misclassified items instead of error

- Minimizing total error:

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in S} \delta_j \\ \text{subject to} \quad & w^T a^j + \delta_j \geq w_0, && j \in S^1 \\ & w^T a^j - \delta_j \leq w_0 - 1, && j \in S^2 \\ & \delta_j \geq 0 && j \in S^1 \cup S^2 \end{aligned}$$

where the variable δ_j measures the classification error of sample j.

- Minimizing number of misclassified items:

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in S} z_j \\ \text{subject to} \quad & w^T a^j + M z_j \geq w_0, && j \in S^1 \\ & w^T a^j - M z_j \leq w_0 - 1, && j \in S^2 \\ & z_j \in \{0, 1\} && j \in S^1 \cup S^2 \end{aligned}$$

where M is a large number (max allowed error) and variable z_j indicates if sample j is classified correctly or not.

Minimizing misclassified items instead of error

- Minimizing total error:

$$\begin{array}{rll} \text{minimize} & \displaystyle\sum_{j \in S} \delta_j & \\ \text{subject to} & w^T a^j + \delta_j \geq w_0, & j \in S^1 \\ & w^T a^j - \delta_j \leq w_0 - 1, & j \in S^2 \\ & \delta_j \geq 0 & j \in S^1 \cup S^2 \end{array}$$

where the variable δ_j measures the classification error of sample j.

- Minimizing number of misclassified items:

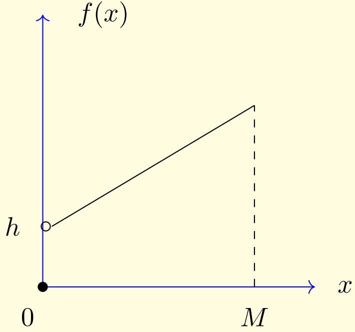
$$\begin{array}{rll} \text{minimize} & \displaystyle\sum_{j \in S} z_j & \\ \text{subject to} & w^T a^j + M z_j \geq w_0, & j \in S^1 \\ & w^T a^j - M z_j \leq w_0 - 1, & j \in S^2 \\ & z_j \in \{0, 1\} & j \in S^1 \cup S^2 \end{array}$$

where M is a large number (max allowed error) and variable z_j indicates if sample j is classified correctly or not.

Using Big Ms: Lot-sizing Problem

Fixed Charges

- Economic activities frequently involve fixed and variable costs.
- Example: A production facility.
  - Fixed cost: if anything is produced at all (e.g., cost of starting up machines).
  - Variable cost: linear in the amount produced (e.g., cost of operating machines).



- In this case, the cost is

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ h + cx & \text{if } x > 0 \end{cases}$$

(with $h, c > 0$).

- This is not a linear function.
- Not even a continuous function.

Modeling Fixed Charges

- We can hanle this using a binary variable $y \in \{0, 1\}$

$$y \stackrel{\text{interpretation}}{=} \begin{cases} 1 & \text{if } x > 0 \quad \text{(some production)} \\ 0 & \text{if } x = 0 \quad \text{(no production).} \end{cases}$$

- Then the total cost of production can now be written as

$$hy + cx.$$

- Let M be some upper bound on the value of variable x.

$$x \leq My$$
$$y \in \{0, 1\}$$
$$x \geq 0$$

Note: $y = 1$, $x = 0$ is feasible but if minimizing with $h > 0$, it is fine!

- Linear programming relaxations of "big M" formulations tend to produce bad LP relaxations.

- One should choose the smallest possible "big M".

Modeling Fixed Charges

- We can hanle this using a binary variable $y \in \{0, 1\}$

$$y \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if } x > 0 \quad \text{(some production)} \\ 0 & \text{if } x = 0 \quad \text{(no production).} \end{cases}$$

- Then the total cost of production can now be written as

$$hy + cx.$$

- Let M be some upper bound on the value of variable x.

$$\begin{aligned} &x \leq My \\ &y \in \{0, 1\} \\ &x \geq 0 \end{aligned}$$

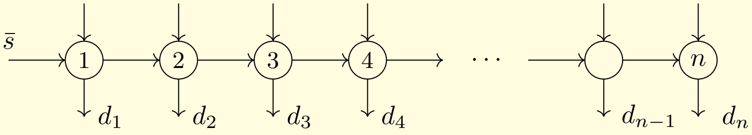
  Note: $y = 1$, $x = 0$ is feasible but if minimizing with $h > 0$, it is fine!

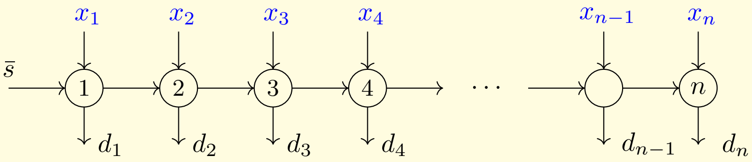
- Linear programming relaxations of "big M" formulations tend to produce bad LP relaxations.
- One should choose the smallest possible "big M".

Example: Uncapacitated lot-sizing problem

Production planning for a single item for an n-period horizon:

- Demand for the item is d_t for $t = 1, \ldots, n$.
- There is a fixed cost f_t of production in period t
- There is a production cost p_t per unit produced in period t
- There is a starting inventory of $\bar{s}$ units available at the beginning.
- There is a storage cost of h_t per unit in period t
- Find the minimum cost production plan to satisfy demand.





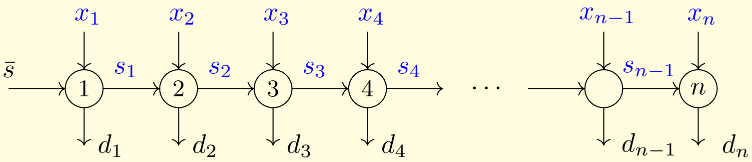
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1, \dots, n\} \\
& x_t \leq M y_t && t \in \{1, \dots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} && t \in \{1, \dots, n\}
\end{aligned}$$

Question: How big should M be?



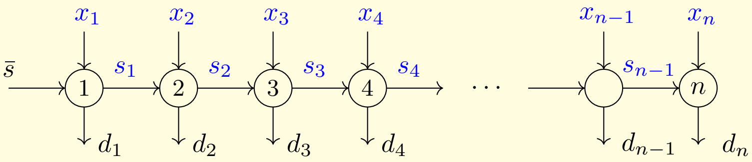
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1, \dots, n\} \\
& x_t \leq M y_t && t \in \{1, \dots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} && t \in \{1, \dots, n\}
\end{aligned}
$$

Question: How big should M be?



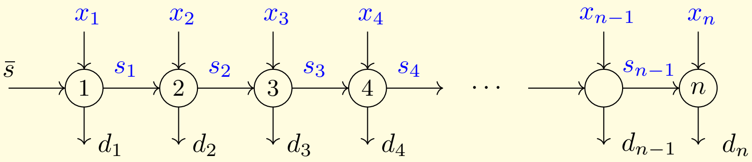
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1, \ldots, n\} \\
& x_t \leq M y_t && t \in \{1, \ldots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} && t \in \{1, \ldots, n\}
\end{aligned}
$$

Question: How big should M be?



Decision variables:

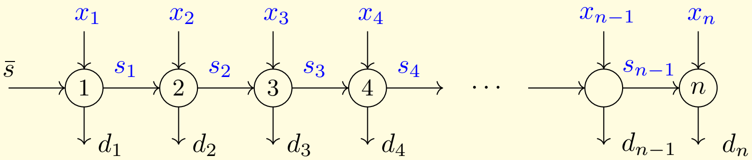
- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0,1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$\begin{array}{lll}
\min & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t & \\
\text{s.t.} & s_t = s_{t-1} + x_t - d_t & t \in \{1,\dots,n\} \\
& x_t \leq M y_t & t \in \{1,\dots,n\} \\
& s_0 = \bar{s}, & \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0,1\},\ x_t \in \mathbb{Z} & t \in \{1,\dots,n\}
\end{array}$$

Question: How big should M be?

Finding the smallest Big M



- How big should M be?

$$x_t \;\leq\; My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

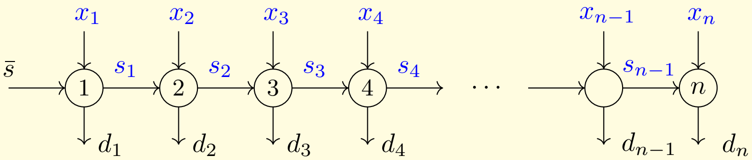
- The big Ms can be different for different t:

$$x_t \;\leq\; M^t y_t \qquad t \in T \;\longleftarrow\; \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \;\leq\; My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

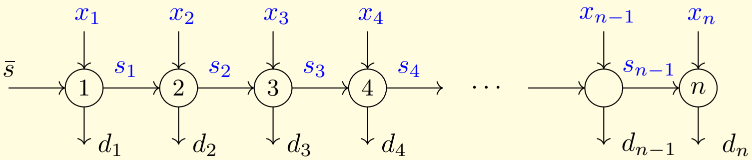
- The big Ms can be different for different t:

$$x_t \;\leq\; M^t y_t \qquad t \in T \;\longleftarrow\; \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \leq My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

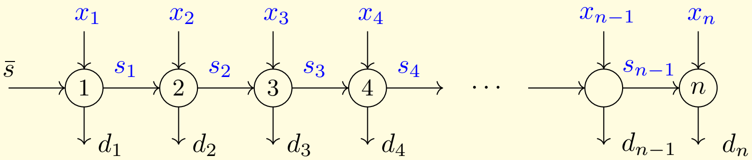
- The big Ms can be different for different t:

$$x_t \leq M^t y_t \qquad t \in T \longleftarrow \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \;\leq\; M y_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

- The big Ms can be different for different t:

$$x_t \;\leq\; M^t y_t \qquad t \in T \;\longleftarrow\; \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Big Ms

- Big M (meaning a very large number) is used in constraints of the form
$$x \leq My$$
where $y \in \{0, 1\}$
- Variable y is forced to be 1, if $x > 0$.
- This helps model fixed costs and similar relations.
- If you use big Ms, then the number M should be at least as large as the largest value x can take.
- If you use big Ms, try to use the smallest possible number to obtain a better formulation.
- If you can formulate your problem without big Ms, do not use big Ms.

2/12/2024

Recap: Linear classifiers for supervised binary classification

- A linear classifier is defined by an n-dimensional coefficient vector $w \in \mathbb{R}^n$ and a number w_0.
- Given an object with feature vector $a \in \mathbb{R}^n$, the classifier declares it to be an apple if

$$\sum_{i=1}^{n} w_i a_i \geq w_0,$$

  and an orange if

$$\sum_{i=1}^{n} w_i a_i < w_0.$$

- In words, a linear classifier makes decisions on the basis of a linear combination of the features of the object.
- Our objective is to use known objects to design a "good" linear classifier.

  (Train on the known objects to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that would lead to a good linear classifier that you can use on new objects)

Recap: How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$
$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \;=\; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0\text{)}$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\begin{aligned} \bar{w}^T a^j &\geq \bar{w}_0, & j &\in S^1 \\ \bar{w}^T a^j &\leq \bar{w}_0 - 1, & j &\in S^2. \end{aligned}$$

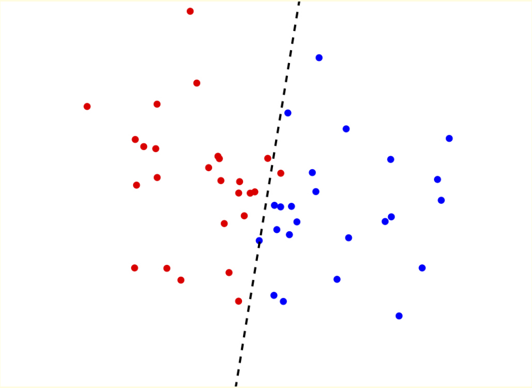
- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.
  (Note: w and w_0 can have negative entries)

Recap: Support vector machine problem

- In practice a perfect linear classifier usually does not exist.

  i.e. the following system would be infeasible for **any choice** of w and w_0:

$$
\begin{aligned}
w^T a^j &\geq w_0, && j \in S^1 \\
w^T a^j &\leq w_0 - 1, && j \in S^2.
\end{aligned}
$$



Recap: Support vector machine problem

- In practice a perfect linear classifier usually does not exist.

  i.e. the following system would be infeasible for **any choice** of w and w_0:

$$\begin{aligned} & w^T a^j \geq w_0, && j \in S^1 \\ & w^T a^j \leq w_0 - 1, && j \in S^2. \end{aligned}$$

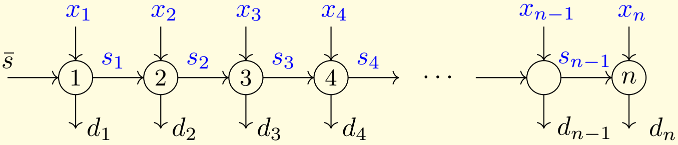
- In this case, we look for a solution that minimizes total error:

$$\begin{array}{rll} \text{minimize} & \displaystyle\sum_{j \in S} \delta_j & \\ \text{subject to} & w^T a^j + \delta_j \geq w_0, & j \in S^1 \\ & w^T a^j - \delta_j \leq w_0 - 1, & j \in S^2 \\ & \delta_j \geq 0 & j \in S^1 \cup S^2 \end{array}$$

  where the variable δ_j measures the classification error of data point j.

- If all $\delta_j = 0$ in the optimal solution, then we have a perfect classifier.

Recap: Uncapacitated lot-sizing problem



Decision variables:

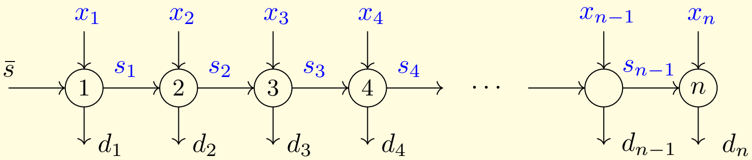
- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t & t \in \{1, \ldots, n\} \\
& x_t \leq M y_t & t \in \{1, \ldots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} & t \in \{1, \ldots, n\}
\end{aligned}
$$

Question: How big should M be?

Recap: Finding the smallest Big M



- How big should M be?

$$x_t \le My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

- The big Ms can be different for different t:

$$x_t \le M^t y_t \qquad t \in T \quad \longleftarrow \quad \{1, \ldots, n\}$$

where

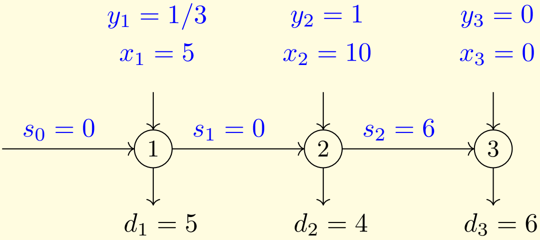
$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\le\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- Now consider a small instance with 3 time periods where
  - Fixed costs are [100, 20, 50] and all other costs are zero.
  - Demand is [5, 4, 6]
  - The optimal LP solution for this instance is



What can we do to fix this?

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\leq\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- We now disaggregate these x_t variables and use variables using
  - $q_{t,i} \geq 0$ to denote production in period t for demand in period $i \geq t$

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\le\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- We now disaggregate these x_t variables and use variables using
  - $q_{t,i} \ge 0$ to denote production in period t for demand in period $i \ge t$

New formulation:

$$\begin{array}{lll}
\min & \displaystyle\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t & \\
\text{s.t.} & s_t = s_{t-1} + x_t - d_t & t \in T \\
& x_t = \displaystyle\sum_{i=t}^{n} q_{t,i} & t \in T \\
& q_{t,i} \le d_i y_t \qquad \cancel{x_t \le M^t y_y} & t \in T,\ i \ge t \\
& \displaystyle\sum_{t=1}^{i} q_{t,i} = d_i & i \in T \\
& s_0 = \bar{s}, \quad s_t, x_t, q_{t,i} \ge 0, \quad y_t \in \{0,1\},\ x_t \in \mathbb{Z} & t \in T
\end{array}$$

Extended formulations

- The new formulation is an extended formulation

  (i.e. use additional variables to formulate the same problem)
- The first formulation had

$$x_t \;\le\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{i=t}^{n} d_i \qquad i \in T$$

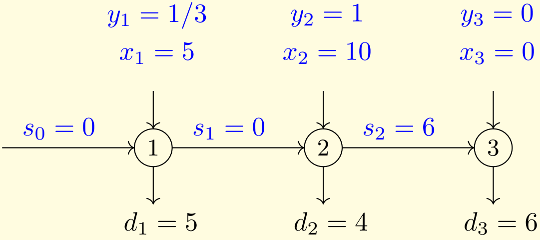
- The new formulation has additional $q_{t,i}$ variables

$$x_t = \sum_{i=t}^{n} q_{t,i}, \quad q_{t,i} \le d_i y_t, \quad \sum_{t=1}^{i} q_{t,i} = d_i \quad i \in T$$

- Notice that if a solution (x, y, s, q) is feasible for new formulation LP, (x, y, s) is feasible for original formulation LP
- But some feasible (x, y, s) for original LP are not feasible for the new LP
- Now LPs might be harder to solve but you will need fewer B&B nodes.
- Computationally, the new formulation is faster to solve te integer to optimality.

Example

- Remember the optimal LP solution to the original formulation:



- There are no possible values for $q_{t,i}$ variables that will make this solution feasible for the second formulation because

$$x_t = \sum_{i=t}^{n} q_{t,i}, \quad q_{t,i} \leq d_i y_t, \quad \sum_{i=1}^{t} q_{t,i} = d_i \quad t \in T$$

  For period $t = 1$, this means

$$q_{1,1} \leq \frac{1}{3}(5) \quad \text{and} \quad q_{1,1} = 1$$

Subset Sum Problem

Subset Sum Problem

Two brothers, Ludwig and Johann, inherit from their dear uncle a set S of antique objects.

- each worth a lot of money, c_i for all $i \in S$
- Ludwig and Johann want to share these objects in a balanced manner

$\Rightarrow$ minimize the difference between total values of objects each one gets

- Let $S_L \subset S$ be the set of objects that Ludwig will get,
- and $S_J = A \setminus S_L$ are the remaining objects (Johann gets)

$\Rightarrow$ The objective is to minimize

$$\left| \underbrace{\sum_{i \in S_L} c_i}_{\text{Ludwig's pile}} - \underbrace{\sum_{i \in S_J} c_i}_{\text{Johann's pile}} \right| \qquad \longleftarrow \quad \text{absolute value}$$

How to model this with IP?

Example: Subset Sum (cont'd)

$$x_i = \begin{cases} 1 & \text{if Ludwig gets the } i\text{-th object} \\ 0 & \text{if Johann gets the } i\text{-th object} \end{cases}$$

Then the integer model is:

$$\min \quad \left| \sum_{i \in S} c_i x_i - \sum_{i \in S} c_i (1 - x_i) \right| \qquad \longleftarrow \quad \text{absolute value}$$

$$x_i \in \{0, 1\} \qquad \forall i \in S$$

where the objective in **not** linear. But we can make it linear, by adding a new variable δ that stands for the difference between their inheritance.

$$\min \quad \delta$$

$$\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i (1 - x_i) \leq \delta$$

$$\textstyle\sum_{i \in S} c_i (1 - x_i) - \sum_{i \in S} c_i x_i \leq \delta$$

$$x_i \in \{0, 1\} \qquad \forall i \in S$$

Example: Subset Sum (cont'd)

$$x_i = \begin{cases} 1 & \text{if Ludwig gets the } i\text{-th object} \\ 0 & \text{if Johann gets the } i\text{-th object} \end{cases}$$

Then the integer model is:

$$\min \quad \left| \sum_{i \in S} c_i x_i - \sum_{i \in S} c_i (1 - x_i) \right| \qquad \longleftarrow \quad \text{absolute value}$$

$$x_i \in \{0, 1\} \qquad \forall i \in S$$

where the objective in **not** linear. But we can make it linear, by adding a new variable δ that stands for the difference between their inheritance.

$$\min \quad \delta$$

$$\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i (1 - x_i) \leq \delta$$

$$\textstyle\sum_{i \in S} c_i (1 - x_i) - \sum_{i \in S} c_i x_i \leq \delta$$

$$x_i \in \{0, 1\} \qquad \forall i \in S$$

What if there were 3 brothers?

- Now the objective is to minimize the largest difference between brothers
- What are the variables

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s.t.} \quad & \left|\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i\right| \le \delta \\ & \left|\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i\right| \le \delta \\ & \left|\textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i\right| \le \delta \\ & \qquad x_i, y_i, z_i \in \{0, 1\} \qquad \forall i \in S \end{aligned}$$

where variable δ denotes the largest difference between brothers' assets.

- Question: Is this formulation correct?

What if there were 3 brothers?

- Now the objective is to minimize the largest difference between brothers
- What are the variables

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s. t.} \quad & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i \right| \leq \delta \\ & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i \right| \leq \delta \\ & \left| \textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i \right| \leq \delta \\ & \qquad x_i, y_i, z_i \in \{0, 1\} \qquad \forall i \in S \end{aligned}$$

where variable δ denotes the largest difference between brothers' assets.

- Question: Is this formulation correct?

What if there were 3 brothers?

- Now the objective is to minimize the largest difference between brothers
- What are the variables

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s. t.} \quad & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i \right| \le \delta \\ & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i \right| \le \delta \\ & \left| \textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i \right| \le \delta \\ & \quad x_i, y_i, z_i \in \{0, 1\} \qquad \forall i \in S \end{aligned}$$

where variable δ denotes the largest difference between brothers' assets.

- Question: Is this formulation correct?

What if there were 3 brothers?

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the correct integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s. t.} \quad & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i \right| \le \delta \\ & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i \right| \le \delta \\ & \left| \textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i \right| \le \delta \\ & \qquad x_i + y_i + z_i = 1 \qquad \forall i \in S \\ & \qquad x_i, y_i, z_i \in \{0, 1\} \qquad \forall i \in S \end{aligned}$$

where variable δ denotes the largest difference between brothers' assets.

What if there were 3 brothers?

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the correct integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s. t.} \quad & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i \right| \leq \delta \\ & \left| \textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i \right| \leq \delta \\ & \left| \textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i \right| \leq \delta \\ & \qquad x_i + y_i + z_i = 1 \qquad \forall i \in S \\ & \qquad x_i, y_i, z_i \in \{0, 1\} \qquad \forall i \in S \end{aligned}$$

  where variable δ denotes the largest difference between brothers' assets.

- We can again make this model linear by replacing each absolute value with 2 linear inequalities. (i.e. $|\alpha| \leq \delta \iff \alpha \leq \delta,\ -\alpha \leq \delta$)

What if there were 3 brothers?

$$x_i = \begin{cases} 1 & \text{if L gets } i \\ 0 & \text{if not} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if J gets } i \\ 0 & \text{if not} \end{cases} \qquad z_i = \begin{cases} 1 & \text{if Max gets } i \\ 0 & \text{if not} \end{cases}$$

- Then the correct integer model is:

$$\begin{aligned} \min \quad & \delta \\ \text{s. t.} \quad & \left|\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i y_i\right| \le \delta \\ & \left|\textstyle\sum_{i \in S} c_i x_i - \sum_{i \in S} c_i z_i\right| \le \delta \\ & \left|\textstyle\sum_{i \in S} c_i y_i - \sum_{i \in S} c_i z_i\right| \le \delta \\ & \qquad x_i + y_i + z_i = 1 \qquad \forall i \in S \\ & \qquad x_i, y_i, z_i \in \{0,1\} \qquad \forall i \in S \end{aligned}$$

Notice that the number of fairness constraints increase quadratically with the number of brothers: one for each pair: $\binom{n}{2}$

where variable δ denotes the largest difference between brothers' assets.

- We can again make this model linear by replacing each absolute value with 2 linear inequalities. (i.e. $|\alpha| \le \delta \iff \alpha \le \delta,\ -\alpha \le \delta$)

The Quadratic Assignment Problem (QAP)

(we will do this later)

The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent from facility k to facility l, for $k, l \in \{1, \ldots, n\}$
- c_{ij} denotes the **unit** transportation cost between location i and j, for $i, j \in \{1, \ldots, n\}$
- Place the facilities to minimize the total transportation cost.

Let variable $x_{k,i}$ to indicate that facility k is assigned to location i.

Constraints are easy:

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{n} x_{k,i} = 1 && k \in \underbrace{\{1, \ldots, n\}}_{\text{Facilities}} \\ & \sum_{k=1}^{n} x_{k,i} = 1 && i \in \underbrace{\{1, \ldots, n\}}_{\text{Locations}} \\ & x_{k,i} \in \{0, 1\} && i, k \in \{1, \ldots, n\} \end{aligned}$$

The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent from facility k to facility l, for $k, l \in \{1, \ldots, n\}$
- c_{ij} denotes the **unit** transportation cost between location i and j, for $i, j \in \{1, \ldots, n\}$
- Place the facilities to minimize the total transportation cost.

Let variable $x_{k,i}$ to indicate that facility k is assigned to location i.

Constraints are easy:

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{i=1}^{n} x_{k,i} = 1 && k \in \underbrace{\{1, \ldots, n\}}_{\text{Facilities}} \\
& \sum_{k=1}^{n} x_{k,i} = 1 && i \in \underbrace{\{1, \ldots, n\}}_{\text{Locations}} \\
& x_{k,i} \in \{0, 1\} && i, k \in \{1, \ldots, n\}
\end{aligned}
$$

The Quadratic Assignment Problem

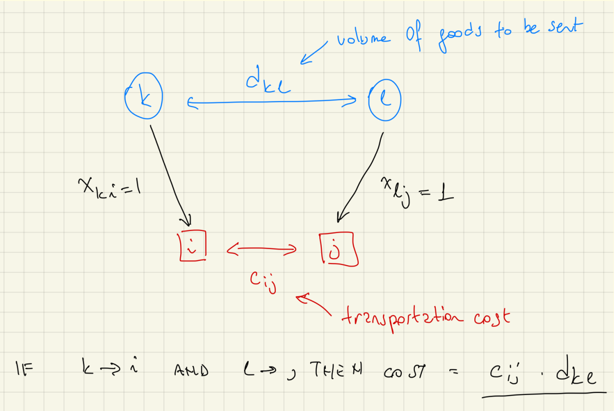
- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent between facilities k and l
- c_{ij} denotes unit transportation cost between locations i and j
- Place the facilities to minimize the total transportation cost.
- Variable $x_{k,i} \in \{0, 1\}$ to indicate that facility k is assigned to location i.

  What is the transportation cost for two facilities k and l?

The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent between facilities k and l
- c_{ij} denotes unit transportation cost between locations i and j
- Place the facilities to minimize the total transportation cost.
- Variable $x_{k,i} \in \{0,1\}$ to indicate that facility k is assigned to location i.

What is the transportation cost for two facilities k and l?



The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent between facilities k and l
- c_{ij} denotes unit transportation cost between locations i and j
- Place the facilities to minimize the total transportation cost.
- Variable $x_{k,i} \in \{0,1\}$ to indicate that facility k is assigned to location i.

  What is the transportation cost for two facilities k and l?

  - If facility k is assigned to i and facility l is assigned to j, you pay $d_{kl}\ c_{ij}$ for that pair:

$$d_{kl} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \quad \underbrace{c_{ij}\ x_{ki}\, x_{l,j}}_{=0 \text{ unless } x_{ki}=x_{l,j}=1}$$

The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent between facilities k and l
- c_{ij} denotes unit transportation cost between locations i and j
- Place the facilities to minimize the total transportation cost.
- Variable $x_{k,i} \in \{0,1\}$ to indicate that facility k is assigned to location i.

What is the transportation cost for two facilities k and l?

- If facility k is assigned to i and facility l is assigned to j, you pay $d_{kl}\ c_{ij}$ for that pair:

$$d_{kl} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \underbrace{c_{ij}\ x_{ki}\, x_{l,j}}_{=0 \text{ unless } x_{ki}=x_{l,j}=1}$$

The objective minimizes total transportation cost:

$$\min \quad \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{kl}\ c_{ij}\ x_{ki}\, x_{l,j}$$

The Quadratic Assignment Problem

- Place n facilities in n locations (one in each location).
- d_{kl} denotes volume of goods to be sent between facilities k and l
- c_{ij} denotes unit transportation cost between locations i and j
- Place the facilities to minimize the total transportation cost.
- Variable $x_{k,i}$ to indicate that facility k is assigned to location i.

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, d_{kl} \underbrace{x_{ki}\, x_{l,j}}_{\text{quadratic}} \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_{k,i} = 1 && k \in \{1, \ldots, n\} \\
& \sum_{k=1}^{n} x_{k,i} = 1 && i \in \{1, \ldots, n\} \\
& x_{k,i} \in \{0, 1\} && i, k \in \{1, \ldots, n\}
\end{aligned}
$$

Remember : Multiplying binary variables

- Let $x \in \{0, 1\}$ and $y \in \{0, 1\}$ be two binary variables.
- Assume we are interested in their product $x \cdot y$.
- How can we express their product $w = xy$ using linear inequalities?

$$w = \begin{cases} 1 & \text{if } x = 1 \text{ and } y = 1 \\ 0 & \text{otherwise.} \end{cases}$$

- Consider the following McCormick inequalities:

$$\begin{aligned} w &\leq x \\ w &\leq y \\ w &\geq 0 \\ w &\geq x + y - 1 \end{aligned}$$

Claim
If $x, y \in \{0, 1\}$ and w satisfies the constraints above, then $w = xy$.

The Quadratic Assignment Problem: IP formulation

- We have to place n facilities in n locations.

$$
\begin{array}{lll}
\min & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} & c_{ij}\, d_{kl}\, \underbrace{\cancel{x_{ki}\, x_{lj}}}_{y_{kilj}} \\
\text{s.t.} & \displaystyle\sum_{i=1}^{n} x_{k,i} = 1 & k \in \{1, \ldots, n\} \\
& \displaystyle\sum_{k=1}^{n} x_{k,i} = 1 & i \in \{1, \ldots, n\} \\
& x_{k,i} \in \{0, 1\} & i, k \in \{1, \ldots, n\} \\
& x_{ki} \geq y_{kilj} & \forall\, i, j, k, l \in \{1, \ldots, n\} \\
& x_{lj} \geq y_{kilj} & \forall\, i, j, k, l \in \{1, \ldots, n\} \\
& y_{kilj} \geq x_{ki} + x_{lj} - 1 & \forall\, i, j, k, l \in \{1, \ldots, n\} \\
& y_{kilj} \geq 0 & \forall\, i, j, k, l \in \{1, \ldots, n\}
\end{array}
$$