Location 8:
Discrete Location and MILP

- What computer technology has had the biggest impact on industry since 1990?
   1. Data science/analytics
   2. AI (visual/voice recognition, ChatGPT, etc)
   3. MILP solver improvements
- MILP biggest impact has been on tactical decisions
   – Production planning
   – Scheduling

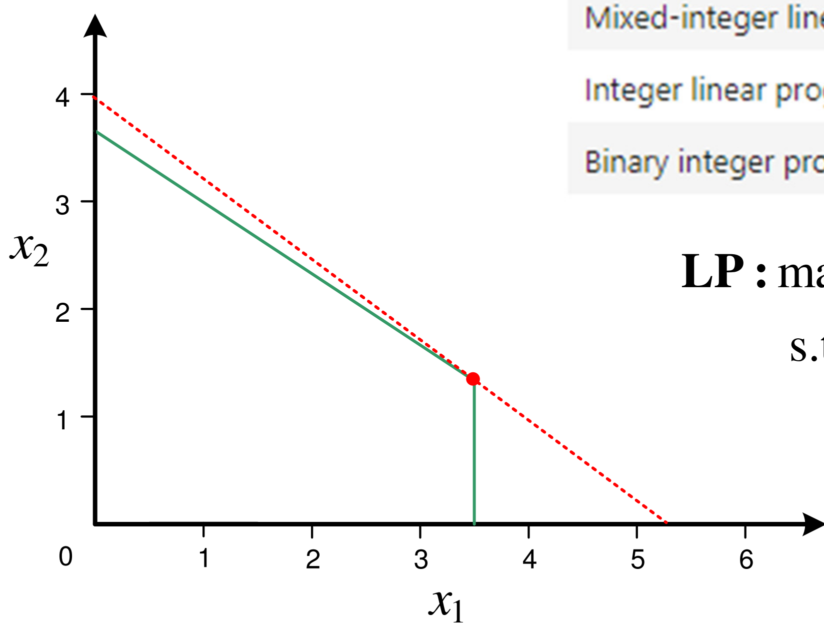
Speedup from 1990-2014:
- 320,000 $\times$ computer speed
- 580,000 $\times$ algorithm improvements

$\Rightarrow$ 10 days of 24/7 processing $\rightarrow$ 1 sec

Mixed-Integer Programming

- Mixed-integer programming used when
  - decision variables need to be integer-valued
  - decisions need to be made as part of the solution procedure, and can only be implemented using discrete (typically binary) decision variables

Mixed-integer program	MIP	NLP + some integer variables
Mixed-integer linear program	MILP	LP + some integer variables
Integer linear program	ILP	LP + all integer variables
Binary integer program	BIP	LP + all binary variables



$$
\begin{aligned}
\textbf{LP}: \max \quad & 6x_1 + 8x_2 \\
\text{s.t.} \quad & 2x_1 + 3x_2 \le 11 \\
& 2x_1 \le 7 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

$$
\begin{aligned}
\text{LP:} \quad & \max \ \mathbf{c}'\mathbf{x} \\
& \text{s.t.} \quad \mathbf{A}\mathbf{x} \le \mathbf{b} \\
& \qquad \mathbf{x} \ge 0 \\
\text{MILP:} \quad & \text{some } x_i \text{ integer} \\
\text{ILP:} \quad & \mathbf{x} \text{ integer} \\
\text{BLP:} \quad & \mathbf{x} \in \{0,1\}
\end{aligned}
$$

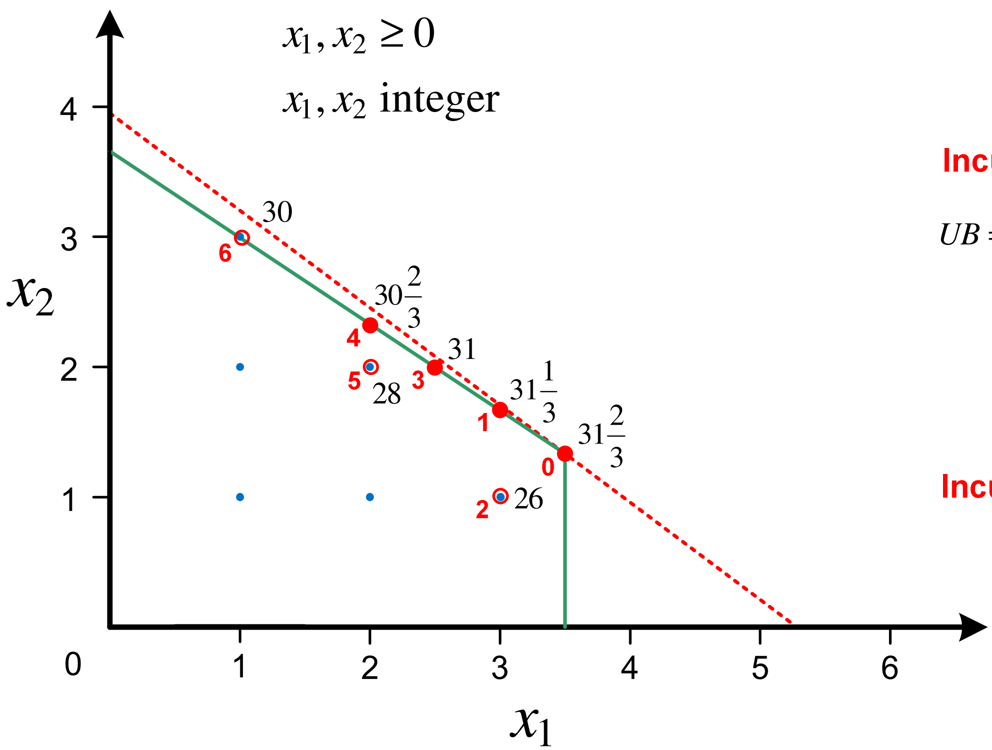
Solving a ILP: Branch and Bound

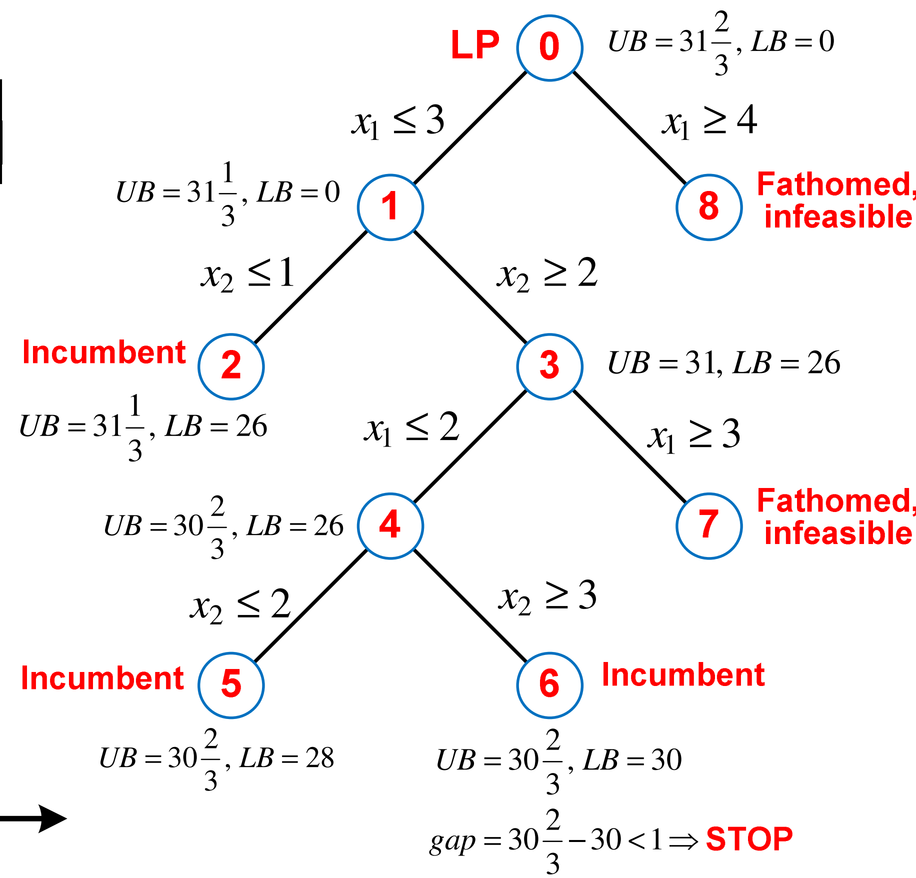
- For maximization ILP problems:
  - LP solutions provide *UB*s
  - Feasible ILP (incumbent) solutions provide *LB*s
- Stop when *gap* = *UB* – *LB* < some threshold

$$\begin{aligned} \max \quad & 6x_1 + 8x_2 \\ \text{s.t.} \quad & 2x_1 + 3x_2 \le 11 \\ & 2x_1 \qquad\;\; \le 7 \\ & x_1, x_2 \ge 0 \\ & x_1, x_2 \text{ integer} \end{aligned}$$

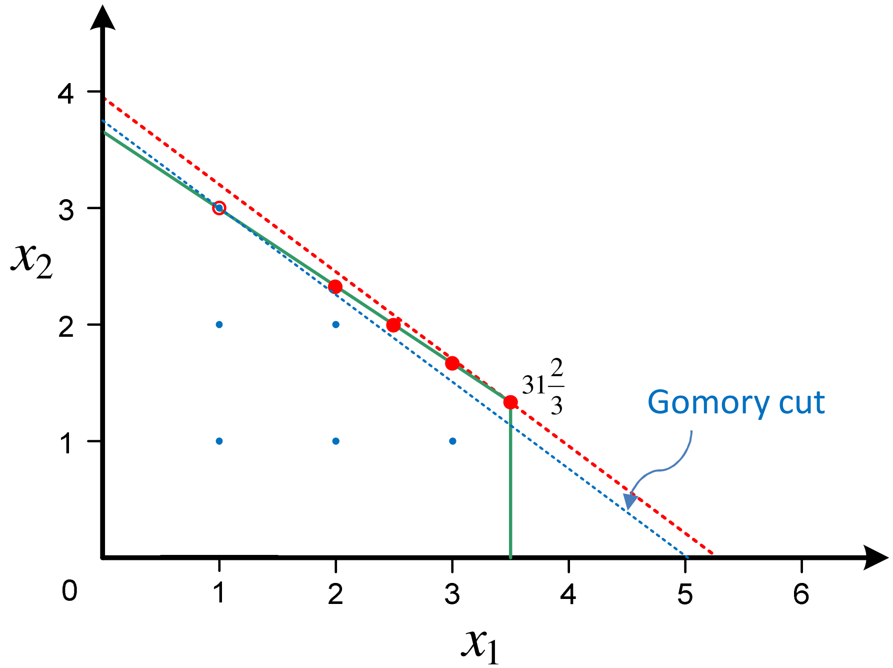
$$\mathbf{c} = \begin{bmatrix} 6 & 8 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 2 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 11 \\ 7 \end{bmatrix}$$





MILP Solvers



Cplex	IBM, first commercial solver
Gurobi	Developed by Robert Bixby
FICO Xpress	Used by LLamasoft
SAS/OR	Part of SAS system (not supported in `JuMP`)
Cbc	COIN-OR open-source solver
GLPK	Free Software Foundation GNU open-source solver

- **Pre-solve**: eliminate variables
  $$2x_1 + 2x_2 \le 1,\ x_1, x_2 \ge 0 \text{ and integer}$$
  $$\Rightarrow x_1 = x_2 = 0$$
- **Cutting planes**: cuts off LP solutions (Gomory cut)
- **Heuristics**: find good initial incumbent solution (Hybrid UFL)
- **Parallel**: use separate cores to solve nodes in B&B tree
- **Speedup** from 1990-2014:
  - 320,000 $\times$ computer speed
  - 580,000 $\times$ algorithm improvements

  $\Rightarrow$ 10 days of 24/7 processing $\rightarrow$ 1 sec
- **Speedup** from 2001-2020:
  - 20 $\times$ computer speed
  - 50 $\times$ algorithm improvements

  $\Rightarrow$ 1000 $\times$ speedup

Total Logistics Costs

- All costs impacting location decision termed *total logistics costs*
- When production costs are represented by a line with intercept (k) and slope (c_p) terms
  - only cost associated with adding another NF is k
  - c_p the same for any number of NFs

Total production cost: $TPC = k + c_p f$

Total transport cost: $TC = f\,rd$

Total logistics cost: $TLC = k + f\,rd$

where

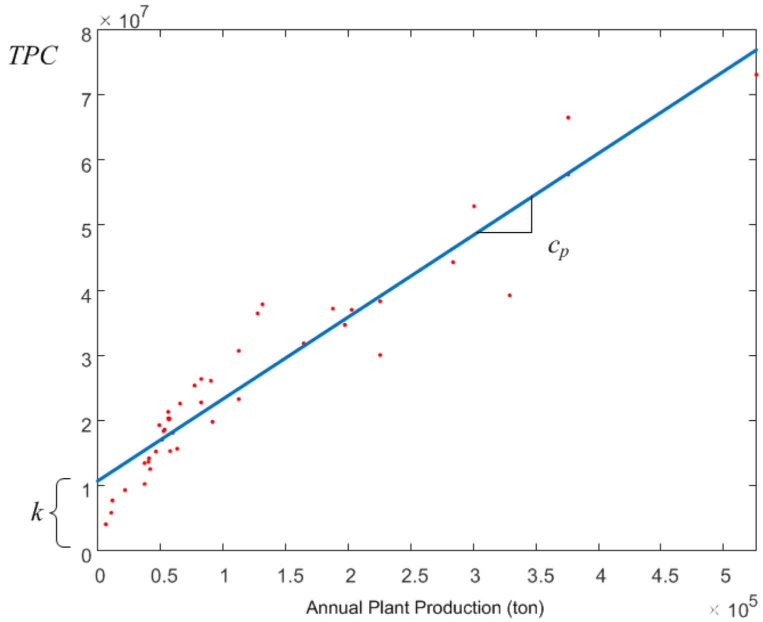
k = fixed production cost (\$/yr)

c_p = unit production cost (\$/ton)

f = production rate (ton/yr)

r = transport rate (\$/ton-mi)

d = transport distance (mi)



Uncapacitated Facility Location (UFL)

- NFs can only be located at discrete set of sites
  - Allows inclusion of fixed cost of locating NF at site $\Rightarrow$ opt number NFs
  - Variable costs are usually transport cost from NF to/from EF
  - Total of $2^n - 1$ potential solutions (all nonempty subsets of sites)

$$M = \{1, \ldots, m\}, \quad \text{existing facilites (EFs)}$$

$$N = \{1, \ldots, n\}, \quad \text{sites available to locate NFs}$$

$$M_i \subseteq M, \quad \text{set of EFs served by NF at site } i$$

$$c_{ij} = \text{variable cost to serve EF } j \text{ from NF at site } i$$

$$k_i = \text{fixed cost of locating NF at site } i$$

$$Y \subseteq N, \quad \text{sites at which NFs are located}$$

$$\begin{aligned} Y^* &= \arg\min_Y \left\{ \sum_{i \in Y} k_i + \sum_{i \in Y} \sum_{j \in M_i} c_{ij} : \bigcup_{i \in Y} M_i = M \right\} \\ &= \text{min cost set of sites where NFs located} \end{aligned}$$

$$\left| Y^* \right| = \text{number of NFs located}$$

UFL Solution Techniques

- Being uncapacitated allows simple heuristics to be used to solve
  - ADD construction: add one NF at a time
  - DROP construction: drop one NF at a time
  - XCHG improvement: move one NF at a time to unoccupied sites
  - HYBRID algorithm combination of ADD and DROP construction with XCHG improvement, repeating until no change in Y
    - Use as default heuristic for UFL
    - See Daskin [2013] for more details
- UFL can be solved as a MILP
  - Easy MILP, LP relaxation usually optimal (for strong formulation)
  - MILP formulation allows constraints to easily be added
    - e.g., capacitated facility location, fixed number of NFs, some NF at fixed location

MILP Formulation of UFL

$$\min \quad \sum_{i\in N} k_i y_i + \sum_{i\in N}\sum_{j\in M} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i\in N} x_{ij} = 1, \qquad j\in M$$

$$\cancel{m y_i \geq \sum_{j\in M} x_{ij}, \qquad i\in N}$$

$$0 \leq x_{ij} \leq 1, \qquad i\in N, j\in M$$

$$y_i \in \{0,1\}, \qquad i\in N$$

- Every EF needs to be assigned to a NF
- NF needs to be established at every site that has EFs assigned to it
  - Max m EFs can be assigned to each NF
- x_{ij}'s always 0 or 1 in UFL
  - But easier to solve if not binary

$$\boxed{y_i \geq x_{ij}, \qquad i\in N, j\in M}$$

Strong Formulation

where

$$k_i = \text{fixed cost of NF at site } i\in N = \{1,\ldots,n\}$$

$$c_{ij} = \text{variable cost from } i \text{ to serve EF } j\in M = \{1,\ldots,m\}$$

$$y_i = \begin{cases} 1, & \text{if NF established at site } i \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} = \text{fraction of EF } j \text{ demand served from NF at site } i.$$

UFL Costs in MILP

$$TLC = \sum_{i \in N} k_i y_i + \sum_{i \in N} \sum_{j \in M} c_{ij} x_{ij} = \sum_{i \in N} k_i y_i + \sum_{i \in N} \sum_{j \in M} \underbrace{f_i \, r \, d_{ij}}_{c_{ij}} x_{ij}$$

where TLC = total fixed and variable cost (\$/yr)

k_i = fixed production cost (\$/yr)

c_{ij} = variable transport cost (\$/yr)

f_i = demand rate (ton/yr)

r = transport rate (\$/ton-mi)

d_{ij} = distance between NF at site i and EF at site j (mi)

Capacitated Facility Location (CFL)

$$\min \quad \sum_{i\in N} k_i y_i + \sum_{i\in N} \sum_{j\in M} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i\in N} x_{ij} = 1, \qquad j \in M$$

$$K_i y_i \geq \sum_{j\in M} f_j x_{ij}, \; i \in N$$

$$0 \leq x_{ij} \leq 1, \qquad i \in N, j \in M$$

$$y_i \in \{0,1\}, \qquad i \in N$$

where

$$k_i = \text{fixed cost of NF at site } i \in N = \{1,\ldots,n\}$$

$$c_{ij} = \text{variable cost from } i \text{ to serve EF } j \in M = \{1,\ldots,m\}$$

$$K_i = \text{capacity of NF at site } i \in N = \{1,\ldots,n\}$$

$$f_j = \text{demand EF } j \in M = \{1,\ldots,m\}$$

$$y_i = \begin{cases} 1, & \text{if NF established at site } i \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} = \text{fraction of EF } j \text{ demand served from NF at site } i.$$

- CFL does not have simple and effective heuristics, unlike UFL
- Other types of constraints:
  - Fix NF *i* at site *j*: set *LB* and *UB* of x_{ij} to 1
  - Convert UFL to p-Median: set all *k* to 0 and add constraint sum{y_i} = *p*

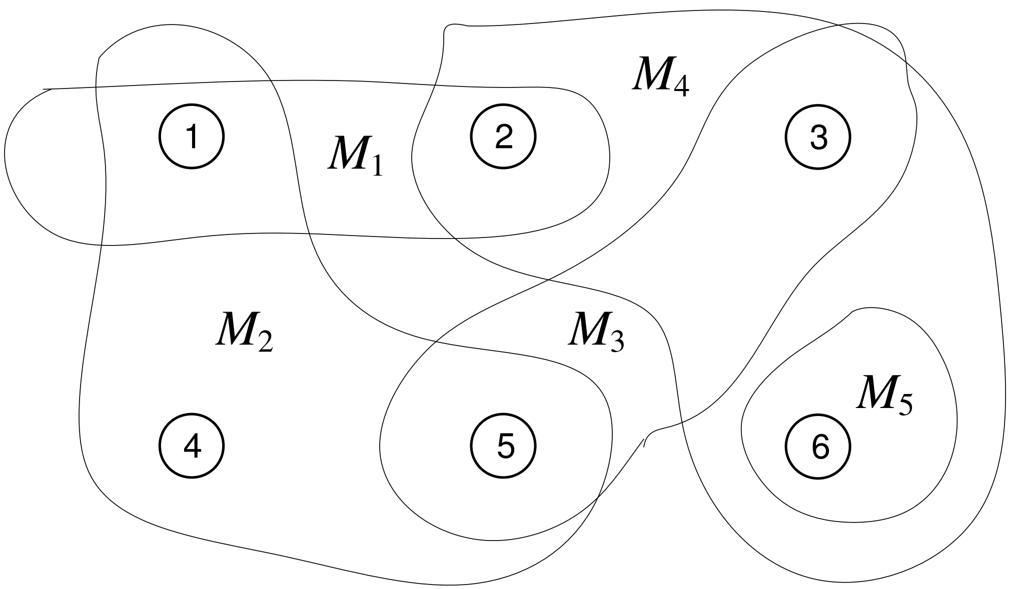
(Weighted) Set Covering

$M = \{1, \ldots, m\}$, objects to be covered

$M_i \subseteq M,\ i \in N = \{1, \ldots, n\}$, subsets of M

$c_i =$ cost of using M_i in cover

$$I^* = \arg\min_I \left\{ \sum_{i \in I} c_i : \bigcup_{i \in I} M_i = M \right\}, \quad \text{min cost covering of } M$$



$M = \{1, \ldots, 6\}$

$i \in N = \{1, \ldots, 5\}$

$M_1 = \{1,2\}, M_2 = \{1,4,5\}, M_3 = \{3,5\}$

$M_4 = \{2,3,6\}, M_5 = \{6\}$

$c_i = 1$, for all $i \in N$

$$I^* = \arg\min_I \left\{ \sum_{i \in I} c_i : \bigcup_{i \in I} M_i = M \right\}$$

$$= \{2,4\}$$

$$\sum_{i \in I^*} c_i = 2$$

(Weighted) Set Covering

$$M = \{1,\ldots,m\}, \quad \text{objects to be covered}$$

$$M_i \subseteq M,\, i \in N = \{1,\ldots,n\}, \quad \text{subsets of } M$$

$$c_i = \text{cost of using } M_i \text{ in cover}$$

$$I^* = \arg\min_I \left\{ \sum_{i\in I} c_i : \bigcup_{i \in I} M_i = M \right\}, \quad \text{min cost covering of } M$$

$$\begin{aligned} \min \quad & \sum_{i\in N} c_i x_i \\ \text{s.t.} \quad & \sum_{i\in N} a_{ji} x_i \geq 1, \qquad j \in M \\ & x_i \in \{0,1\}, \qquad i \in N \end{aligned}$$

```
model = Model(Cbc.Optimizer)   # Unweighted set cover
@variable(model, x[N], Bin )
@objective(model, Min, sum(x[i] for i ∈ N) )
@constraint(model, [j ∈ M], sum(A[j,i]*x[i] for i ∈ N) >= 1 )
```

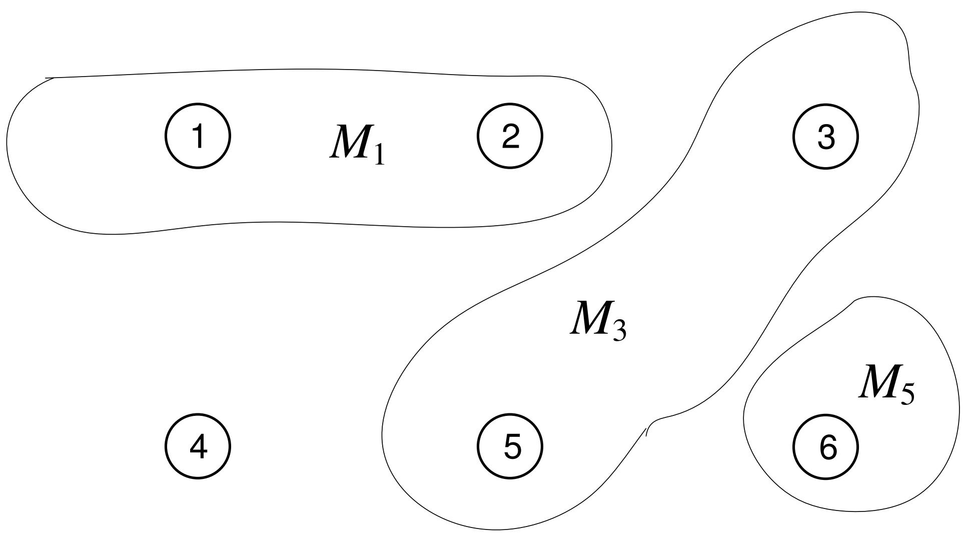
where

$$x_i = \begin{cases} 1, & \text{if } M_i \text{ is in cover} \\ 0, & \text{otherwise} \end{cases}$$

$$a_{ji} = \begin{cases} 1, & \text{if } j \in M_i \\ 0, & \text{otherwise.} \end{cases}$$

Set Packing

- Maximize the number of mutually disjoint sets
  - Dual of Set Covering problem
  - Not all objects required in a packing
  - Limited logistics engineering application (c.f. bin packing)



$$
\begin{aligned}
\max \quad & \sum_{i \in N} x_i \\
\text{s.t.} \quad & \sum_{i \in N} a_{ji} x_i \le 1, \qquad j \in M \\
& x_i \in \{0,1\}, \qquad i \in N
\end{aligned}
$$