the MILP way

a practical view

Sophie Demassey

CMA Centre de Mathématiques Appliquées


scheduling
planning
packing
allocation
practical decision is
combinatorial optimization
design
sizing
assignment
routing
cover

MILP perks

fast
based on LP + enumeration
+ advanced features

declarative
create the model,
apply a solver

generic & specific
algorithms

optimality
primal-dual bounds

expressive
logic, nonlinear, discrete
many decision problems

flexible
change the model,
not the solver

1 how to model ?

techniques & applications

2 how difficult ?

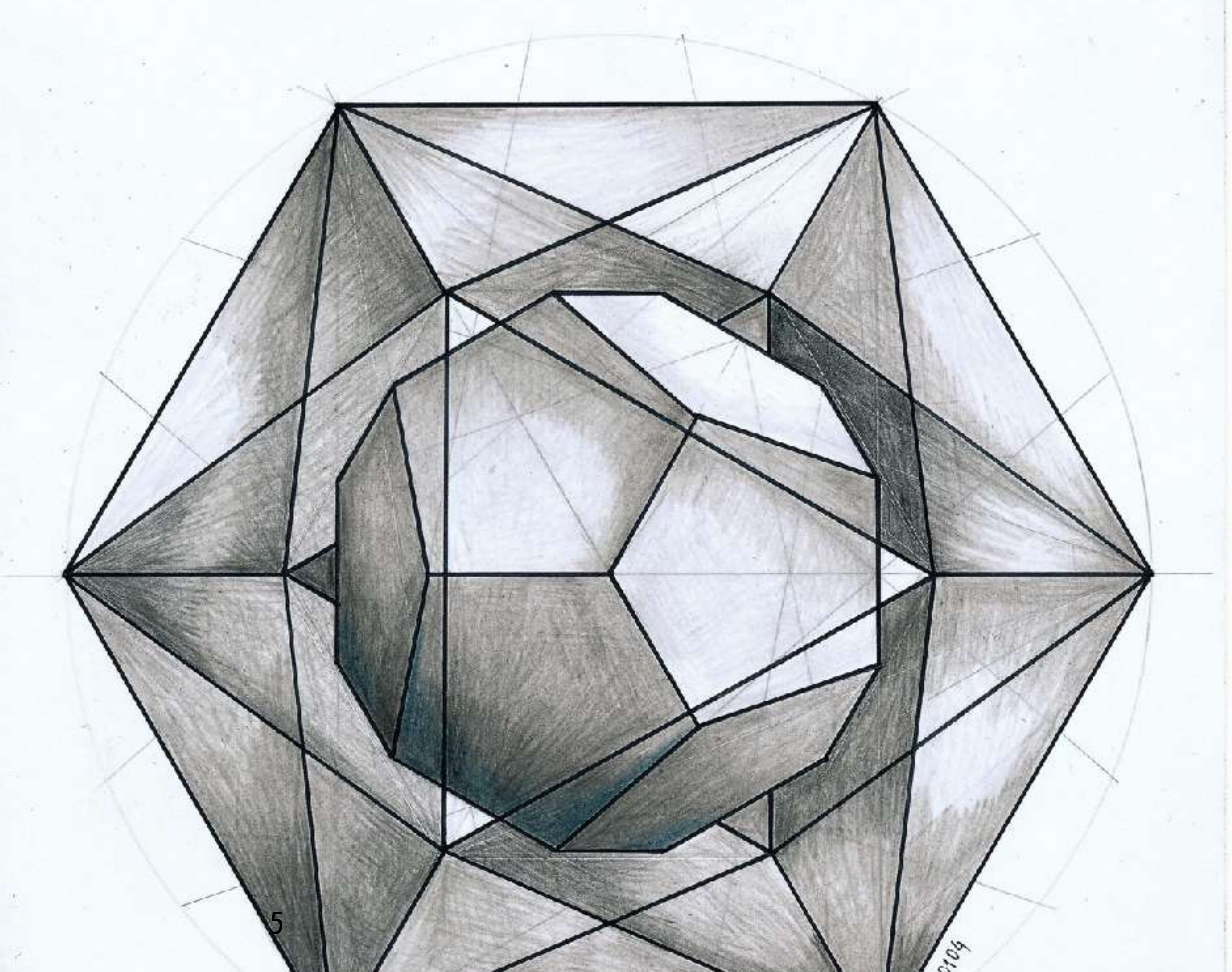
complexity & distance to LP

3 how to solve ?

main techniques & modern solvers

decomposition methods

1 how to model ?



Mixed Integer Linear Program

$$\min cx$$
$$Ax \geq b$$
$$x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}$$

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

objective	cx
linear constraints	$Ax \geq b$
integrity constraints	$x_j \in \mathbb{Z}$
right hand side	b
cost vector	c
solution space	$\mathbb{R}^n$
feasible set	$\{x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \mid Ax \geq b\}$

modeling with $\mathbb{B}$

~~true~~ 1 or ~~false~~ 0

- select item `j` $\qquad x_j = 1, \quad x_j \in \{0,1\}$
- associate item `j` to resource `i` $\qquad x_{ij} = 1, \quad x_{ij} \in \{0,1\}$
- variable $y \geq 0$ greater than constant a ? $\qquad y \geq ax, x \in \{0,1\}$
- select at most n items $\qquad x_1, \ldots, x_n \in \{0,1\}$

Integer Knapsack Problem

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_j \le K$$

$$x_j \in \{0,1\} \qquad\qquad j = 1..n$$

Input n items, value c_j and weight w_j for each item j, capacity K.

Output a maximum value subset of items whose total weight does not exceed K.

x_j is item j packed ?

logic with binaries

x,y binary variables; f continuous variable; a, k, n constants

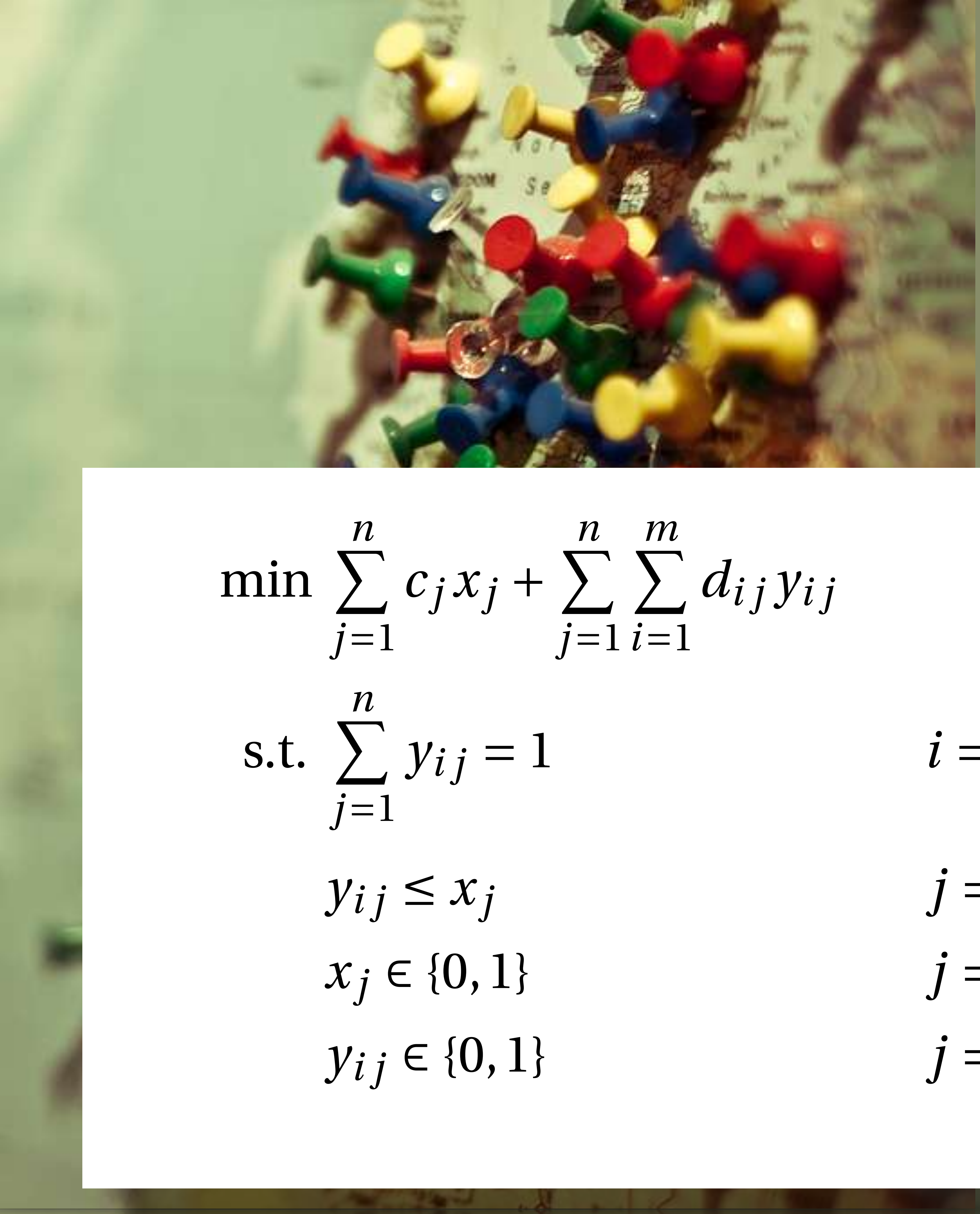
- either x or y $$x + y = 1$$
- if x then y $$y \geq x$$
- if x then f ≤ a $$f \leq ax + M(1 - x)$$
- at most 1 out of n
- at least k out of n

"big M"
big enough but
keep it tight !

logic with binaries

x,y binary variables; f continuous variable; a, k, n constants

- either x or y $$x + y = 1$$
- if x then y $$y \geq x$$
- if x then f ≤ a $$f \leq ax + M(1 - x)$$
- at most 1 out of n $$x_1 + \cdots + x_n \leq 1$$
- at least k out of n $$x_1 + \cdots + x_n \geq k$$



Uncapacitated Facility Location Problem

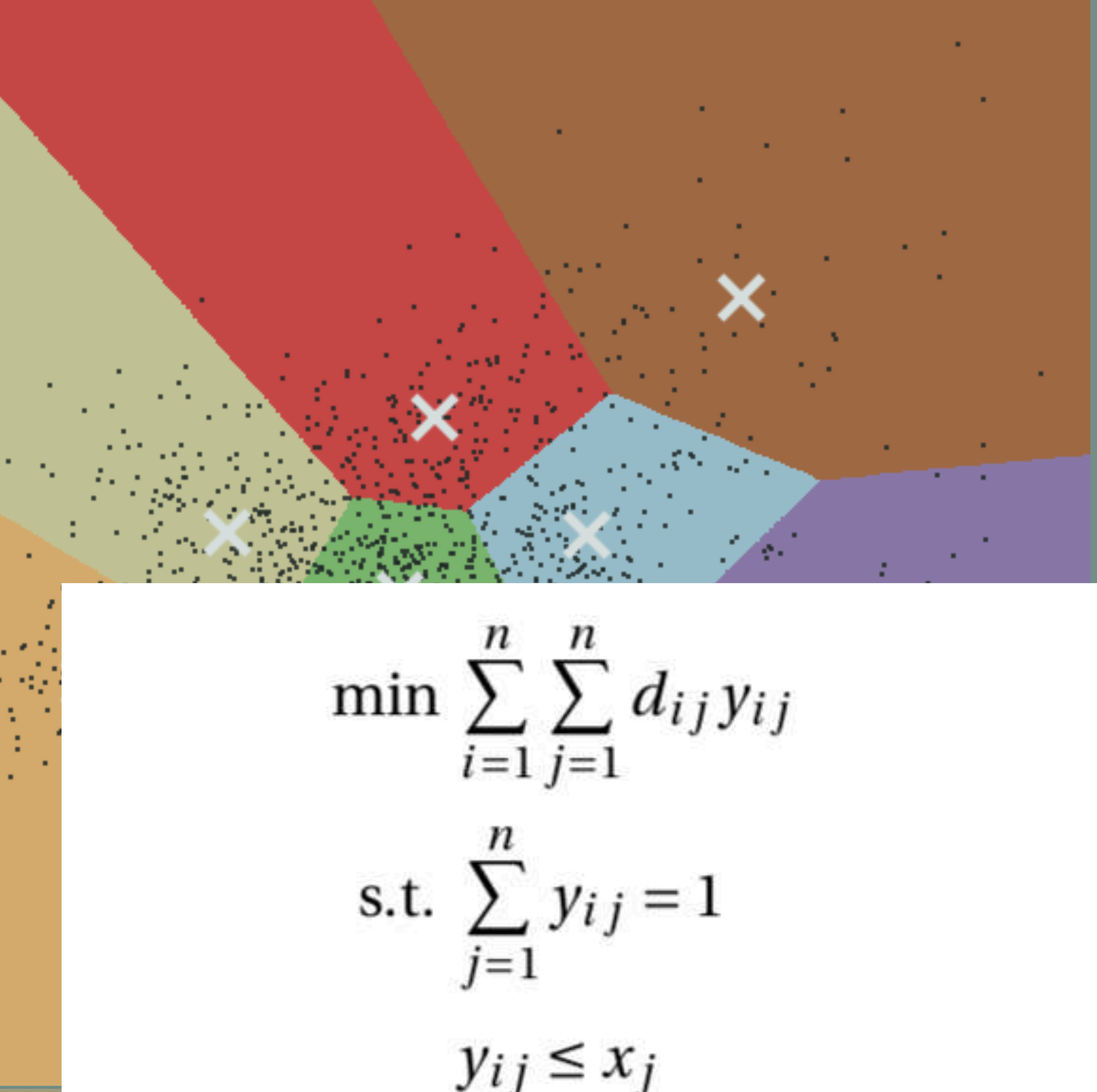
$$\min \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} \sum_{i=1}^{m} d_{ij} y_{ij}$$

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

Input n facility locations, m customers, cost c_j to open facility j, cost d_{ij} to serve customer i from facility j

Output a mimimum (opening and service) cost assignment of customers to facilities.

x_j is location j open ? y_{ij} is customer i served from j ?

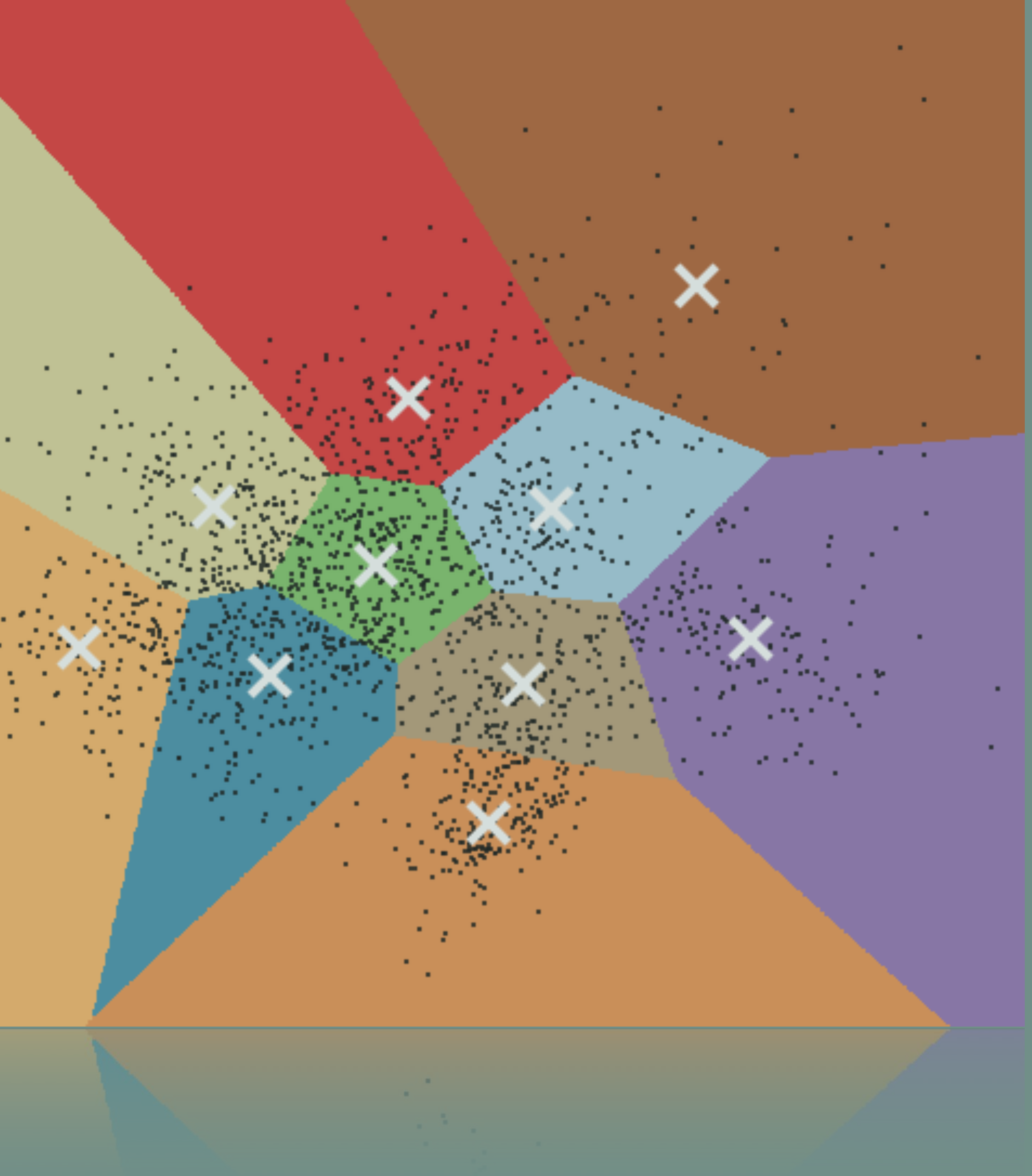


K-median clustering

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

Input n data points, distance d_{ij} between each two points i,j, number k of clusters. **Output** k centers minimizing the sum of distances between each point and its nearest center.

x_j is j a center ? y_{ij} is j the nearest center of i ?



Input n data points $m_j \in \mathbb{R}^p$, a number K of clusters. Euclidean distance.

K-median clustering

Output define K points as centers so as to minimize the sum of the distances between each point and its nearest center.

K-mean clustering

Output partition the points into K sets so as to minimize the sum of the distances between each point and the mean of points in its cluster.

K-mean clustering

cannot precompute the distance to the centers anymore: modeled with nonlinear constraints

x_{jk} is j assigned to cluster k ?
y_k coordinates of the center of k ?
d_{jk} distance from j to the center of k ?

$$\min \sum_{k=1}^{K} \sum_{j=1}^{n} x_{jk} d_{jk}$$

$$s.t. \quad \boxed{d_{jk} = \sum_{i=1}^{p} (m_j^i - y_k^i)^2 \quad \forall j, k}$$

$$\sum_{k=1}^{K} x_{jk} = 1 \quad \forall j$$

$$x_{jk} \in \{0,1\}, y_k^i \in \mathbb{R}, d_{jk} \geq 0$$

non convex

K-mean clustering

x_{jk} is j assigned to cluster k ?

y_k coordinates of the center of k ?

d_{jk} distance from j to the center of its cluster k ?

$$\min \sum_{k=1}^{K} \sum_{j=1}^{n} d_{jk}$$

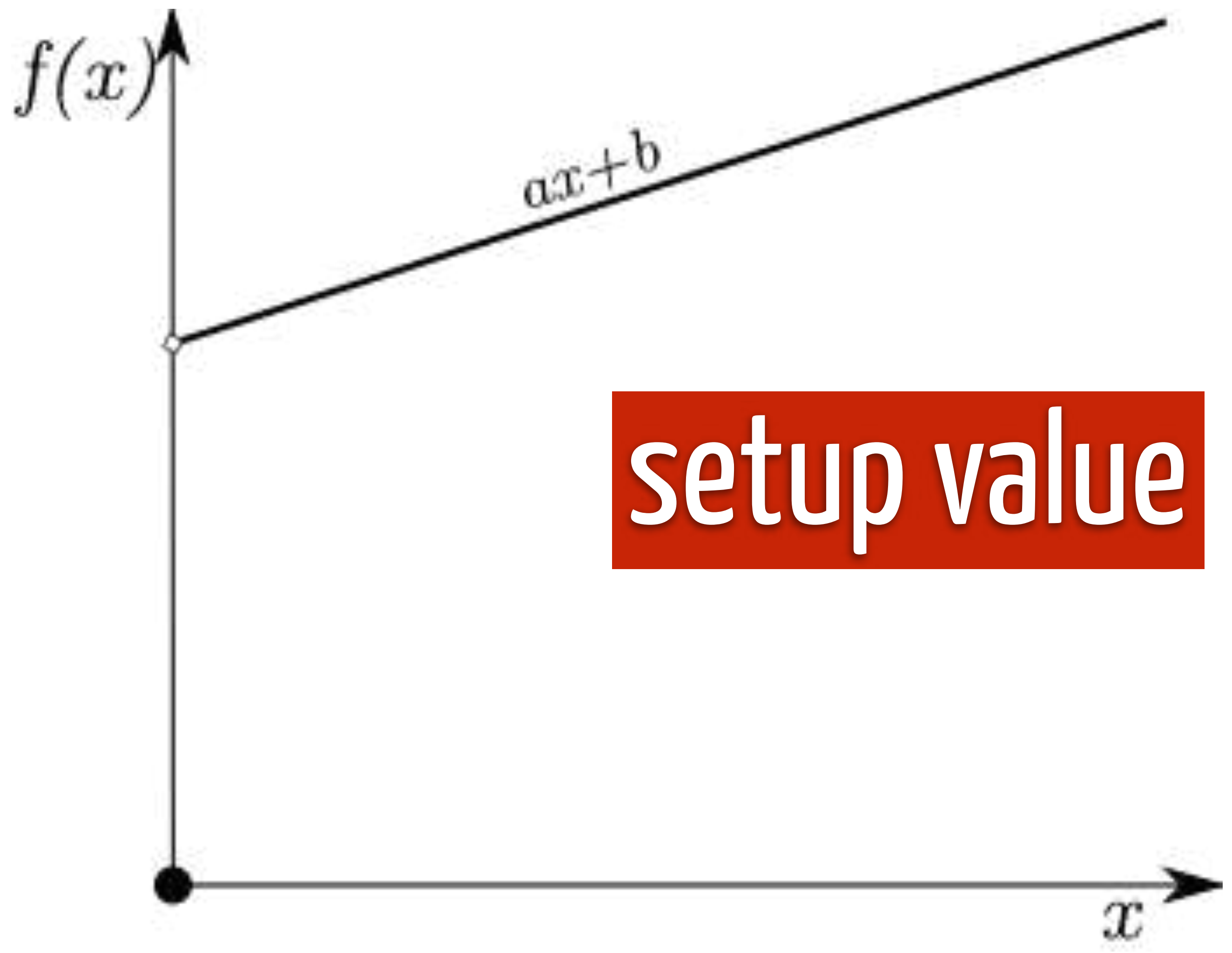
$$s.t. \; \boxed{d_{jk} \geq \sum_{i=1}^{p} (m_j^i - y_k^i)^2 - \bar{d}_{jk}(1 - x_{jk}) \quad \forall j, k}$$

$$\sum_{k=1}^{K} x_{jk} = 1 \quad \forall j$$

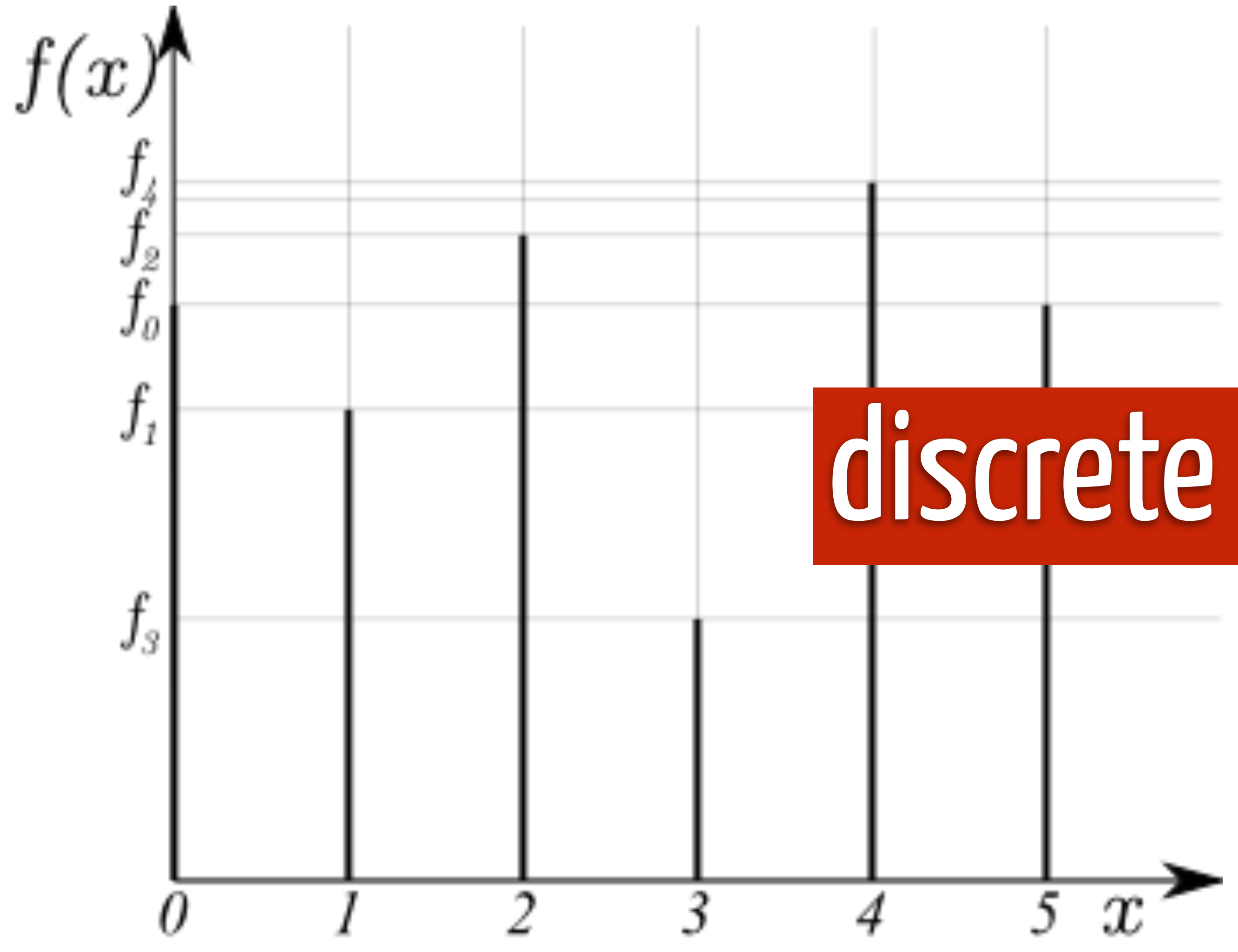
$$x_{jk} \in \{0,1\}, y_k^i \in \mathbb{R}, d_{jk} \geq 0$$

convexify the nonlinear constraints using big-M (optimization is still nonconvex because of integrality)

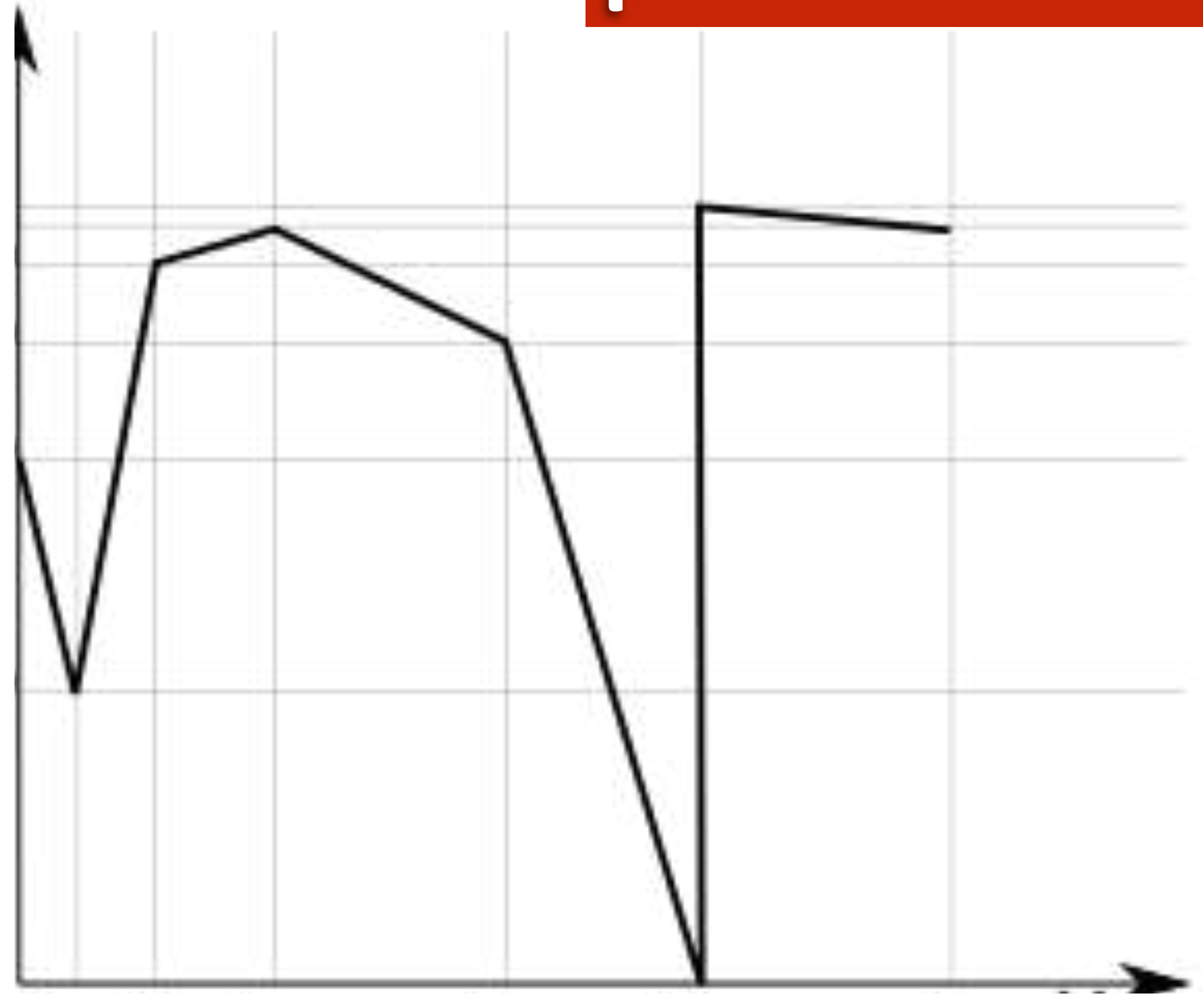
non-linear functions



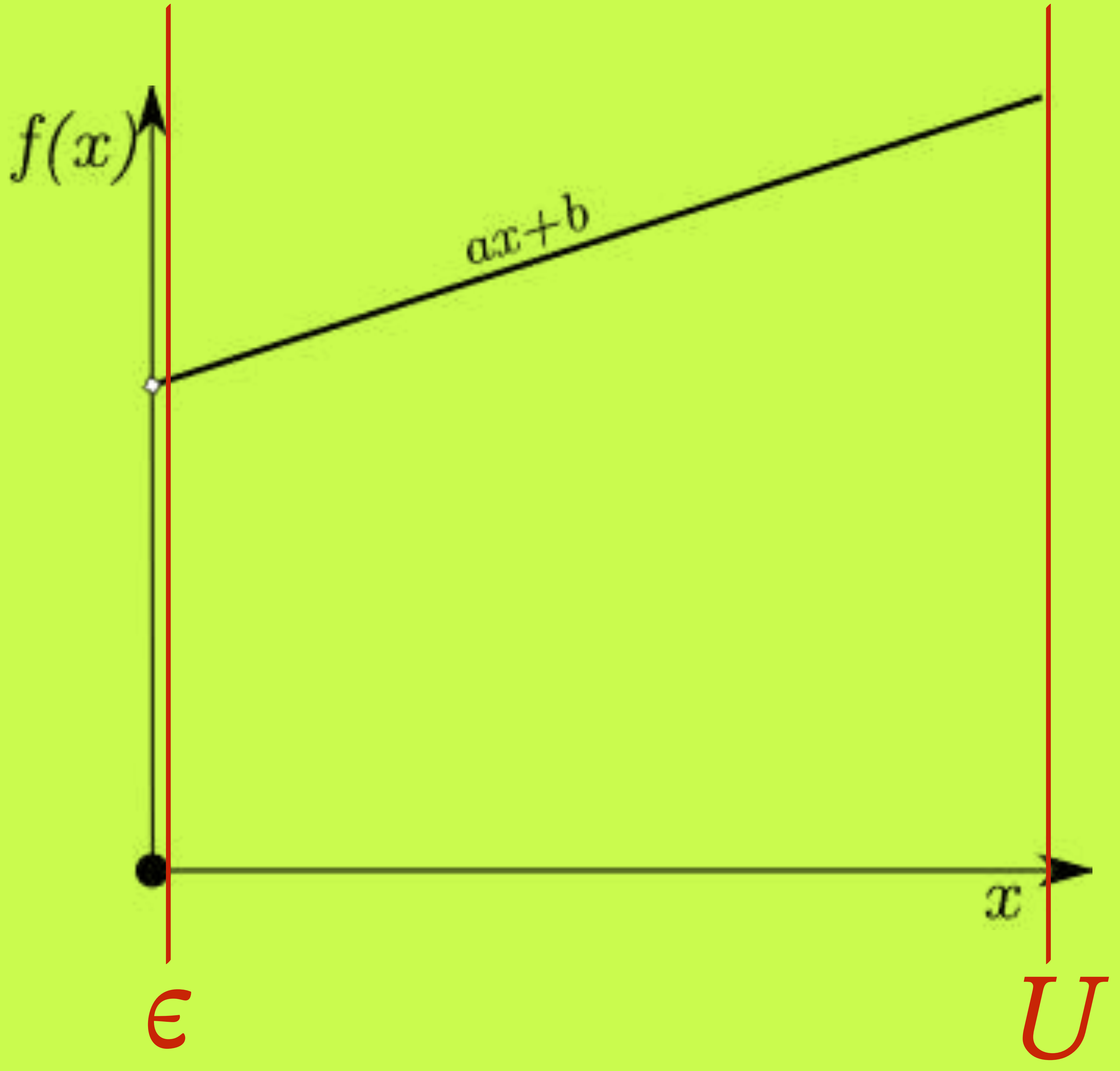
setup value



discrete values



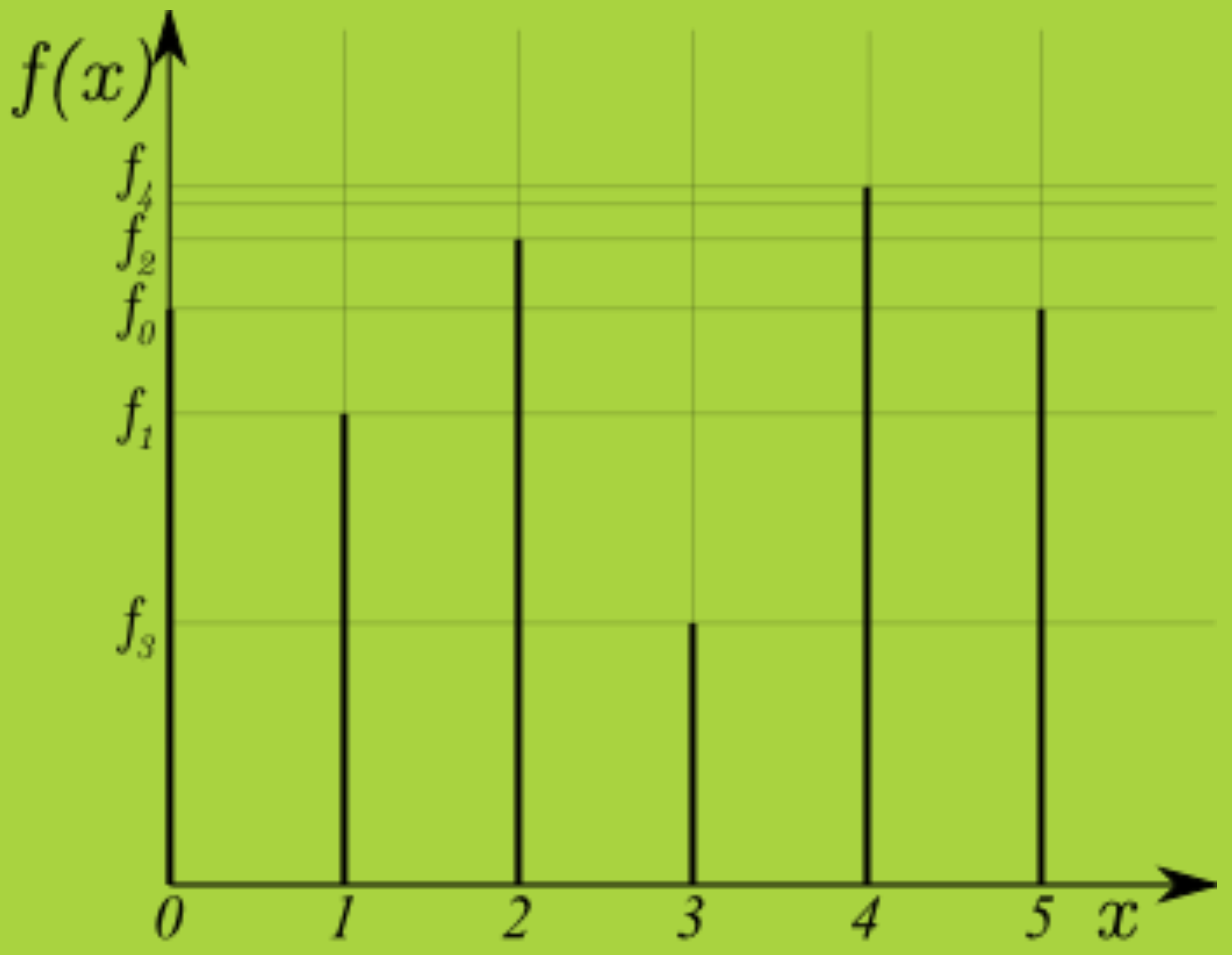
piecewise linear



setup value

$$f(x) = ax + b\delta$$
$$\epsilon\delta \leq x \leq U\delta$$
$$\delta \in \{0, 1\}$$

δ is x positive ?

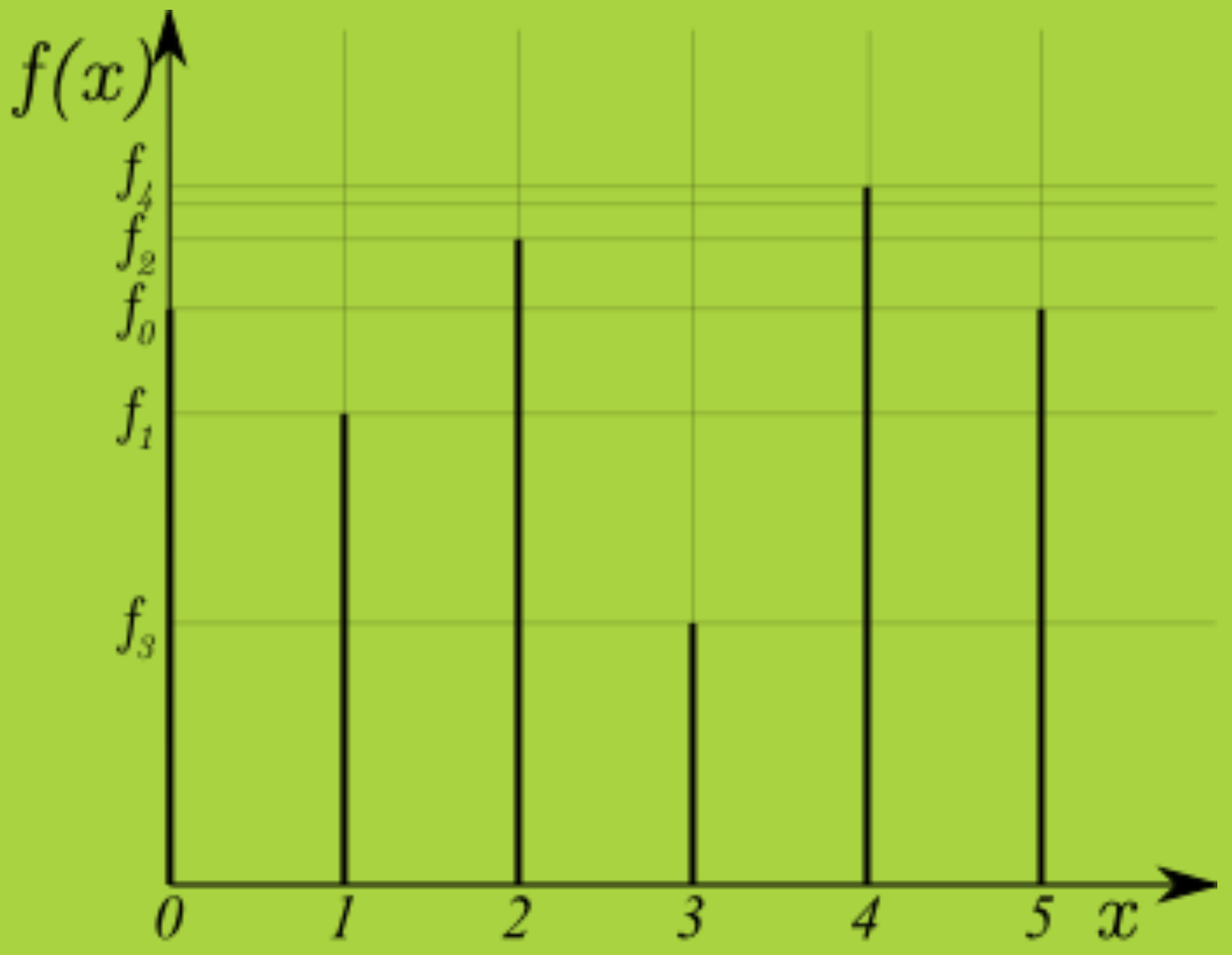


discrete values

$$f(x) = \sum_i \delta_i f_i$$
$$\sum_i i\delta_i = x$$
$$\sum_i \delta_i = 1$$
$$\delta_i \in \{0, 1\} \; i = 0..n$$

δ_i is $x{=}i$ (and $f(x){=}f_i$) ?

Special Ordered Set of type 1:
<u>ordered</u> set of variables, all zero except at most one



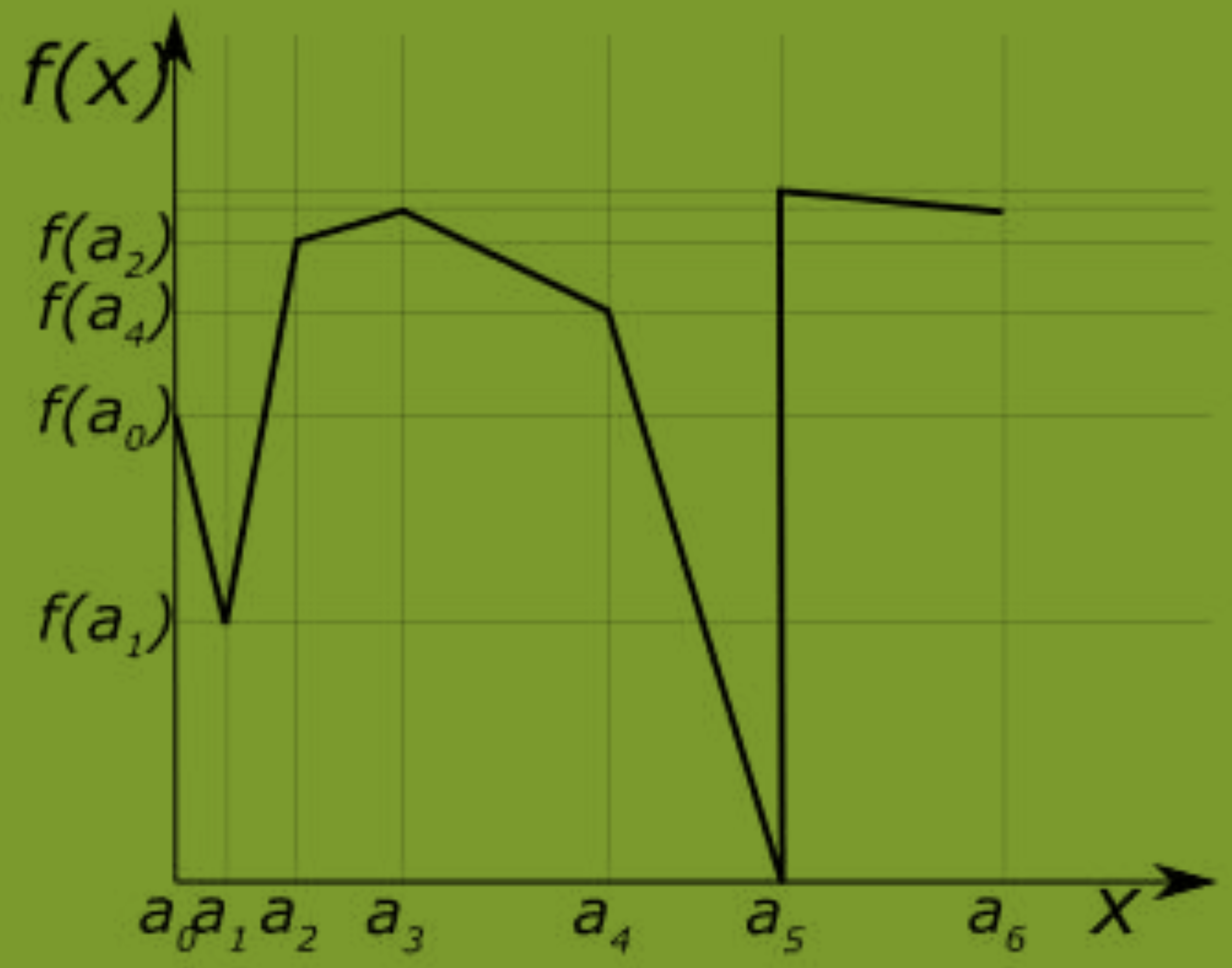
discrete values

$$f(x) = \sum_i \delta_i f_i$$
$$\sum_i i\delta_i = x$$
$$\sum_i \delta_i \geq 1$$
$$\delta_i \in \{0, 1\} \; i = 0..n$$

SOS1(δ)

δ_i is $x=i$ (and $f(x)=f_i$) ?

Special Ordered Set of type 2:
ordered set of variables, all zero except at most two consecutive



piecewise linear

$$f(x) = \sum_i \lambda_i f(a_i)$$
$$\sum_i a_i \lambda_i = x$$
$$\sum_i \lambda_i = 1$$
$$\lambda_i \in [0,1] \quad i = 0..n$$
$$SOS2(\lambda)$$

λ_i is $x=a_i$? (then $\lambda_i a_i + \lambda_{i+1} a_{i+1}$ in $[a_i, a_{i+1}]$ if $\lambda_i + \lambda_{i+1} = 1$)

modeling with $\mathbb{Z}$

$$x_i = 5$$

to order i is the 5th item

to count 5 items are selected

to measure time task i starts at time 5

to measure space item i is located on floor 5

$$\simeq \delta_{i5} = 1$$

Binary Integer Linear Program (BIP) $\{0,1\}^n$

Integer Linear Program (IP) $\mathbb{Z}^n$

Mixed Integer Linear Program (MIP) $\mathbb{Z}^n \cup \mathbb{Q}^n$

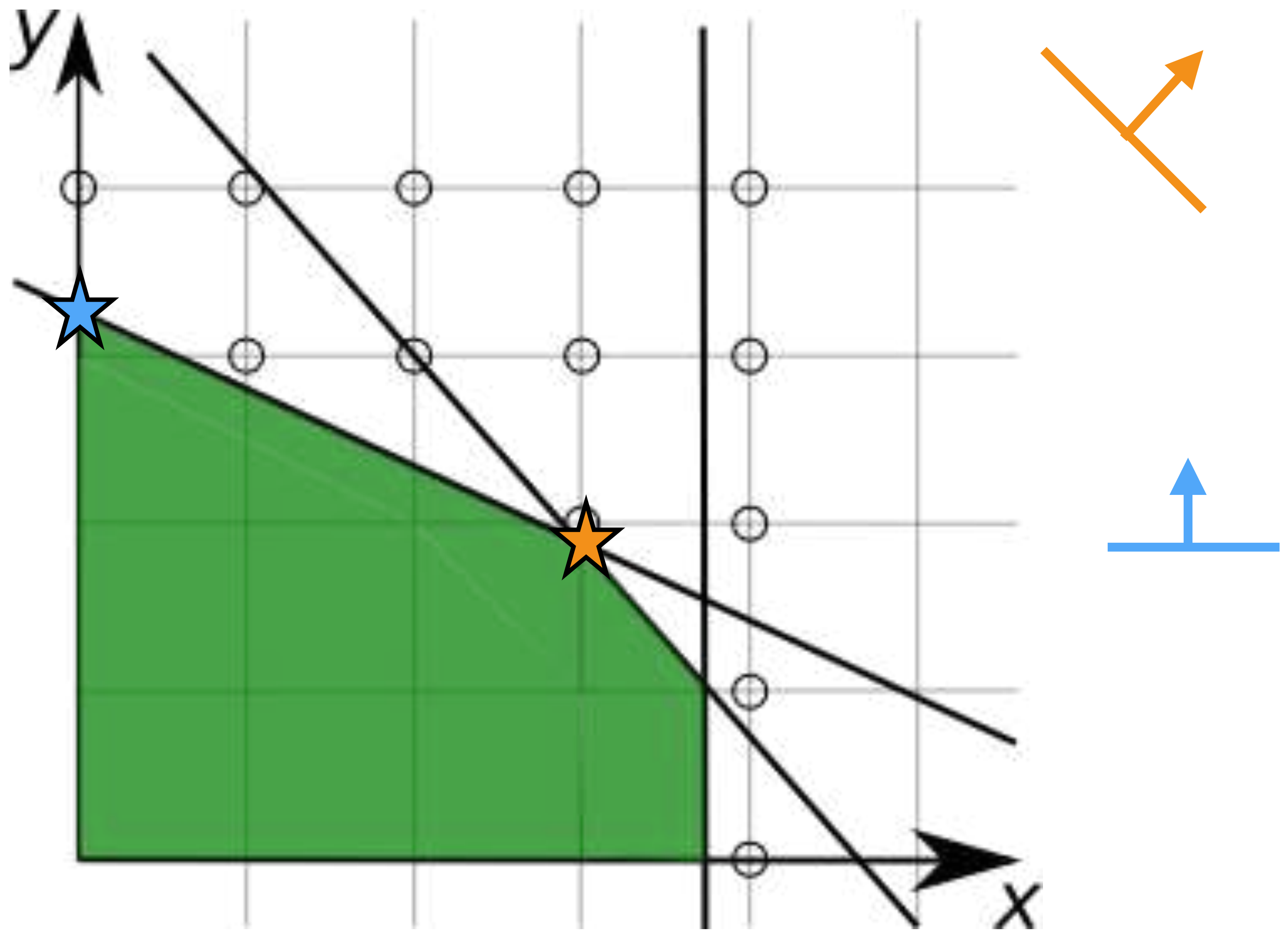
1 how to model ?

2 how difficult ?

3 how to solve ?

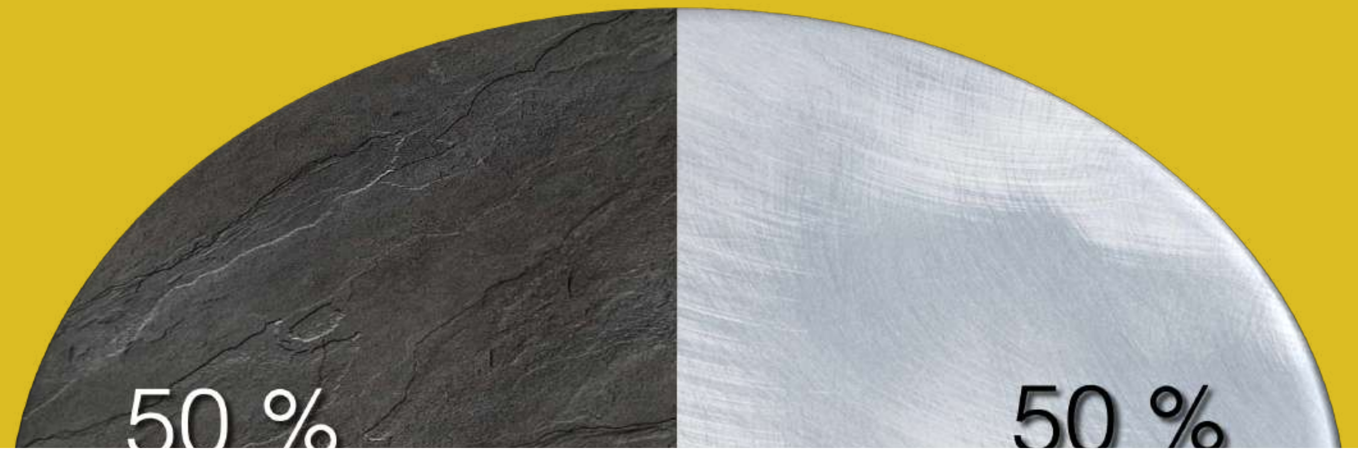
Linear Programming cheat sheet

LP is easy



- MILP without integrality = LP relaxation
- LP feasible set = polyhedron
- convex optimization
- if LP is feasible and bounded, at least one vertex is optimal
- primal simplex algorithm: visit adjacent vertices as cost decreases
- strong duality: $\min\{cx \mid Ax \geq b, x \geq 0\} = \max\{ub \mid uA \leq c, u \geq 0\}$
- interior point method runs in polynomial time (simplex can be better in practice)

Market Split Problem



$$\min \sum_{j=1}^{m} s_j^+ + s_j^-$$

$$\text{s.t.} \sum_{i=1}^{n} a_{ij} x_i + s_j^+ - s_j^- = \frac{d_j}{2} \qquad j = 1..m$$

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

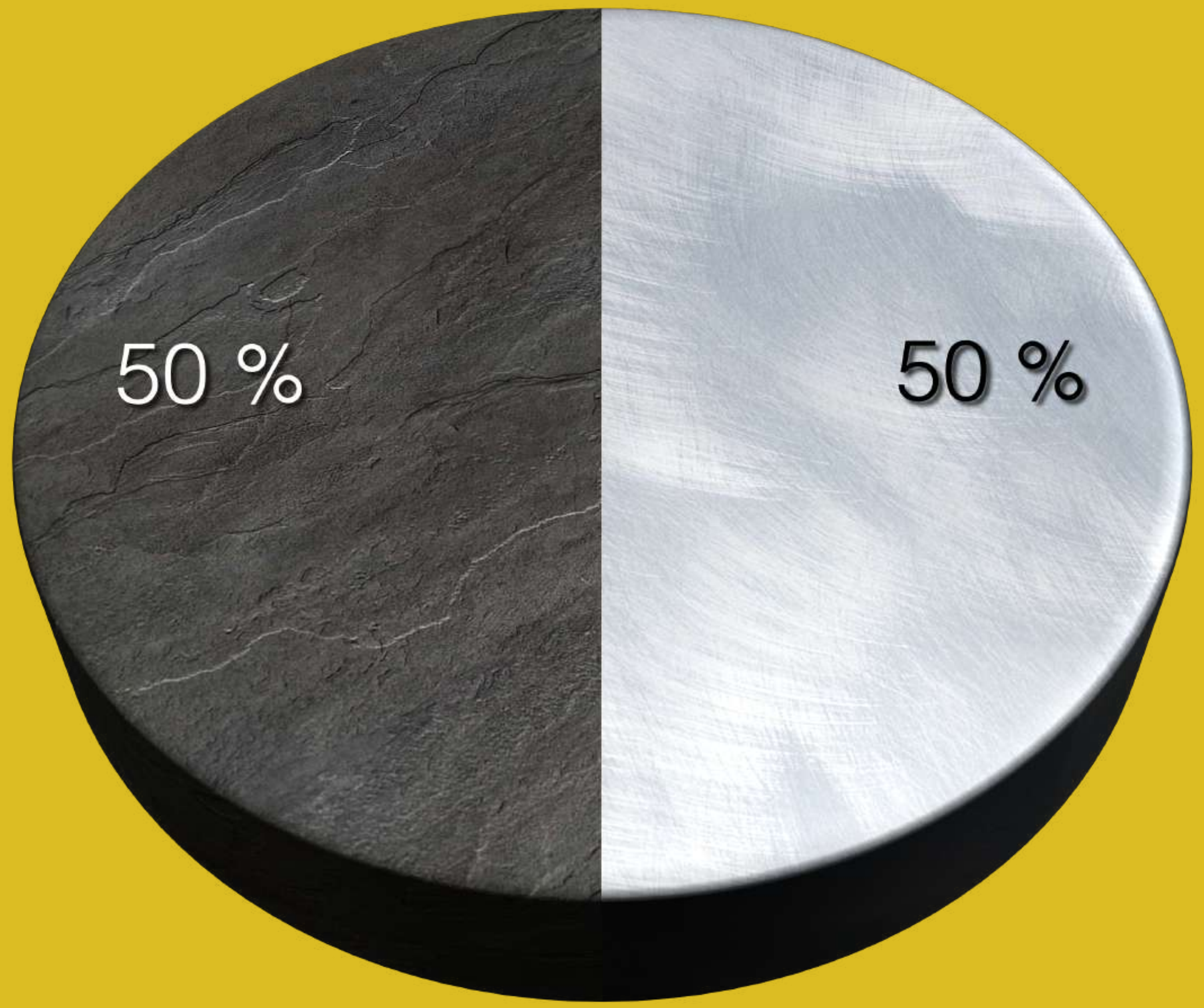
$$s_j^+ \geq 0, s_j^- \geq 0 \qquad j = 1..m$$

Input 1 company, 2 divisions, m products with availabilities d_j, n retailers with demands a_{ij} in each product j.

Output an assignment of the retailers to the divisions approaching a 50/50 production split.

x_i is retailer i assigned to division 1 ?

s_j gap to the 50% split goal for product j

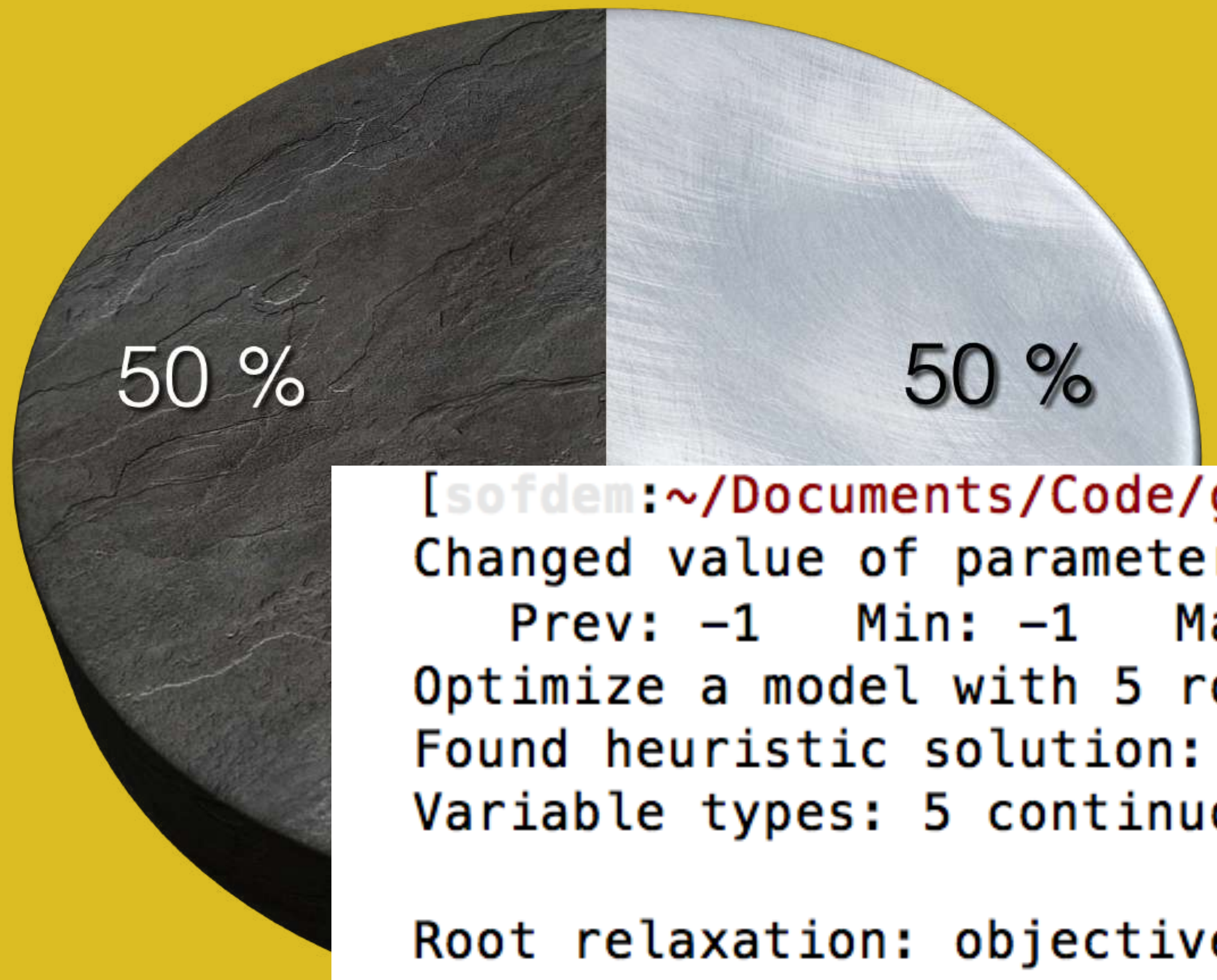


MIPLIB markshare_5_0

Input 5 products, 40 retailers
Output . . . . (hold the line please) . . . .

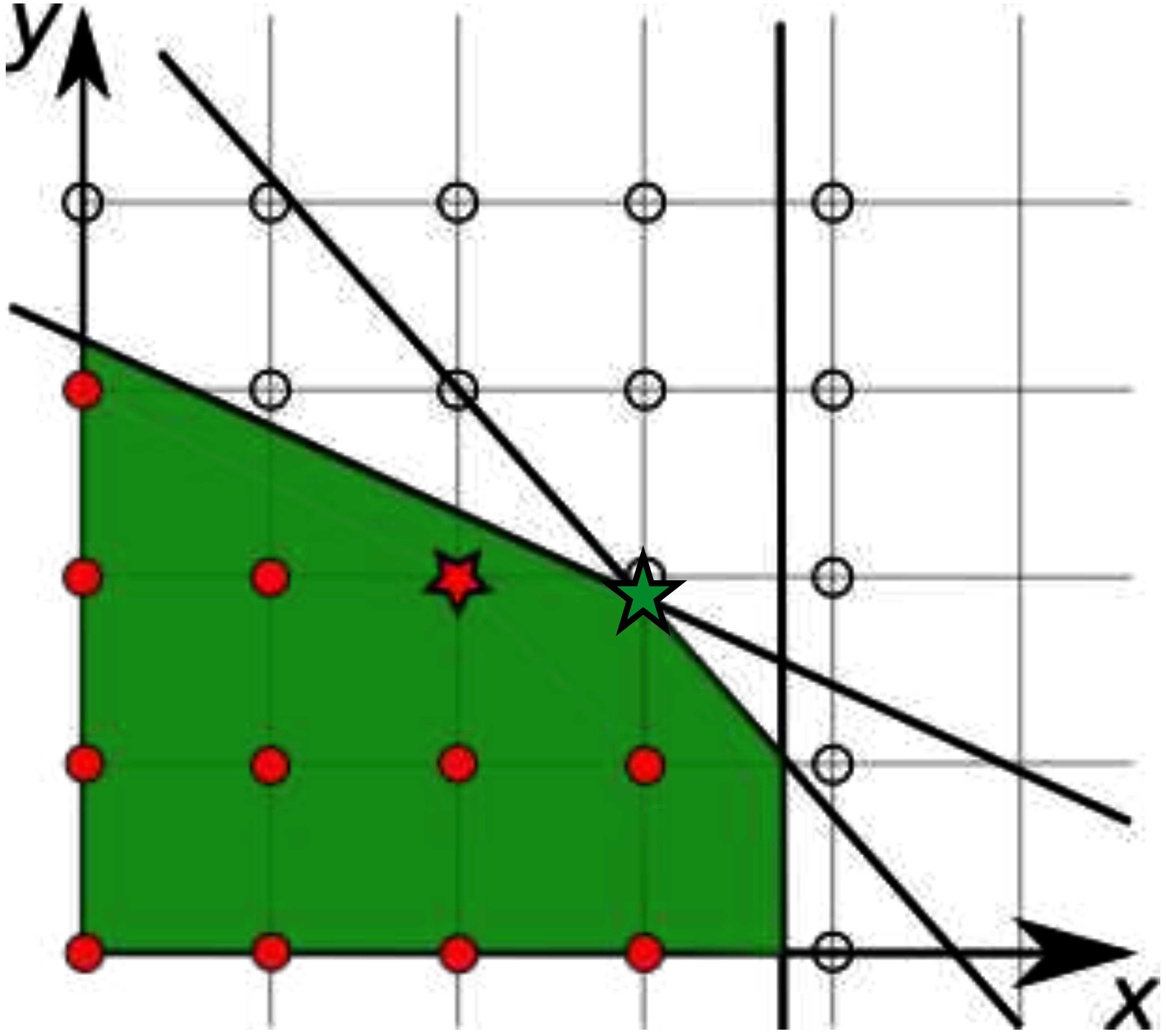
Int Opt = 1
Solution time = 20 minutes
Proof time = > 1 hour

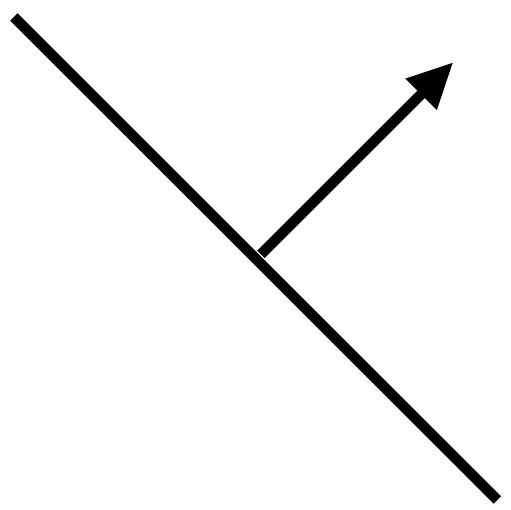
MIPLIB markshare_5_0



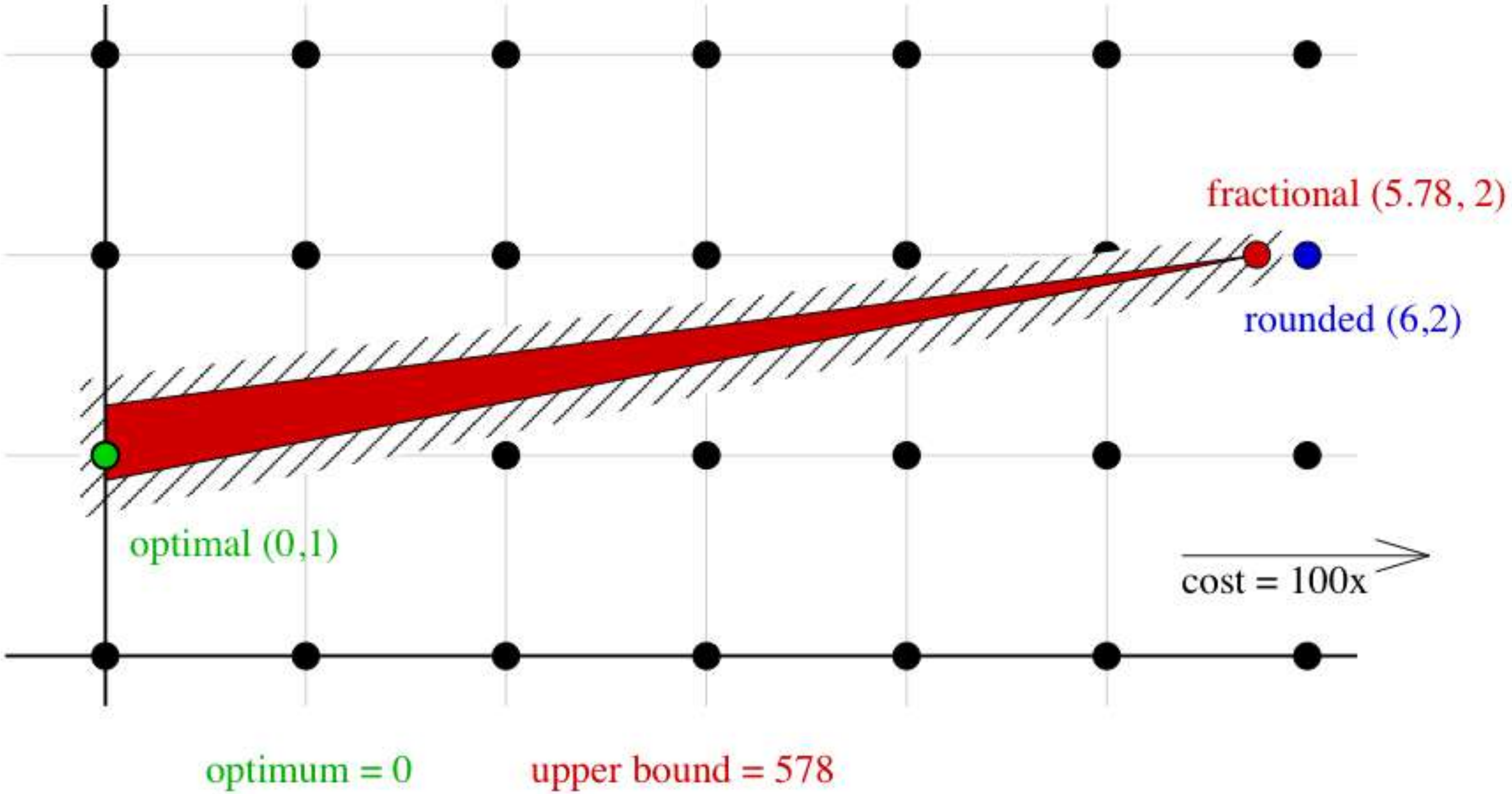

```
[sofdem:~/Documents/Code/gurobi]$ gurobi.sh mymip.py markshare_5_0.mps.gz
Changed value of parameter Presolve to 0
   Prev: -1   Min: -1   Max: 2   Default: -1
Optimize a model with 5 rows, 45 columns and 203 nonzeros
Found heuristic solution: objective 5335
Variable types: 5 continuous, 40 integer (40 binary)

Root relaxation: objective 0.000000e+00, 15 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.00000    0     5 5335.00000    0.00000  100%     -    0s

*62706364 28044              38           1.0000000    0.00000   100%   2.1 1241s

Explored 233848403 nodes (460515864 simplex iterations) in 3883.56 seconds
Thread count was 4 (of 4 available processors)

Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0%
Optimal objective: 1
```

ILP ≠ LP relaxation





ILP ≠ round LP relaxation



general ILP is NP-hard

small problems are easy
some specific problems are easy

1||Cmax Scheduling Problem

$$
\begin{aligned}
&\min\ s_{n+1} \quad = p_1+\ldots+p_n \\
&\text{s.t. } s_{n+1} \geq s_j + p_j & j = 1..n \\
&\quad s_j - s_i \geq M x_{ij} + (p_i - M) & i,j = 1..n \\
&\quad x_{ij} + x_{ji} = 1 & i,j = 1..n; i < j \\
&\quad s_j \in \cancel{\mathbb{Z}_+} \ \geq 0 & j = 1..n+1 \\
&\quad x_{ij} \in \{0,1\} & i,j = 1..n
\end{aligned}
$$

Input n tasks, duration p_i for each task i, 1 machine
Output a minimal makespan schedule of the tasks on the machine without overlap

Capacitated Transhipment Problem

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{j\in\delta^+(i)} x_{ij} - \sum_{j\in\delta^-(i)} x_{ij} = b_i & i \in V \\
& x_{ij} \leq h_{ij} & (i,j) \in A \\
& x_{ij} \in \cancel{\mathbb{Z}_+} \geq 0 & (i,j) \in A
\end{aligned}
$$

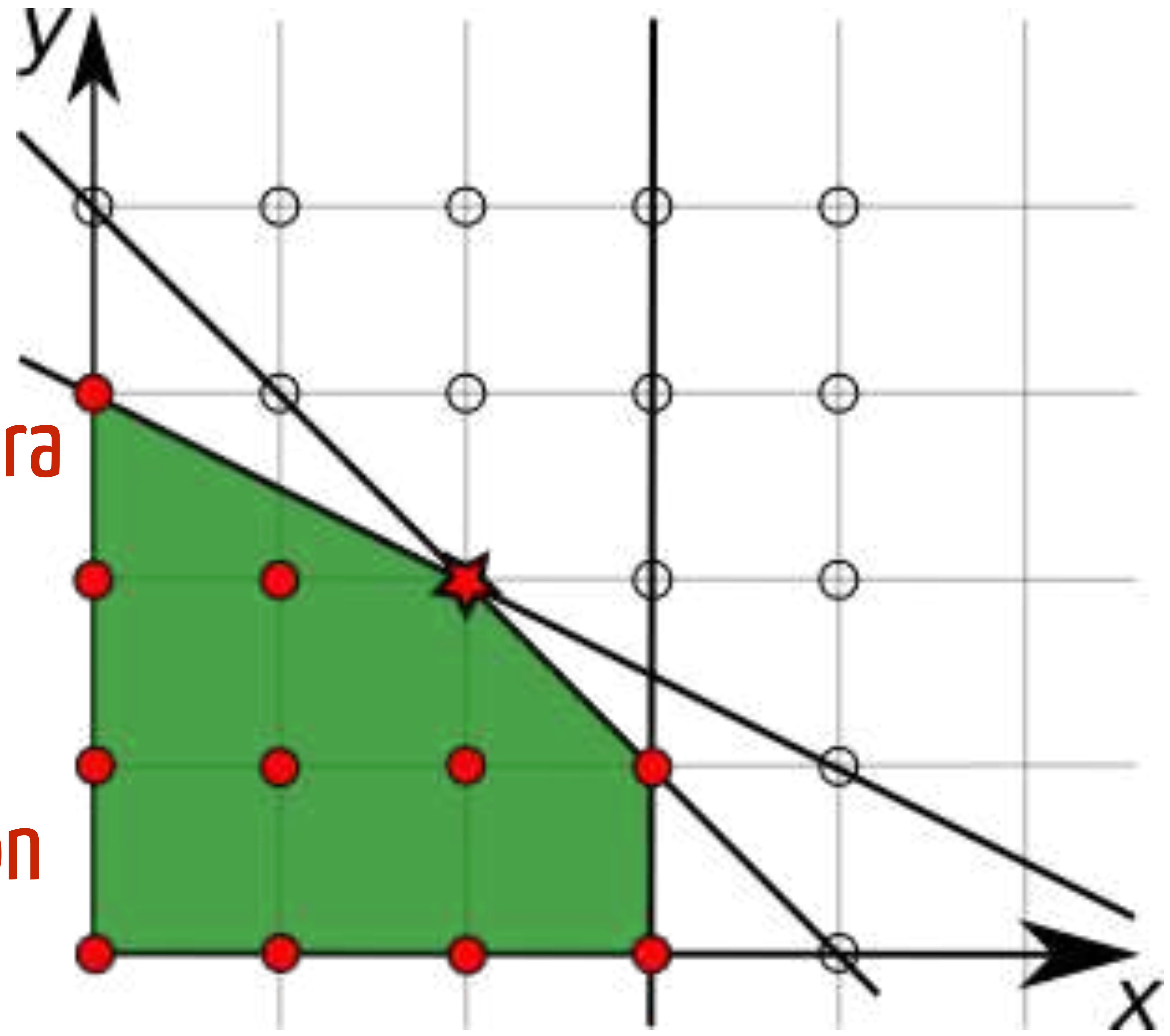
Input digraph (V,A), demand or supply b_i at each node i, capacity h_{ij} and unit flow cost c_{ij} for each arc (i,j)

Output a mimimum cost integer flow to satisfy the demand

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

LP = ILP sometimes

integral polyhedra
=
convex hull
=
ideal formulation



totally unimodular matrix
(theory)

$$(P) = \max\{\ cx \mid Ax \leq b, x \in \mathbb{Z}^n_+\ \}$$

- basic feasible solutions of the LP relaxation $(\bar{P})$ take the form: $\bar{x} = (\bar{x}_B, \bar{x}_N) = (B^{-1}b, 0)$ where B is a square submatrix of (A, I_m)
- Cramer's rule: $B^{-1} = B^*/det(B)$ where B^* is the adjoint matrix (made of products of terms of B)
- Proposition: if (P) has integral data (A, b) and if $det(B) = \pm 1$ then $\bar{x}$ is integral

Definition

A matrix A is totally unimodular (TU) if every square submatrix has determinant $+1$, -1 or 0.

Proposition

If A is TU and b is integral then any optimal solution of $(\bar{P})$ is integral.

totally unimodular matrix
(practice)

How to recognize TU ?

Sufficient condition

A matrix A is TU if

- all the coefficients are $+1$, -1 or 0
- each column contains at most 2 non-zero coefficient
- there exists a partition (M_1, M_2) of the set M of rows such that each column j containing two non zero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.

Proposition

A is TU $\iff$ A^t is TU $\iff$ (A, I_m) is TU
where A^t is the transpose matrix, I_m the identitiy matrix

Interlude

Show that the Transhipment ILP is ideal

Show that the Scheduling ILP is NOT ideal