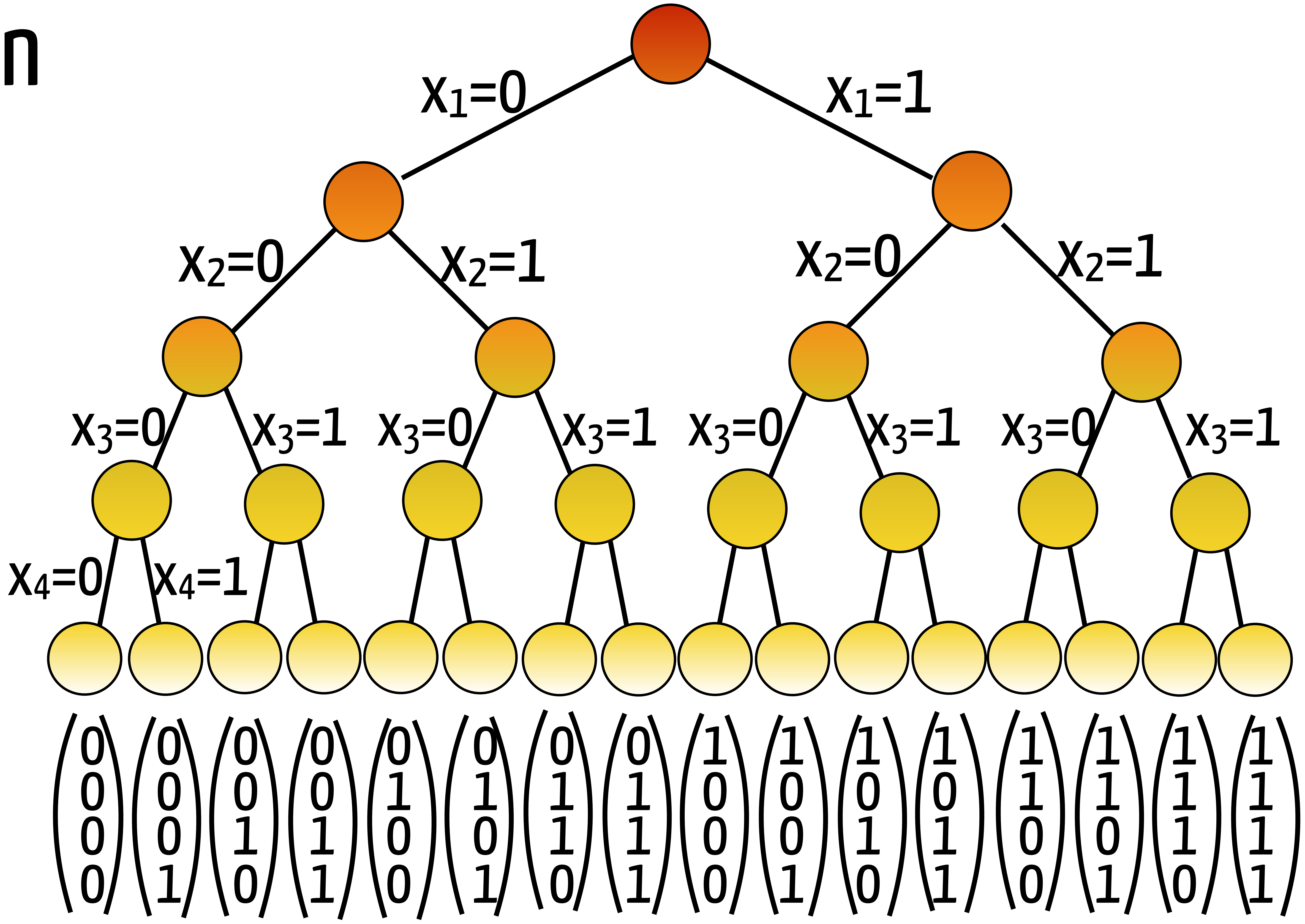
1 how to model ?

2 how difficult ?

3 how to solve ?

Complete enumeration

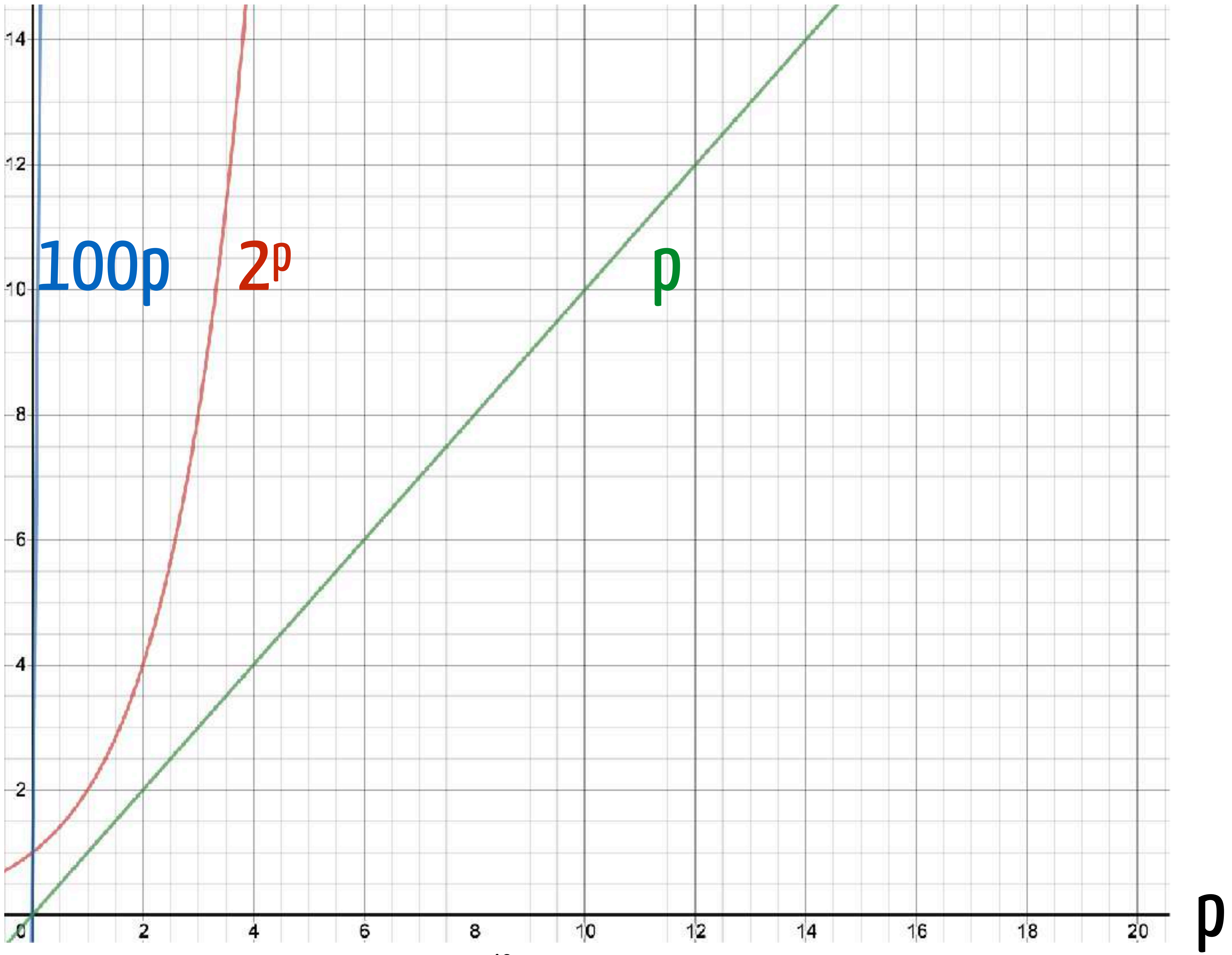


MILP with p binaries

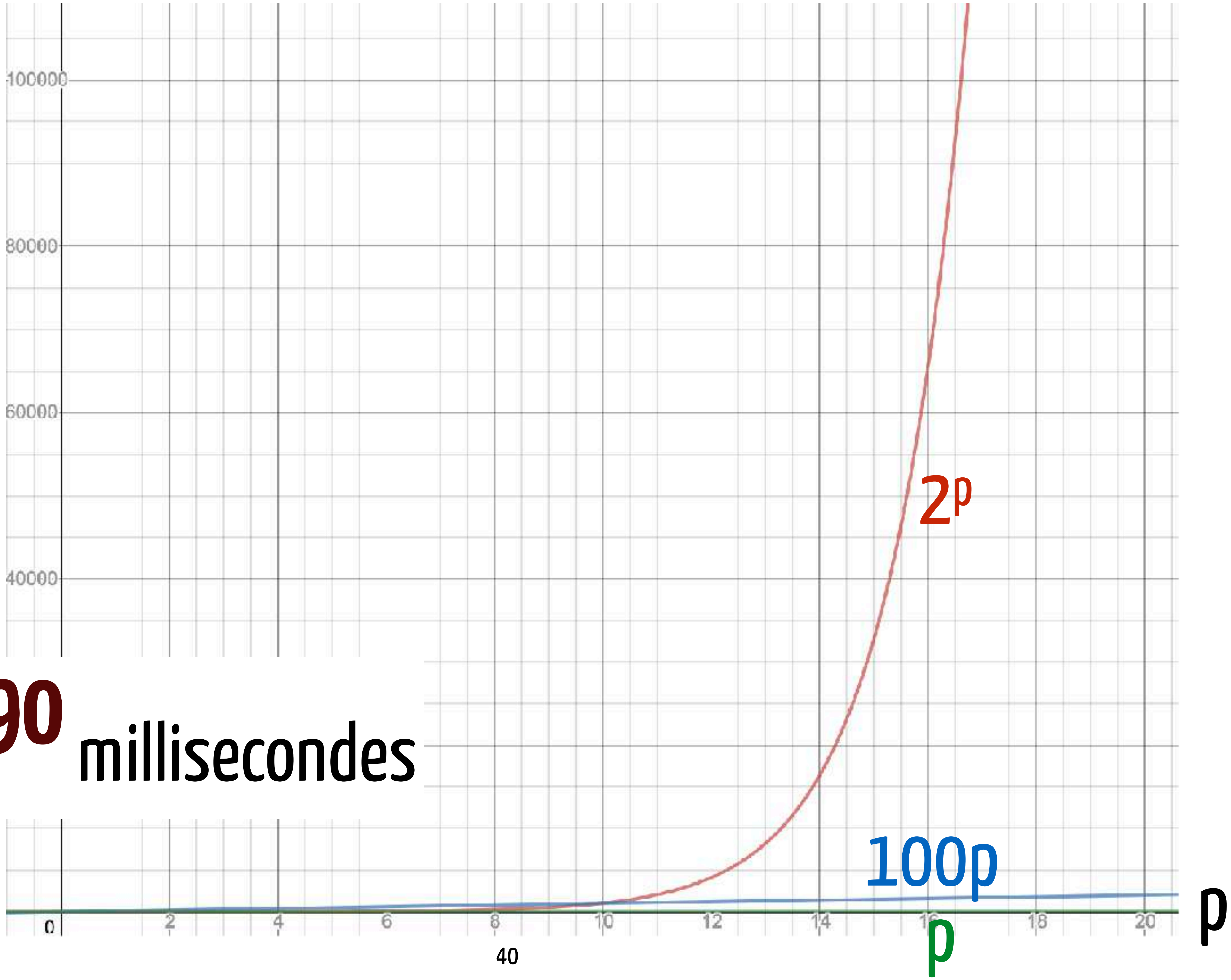
$$\min\{cx \mid Ax \geq b, x \in \{0,1\}^p \times \mathbb{R}^{n-p}\}$$

= 2^p LPs to solve

Combinatorial explosion



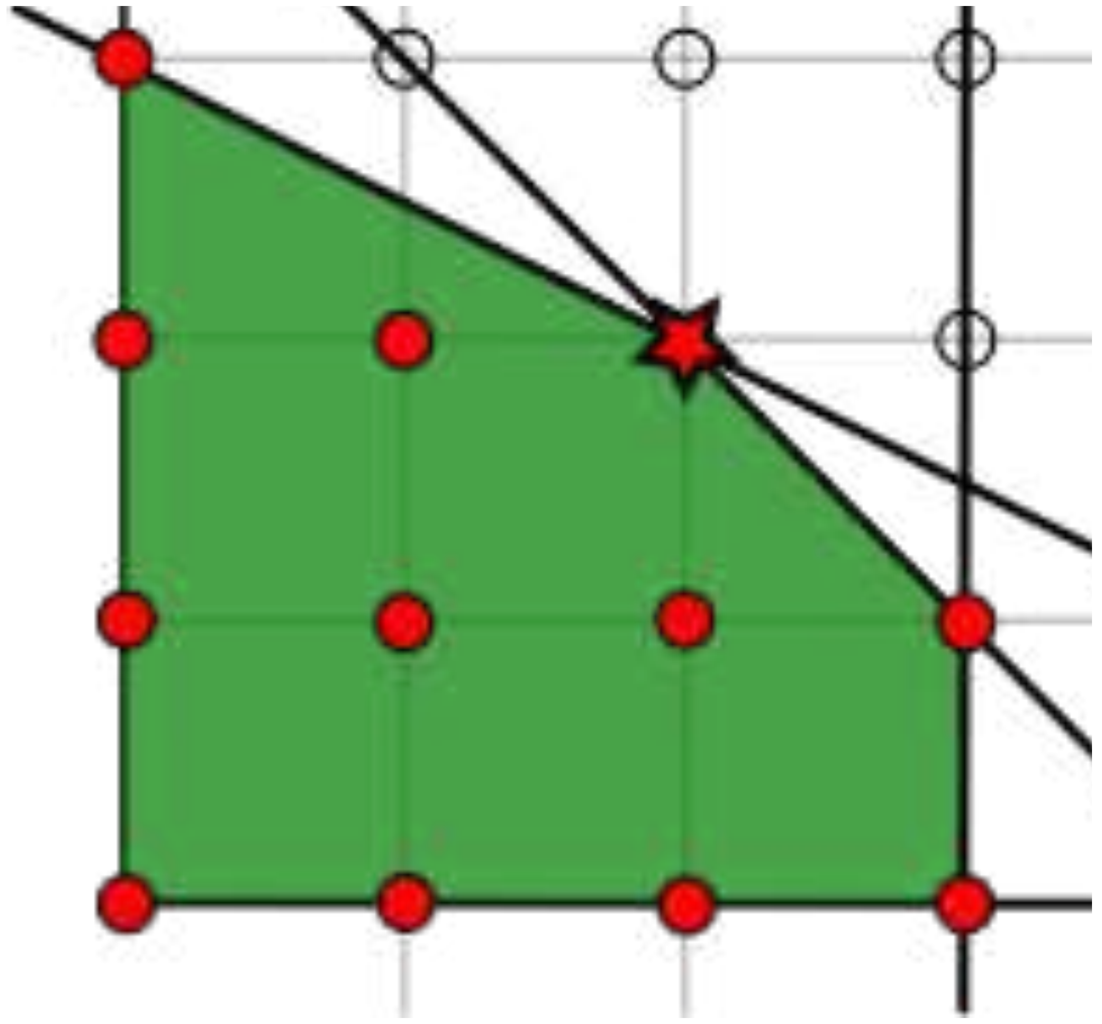
Combinatorial explosion



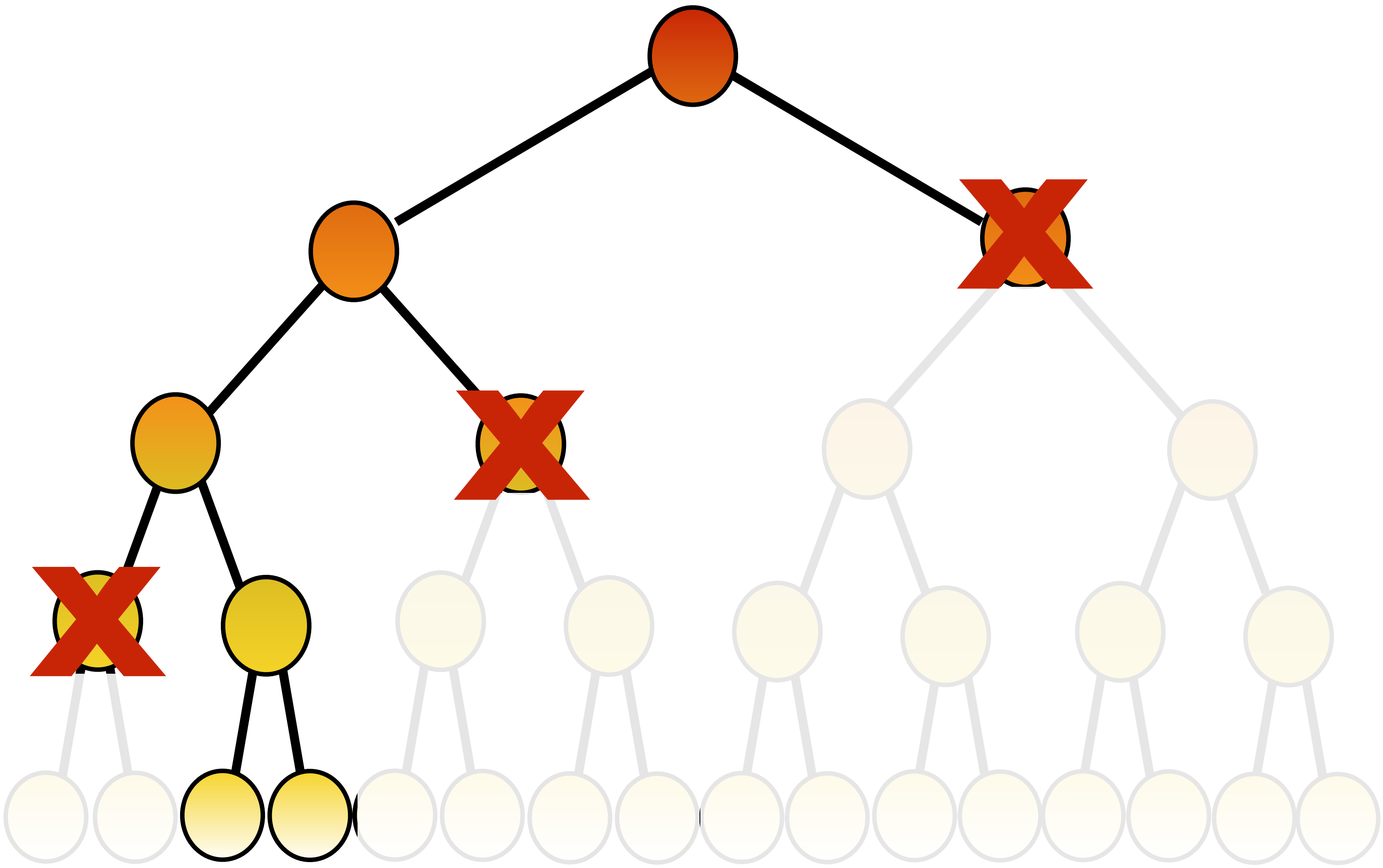
âge de l'univers ≈ 2^{90} millisecondes

Two options

1 compute an ideal formulation



2 evaluate partial solutions progressively

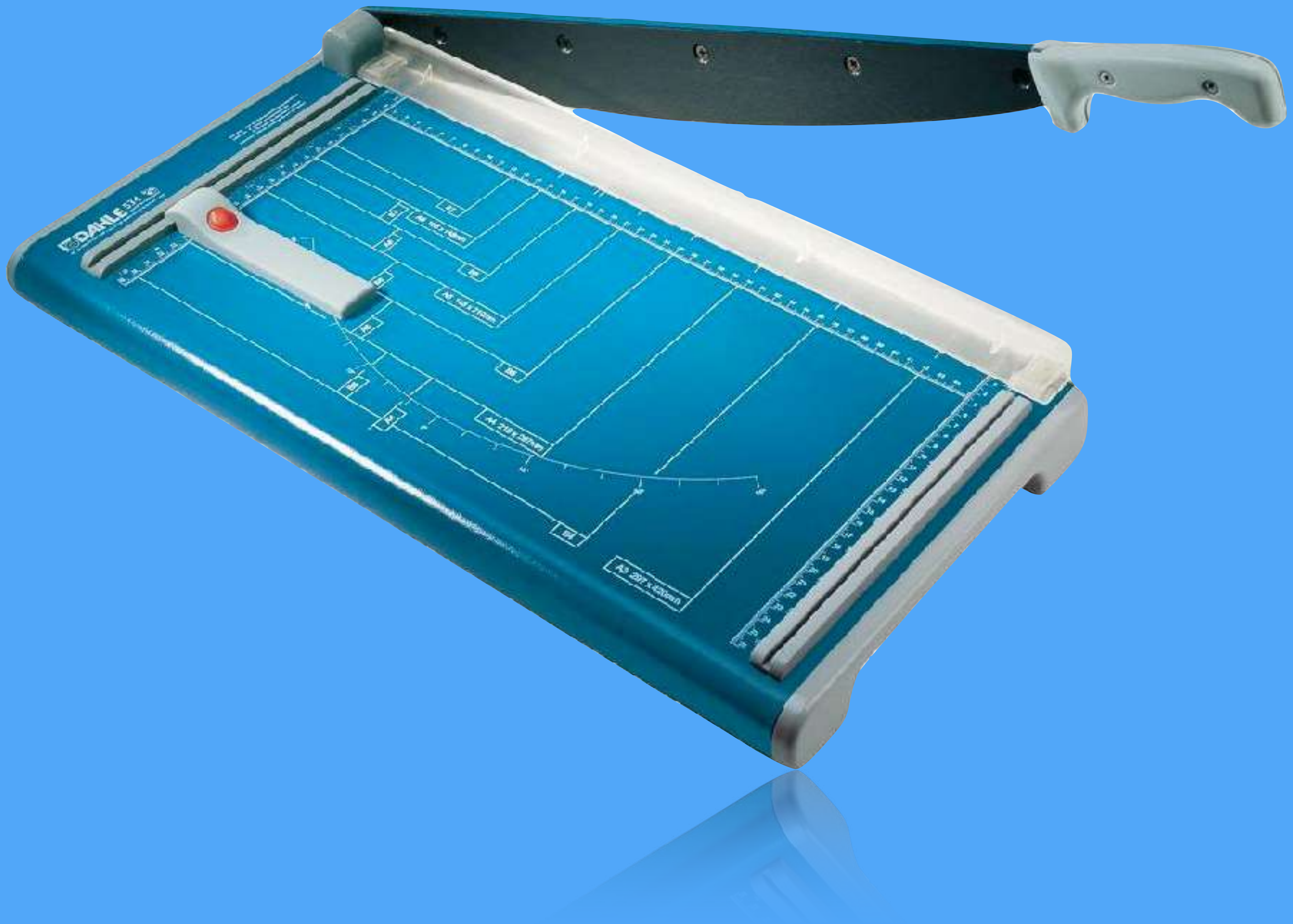


1 Cut Generation compute an ideal formulation

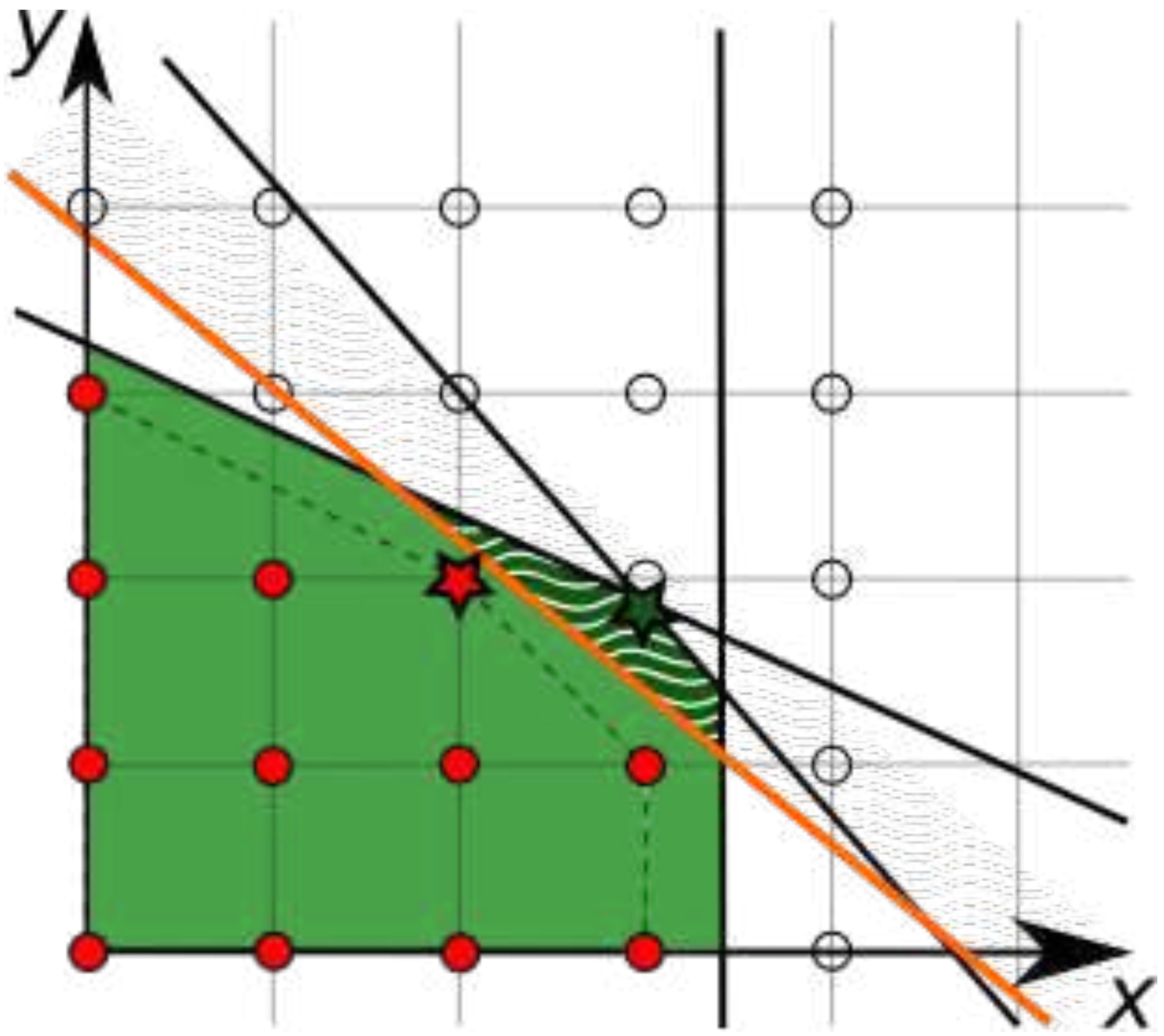
2 Branch&Bound evaluate partial solutions progressively

3 modern Branch&Cut mix up+presolve+heuristics

4 decomposition methods (Branch&Price, Lagrangian relaxation, Benders)



Cutting Plane Algorithm



Cut valid inequality that separates a relaxed LP solution

Farkas Lemma cuts are linear combinations of constraints

cutting plane algorithm

1. solve the LP relaxation of (P), get $\bar{x}$
2. if $\bar{x}$ is integral STOP:
   feasible then optimal for (P)
3. find cuts C for (P,$\bar{x}$) from template T

separation subproblem

4. add constraints C to (P) then 1.

templates

general-purpose

mixed integer rounding, split, Chvátal-Gomory

structure-based

clique, cover, flow cover, zero half

problem-specific

subtour elimination (TSP), odd-set (matching)

ex 1 Chvátal-Gomory cuts

$(P) : \max\{cx \mid Ax \leq b, x \in \mathbb{Z}_+\}$

For any $u \in \mathbb{R}^m_+$ the following inequalities are valid:

1. surrogate: $\sum_j \sum_i u_i a_{ij} x_j \leq \sum_i u_i b_i$ $\quad (u \geq 0)$
2. round off: $\sum_j \lfloor \sum_i u_i a_{ij} \rfloor x_j \leq \sum_i u_i b_i$ $\quad (x \geq 0)$
3. Chvátal-Gomory: $\sum_j \lfloor \sum_i u_i a_{ij} \rfloor x_j \leq \lfloor \sum_i u_i b_i \rfloor$ $\quad (\lfloor uA \rfloor x \in \mathbb{Z})$

variants in the choice of u, ex: Gomory or MIR cuts

ex 2 Cover cuts

$$S = \{y \in \{0,1\}^7 | 11y_1 + 6y_2 + 6y_3 + 5y_4 + 5y_5 + 4y_6 + y_7 \leq 19\}$$

- (y_3, y_4, y_5, y_6) is a minimal cover for $11y_1 + 6y_2 + 6y_3 + 5y_4 + 5y_5 + 4y_6 + y_7 \leq 19$ as $6 + 5 + 5 + 4 > 19$ then $y_3 + y_4 + y_5 + y_6 \leq 3$ is a cover inequality
- we can derive a stronger valid inequality $y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 3$ by noting that y_1, y_2 has greater coefficients than any variable in the cover
- note furthermore that (y_1, y_i, y_j) is a cover $\forall i \neq j \in \{2,3,4,5,6\}$ then $2y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \leq 3$ is also valid

lifting

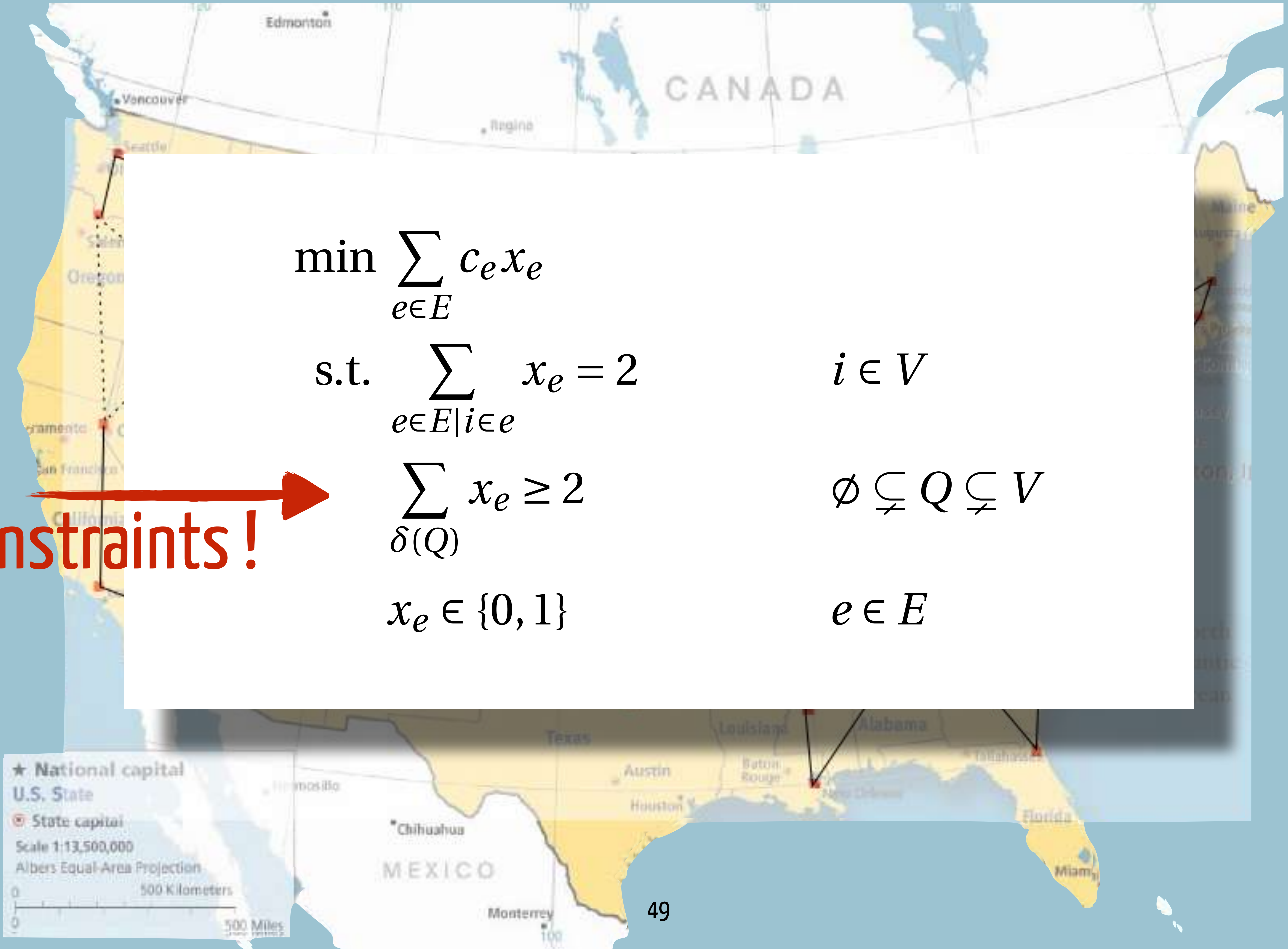
separation: solve knapsack $\min\{\sum_j (1 - \bar{y}_j)x_j \mid \sum a_j x_j \geq b + \epsilon, x \in \{0,1\}^n\}$

get coefficients x^* of the cover inequality $\sum x_j^* y_j \leq \sum x_j^* - 1$

if $\sum (1 - \bar{y}_j)x_j^* < 1$ then it is a cut (not satisfied by current LP solution $\bar{y}$)

ex 3 Subtour for TSP

ex 3 Subtour for TSP



$$\min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in E | i \in e} x_e = 2 \qquad i \in V$$

$$\sum_{\delta(Q)} x_e \geq 2 \qquad \emptyset \subsetneq Q \subsetneq V$$

$$x_e \in \{0, 1\} \qquad e \in E$$

2^n constraints !

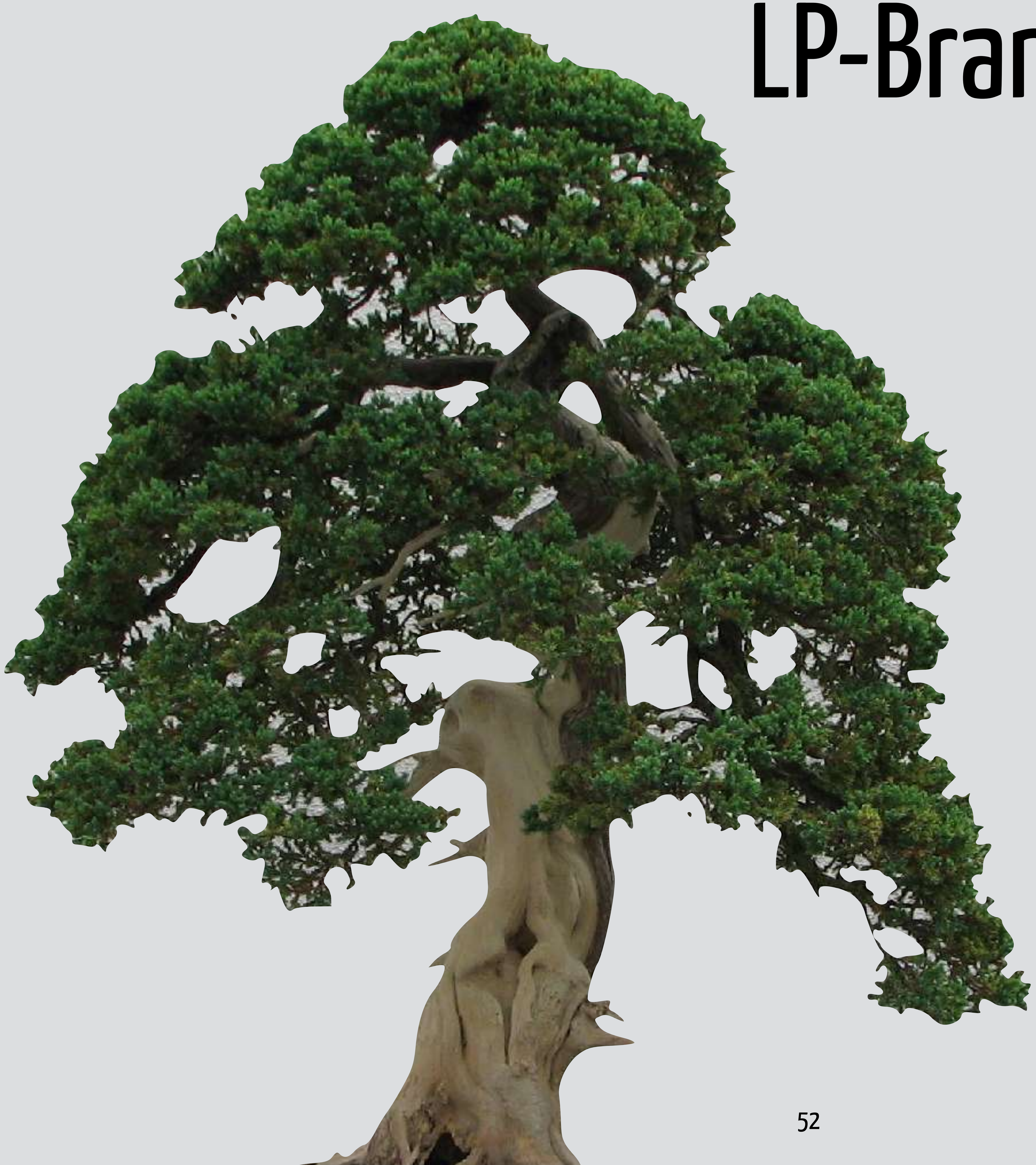
ex 3 Subtour for TSP

separation: solve min s-t cut in $(V, \overrightarrow{E}, \bar{x})$ for some fixed s and each $t \in V \backslash \{s\}$ to find a cutset $\delta(Q)$ of capacity < 2 or prove none exists

limits depending on the templates

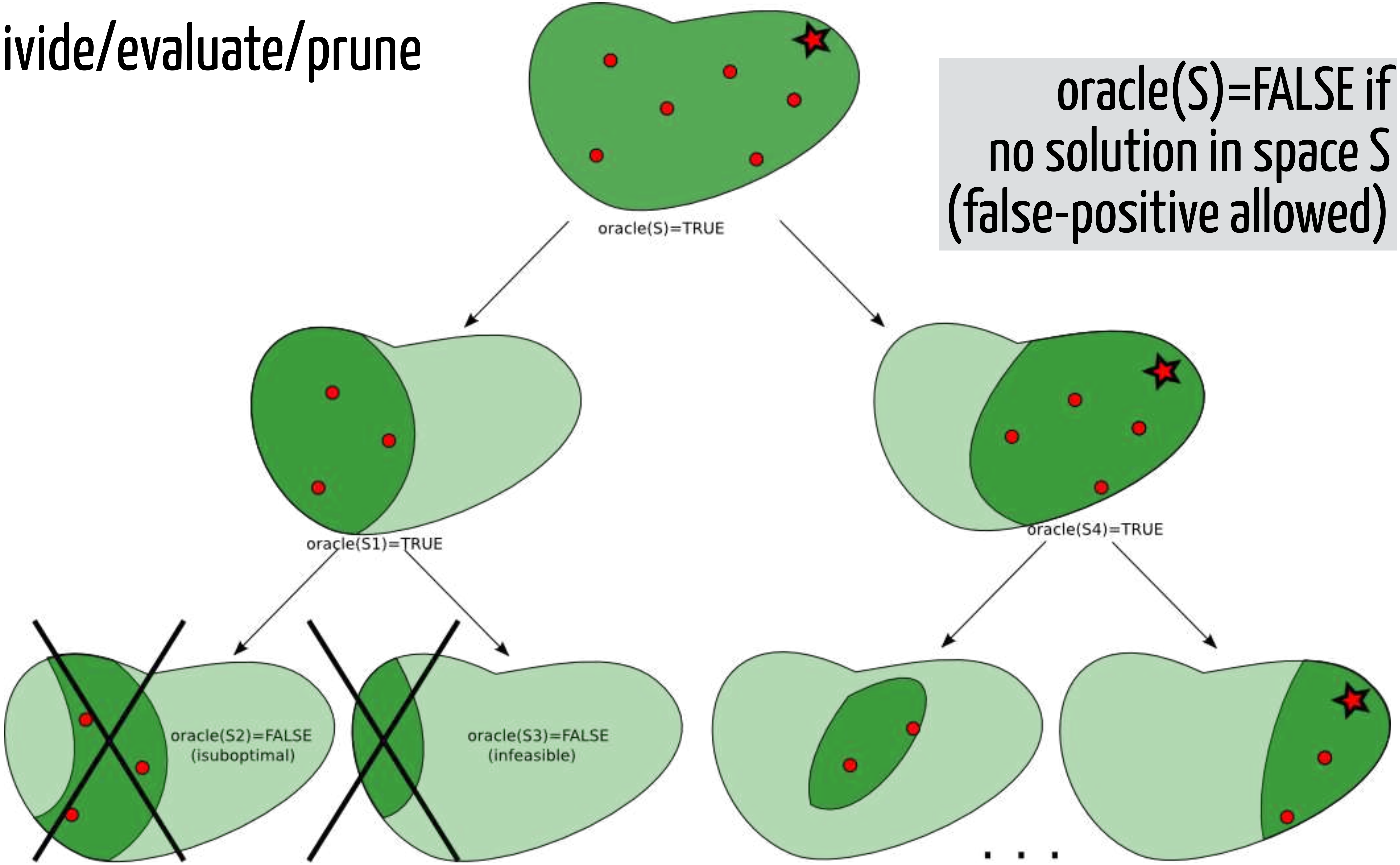
- the algorithm may stop prematurely
- the algorithm may not converge
- the algorithm may converge slowly
- the separation procedure may be NP-hard
- the LP relaxation grows
- the LP relaxation structure changes

LP-Branch and Bound



Search tree

divide/evaluate/prune

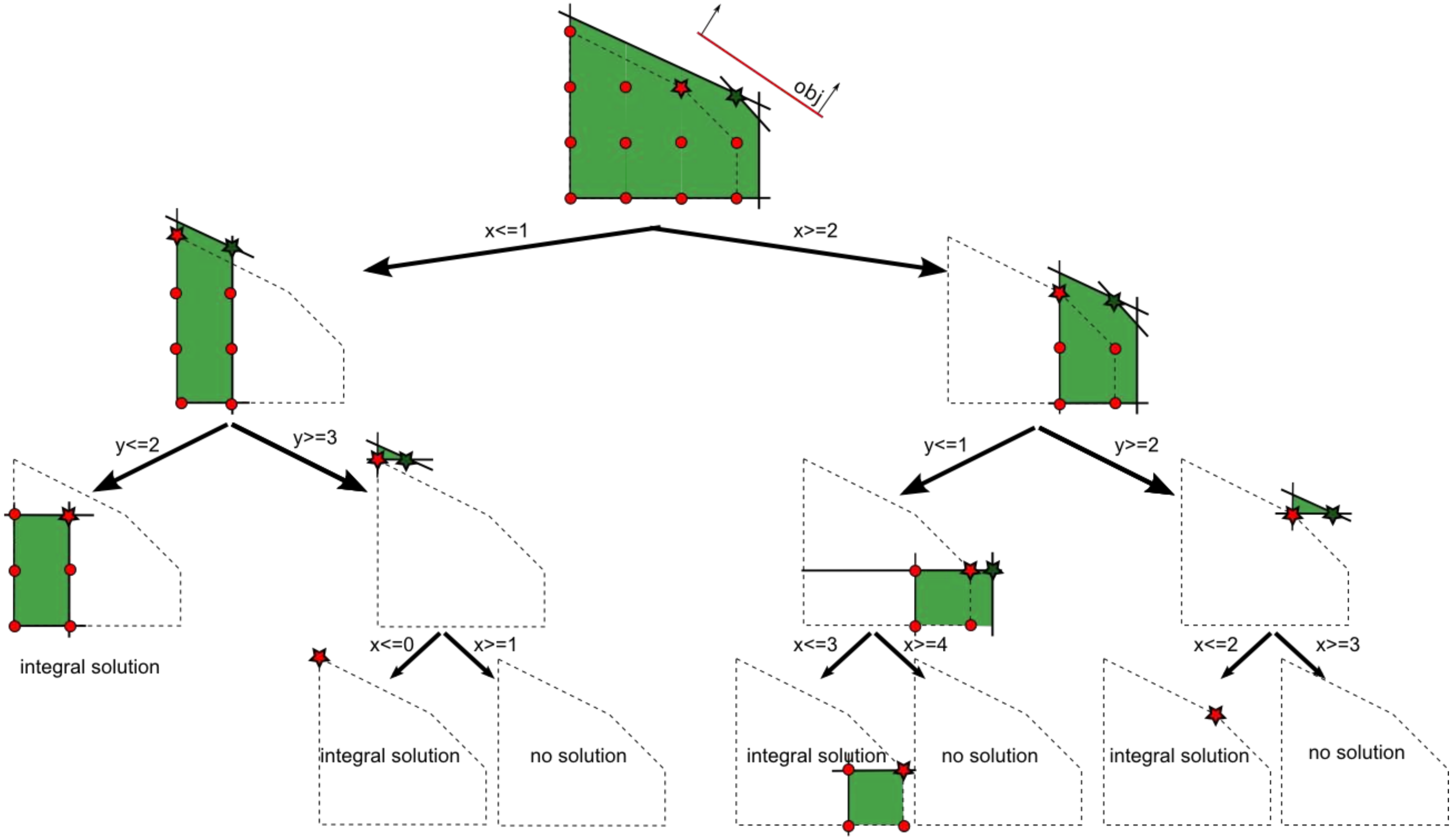


oracle(S)=FALSE if
no solution in space S
(false-positive allowed)

LP-based branch and bound

1. evaluate by solving the LP relaxation and compare bounds
2. divide with variable bounding (hyperplanes)

```
oracle(S) = FALSE if either:
 – the LP relaxation is unfeasible on S
 – the relaxed LP solution x̄ is not
   better than the best integer solution
   found so far x*
 – x̄ is integer (then update x*)
```


obj
x<=1
x>=2
y<=2
y>=3
y<=1
y>=2
integral solution
x<=0
x>=1
x<=3
x>=4
x<=2
x>=3
integral solution
no solution
integral solution
no solution
integral solution
no solution

branching

node selection

which order to visit nodes ?

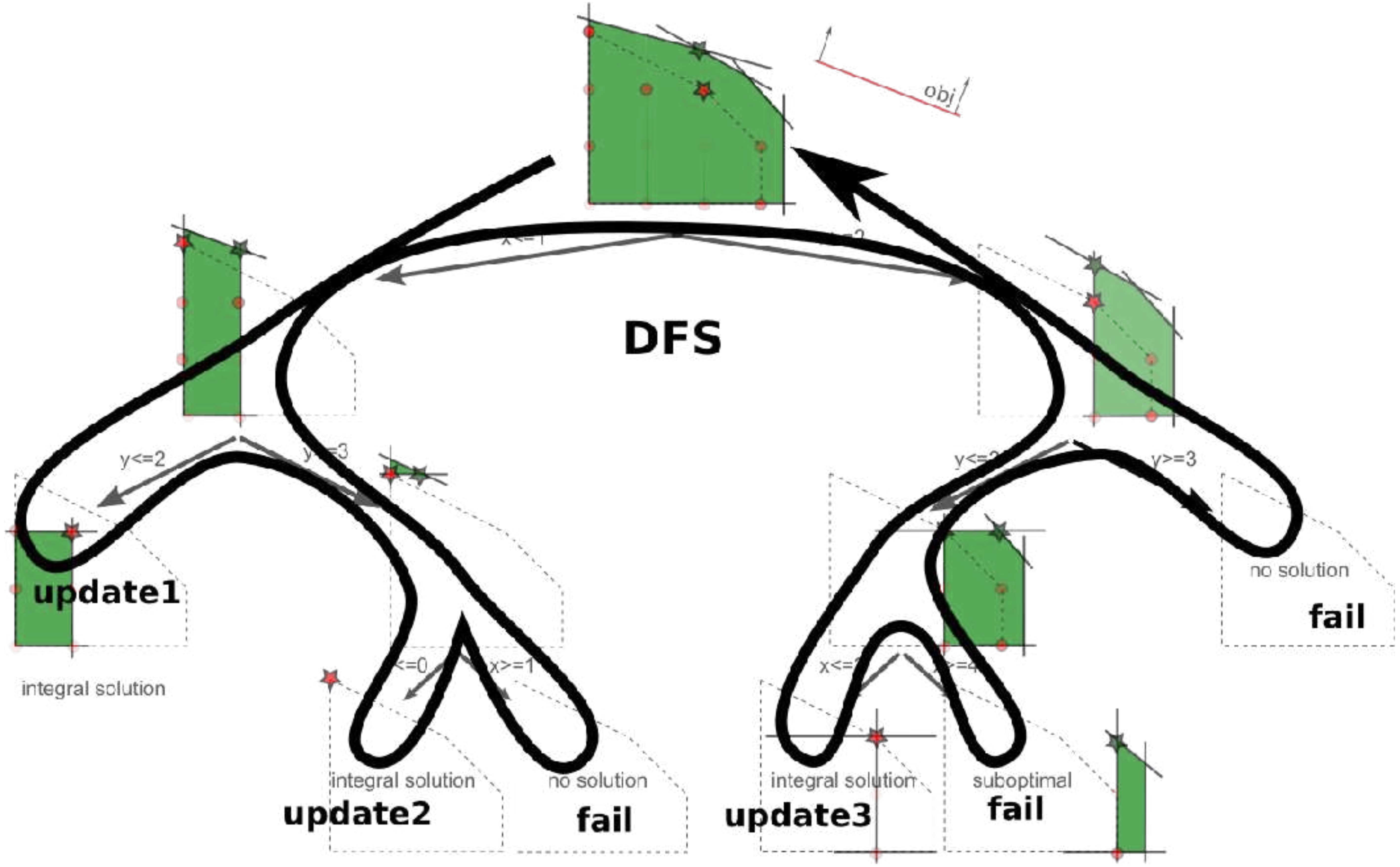
variable selection

how to separate nodes ?

constraint branching

versus variable branching

node selection

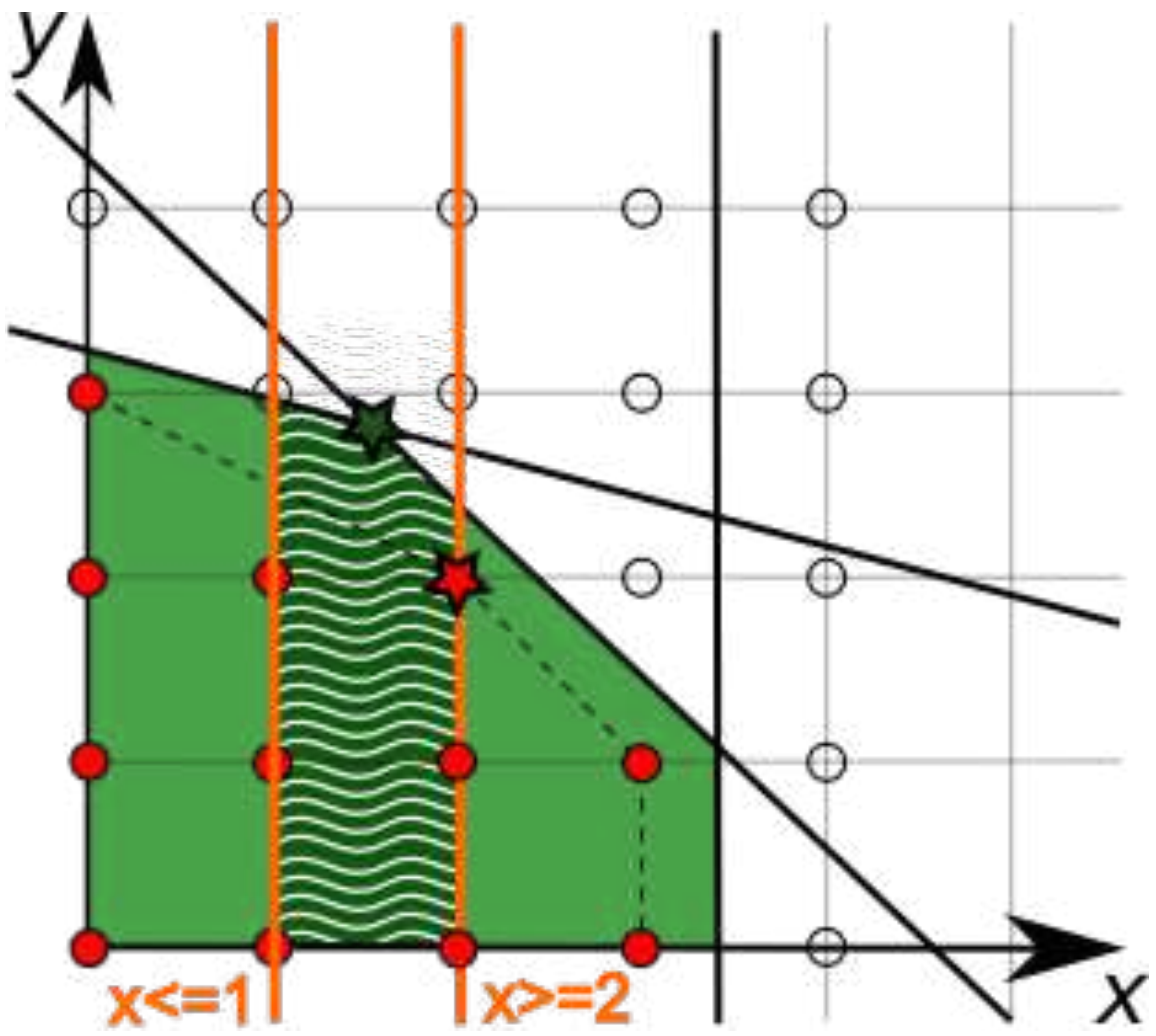


Best Bound First Search explore less nodes, manages larger trees

Depth First Search sensible to bad decisions at or near the root

DFS (up to n solutions) + **BFS** (to prove optimality)

variable selection



most fractional easy to implement but not better than random

strong branching best improvement among all candidates (impractical)

pseudocost branching record previous branching success for each var (inaccurate at root)

reliability branching pseudocosts initialised with strong branching

constraint branching

example: GUB dichotomy

- if (P) contains a GUB constraint $\sum_C x_i = 1, x \in \{0,1\}^n$
- choose $C' \subseteq C$ s.t. $0 < \sum_{C'} \bar{x}_i < 1$
- create two child nodes by setting either $\sum_{C'} x_i = 0$ or $\sum_{C'} x_i = 1$

- enforced by fixing the variable values
- leads to more balanced search trees

SOS1 branching in a facility location problem

choose a warehouse depending on its size/cost:

$$\begin{aligned} \text{COST} &= 100x_1 + 180x_2 + 320x_3 + 450x_4 + 600x_5 \\ \text{SIZE} &= 10x_1 + 20x_2 + 40x_3 + 60x_4 + 80x_5 \\ \text{(SOS1)} &\ : x_1 + x_2 + x_3 + x_4 + x_5 = 1 \end{aligned}$$

- let $\bar{x}_1 = 0.35$ and $\bar{x}_5 = 0.65$ in the LP solution then SIZE$= 55.5$
- choose $C' = \{1,2,3\}$ in order to model SIZE≤ 40 or SIZE≥ 60

modern solvers

LP solver

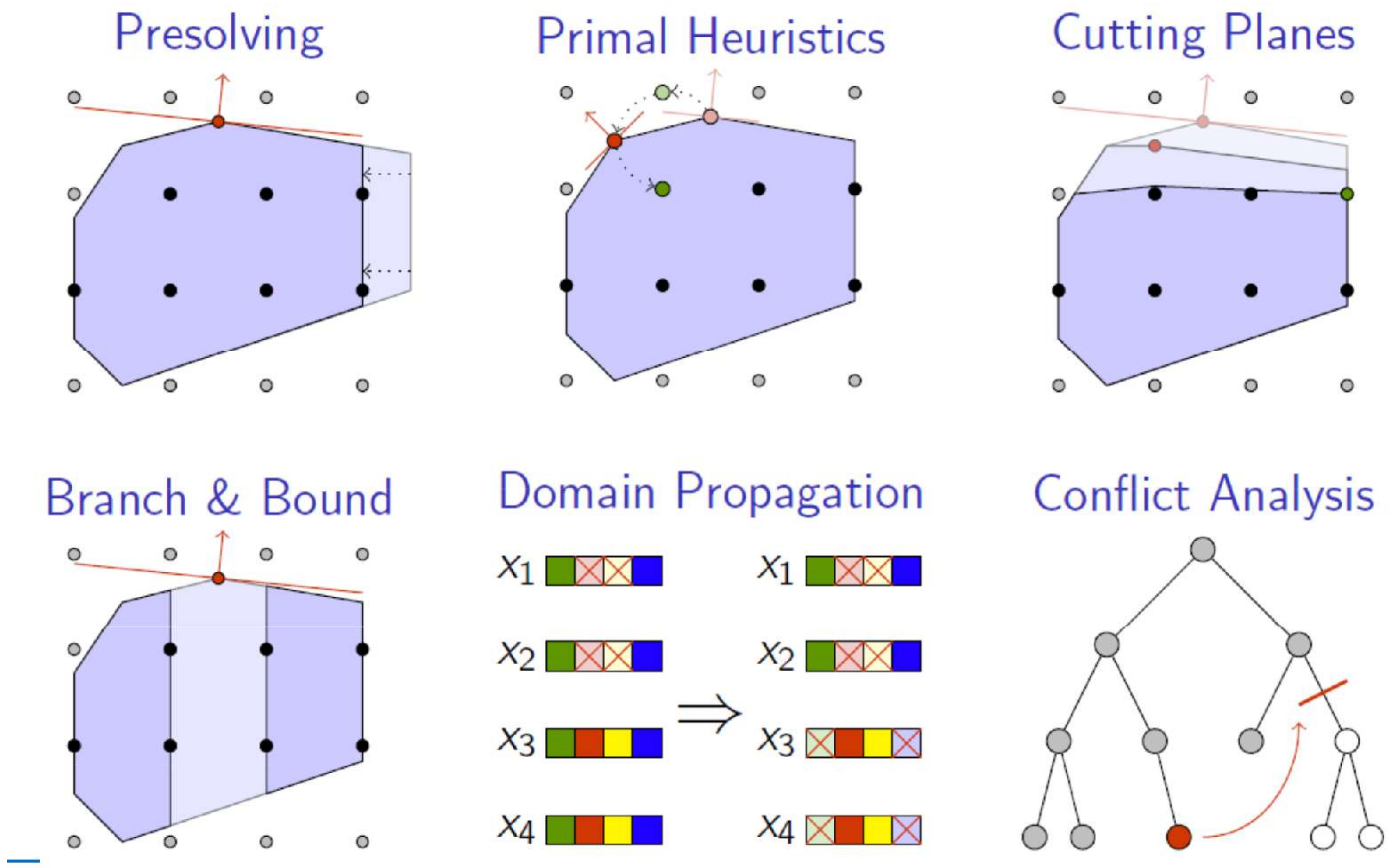
preprocessing

strategies

Branch & Cut

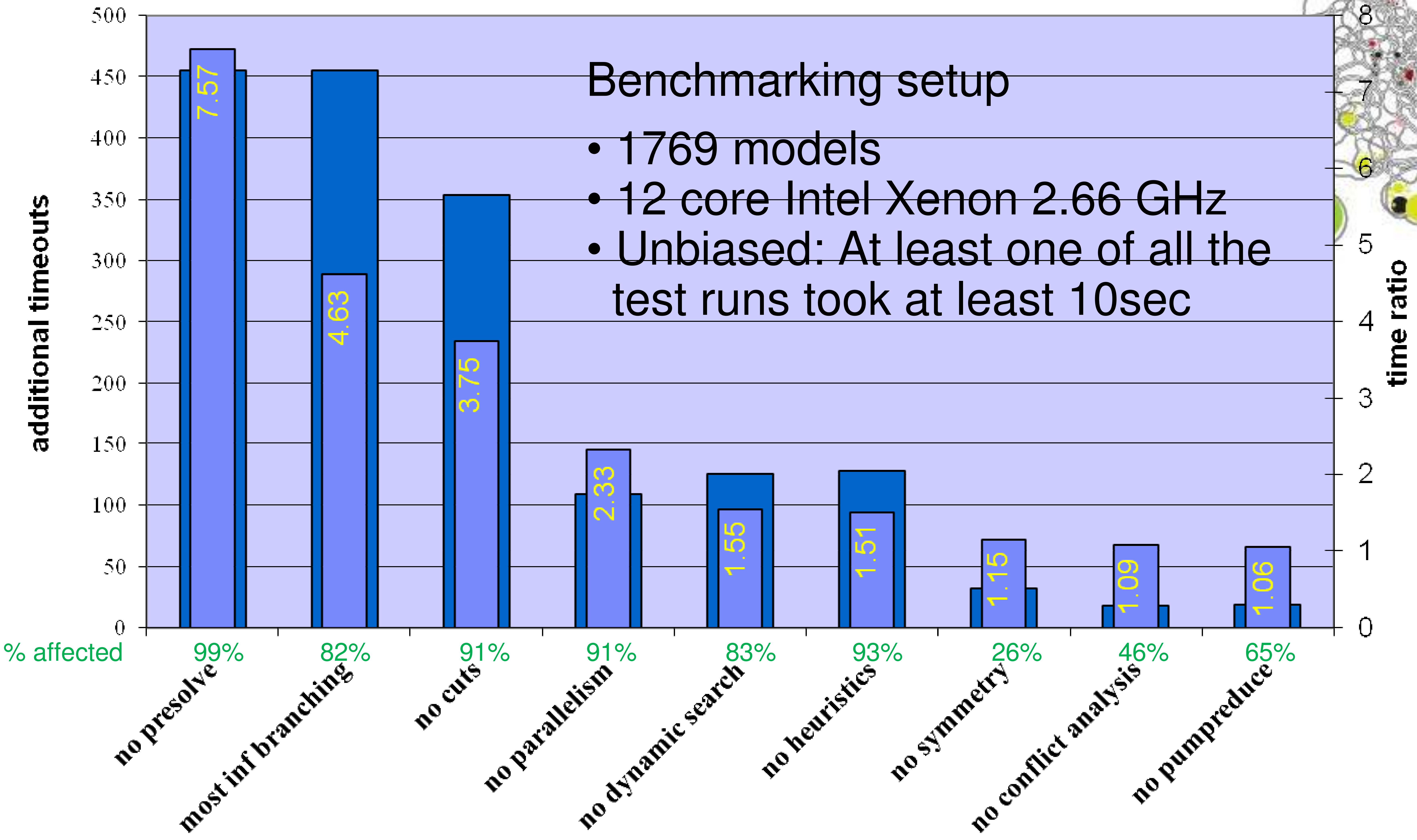
heuristics

parallelism



Slide from Martin Grötschel 2015

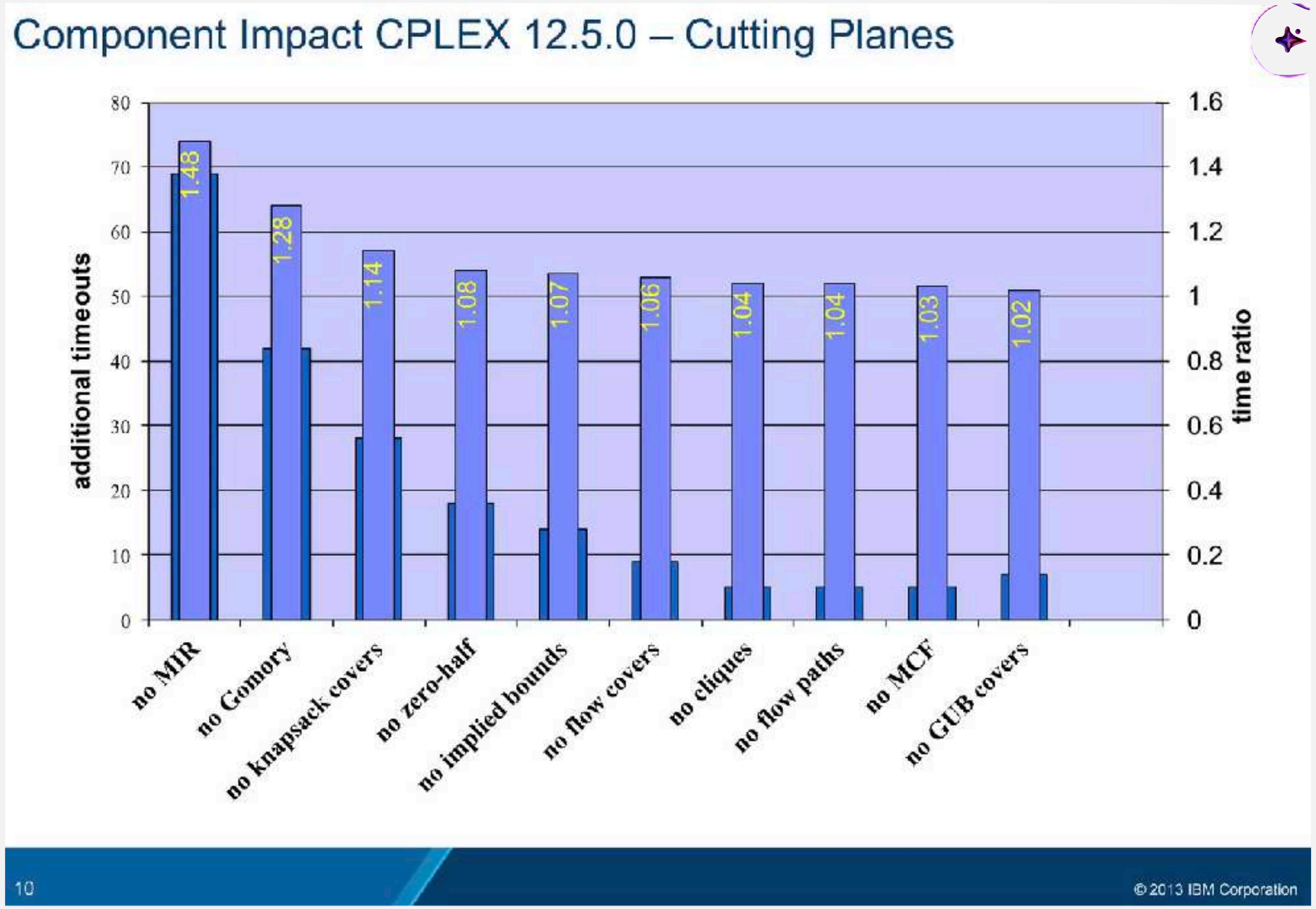
Component Impact CPLEX 12.5 Summary



Benchmarking setup

- 1769 models
- 12 core Intel Xenon 2.66 GHz
- Unbiased: At least one of all the test runs took at least 10sec

GUROBI 12



BQPCuts	BQP cut generation
Cuts	Global cut generation control
CliqueCuts	Clique cut generation
CoverCuts	Cover cut generation
CutAggPasses	Constraint aggregation passes performed during cut generation
CutPasses	Root cutting plane pass limit
DualImpliedCuts	Dual implied bound cut generation
FlowCoverCuts	Flow cover cut generation
FlowPathCuts	Flow path cut generation
GomoryPasses	Root Gomory cut pass limit
GUBCoverCuts	GUB cover cut generation
ImpliedCuts	Implied bound cut generation
InfProofCuts	Infeasibility proof cut generation
LiftProjectCuts	Lift-and-project cut generation
MIPSepCuts	MIP separation cut generation
MIRCuts	MIR cut generation
MixingCuts	Mixing cut generation
ModKCuts	Mod-k cut generation
NetworkCuts	Network cut generation
ProjImpliedCuts	Projected implied bound cut generation
PSDCuts	PSD cut generation
RelaxLiftCuts	Relax-and-lift cut generation
RLTCuts	RLT cut generation
StrongCGCuts	Strong-CG cut generation
SubMIPCuts	Sub-MIP cut generation
ZeroHalfCuts	Zero-half cut generation

Preprocessing

reduce size

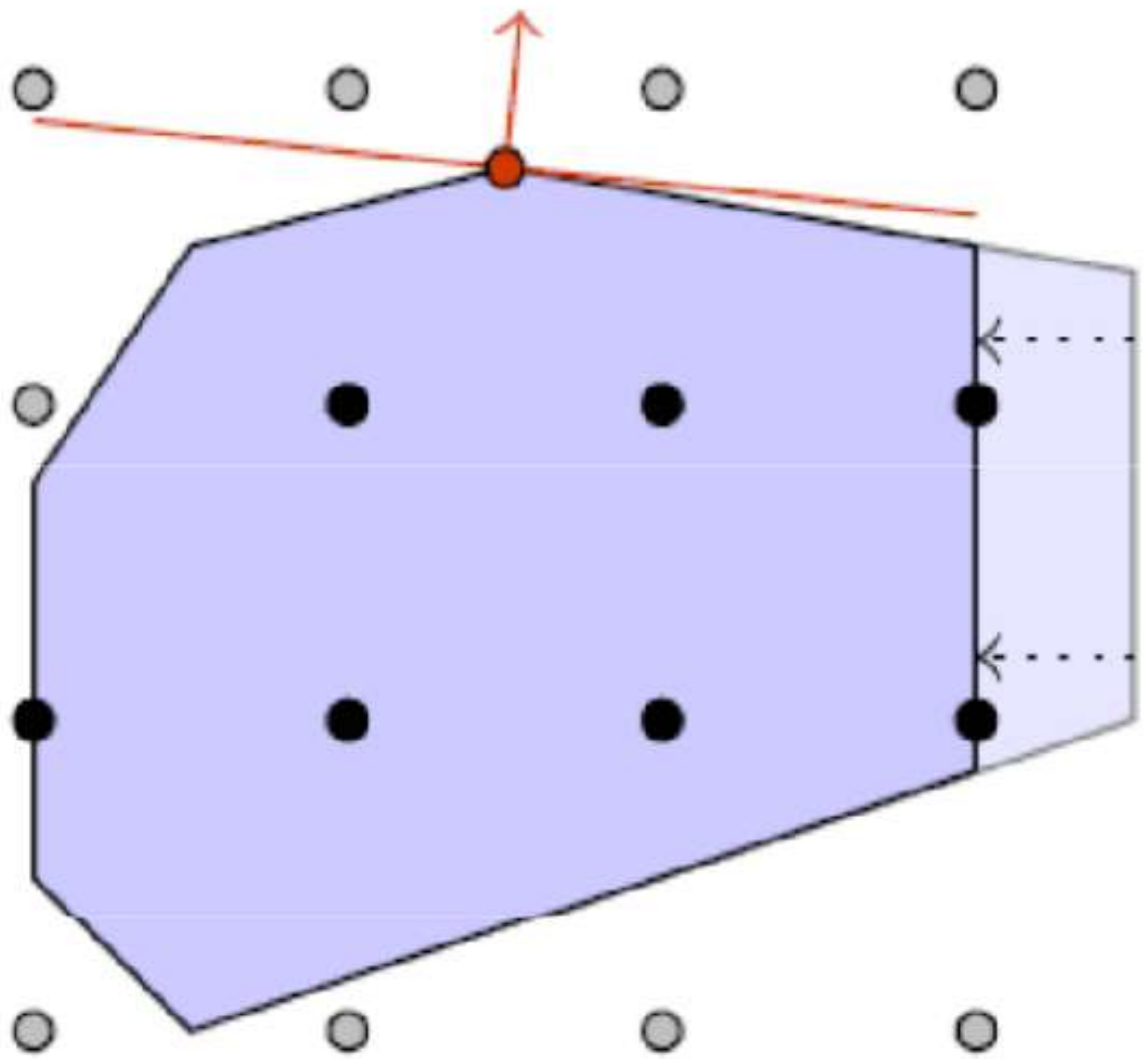
remove redundancies $x+y\leq 3$, binaries
substitute variables $x+y-z=0$
fix variables by duality $c_j\geq 0,\ A_j\geq 0 \Rightarrow x=x_{min}$
fix variables by probing $x=1$ infeas $\Rightarrow x=0$

strengthen LP relaxation

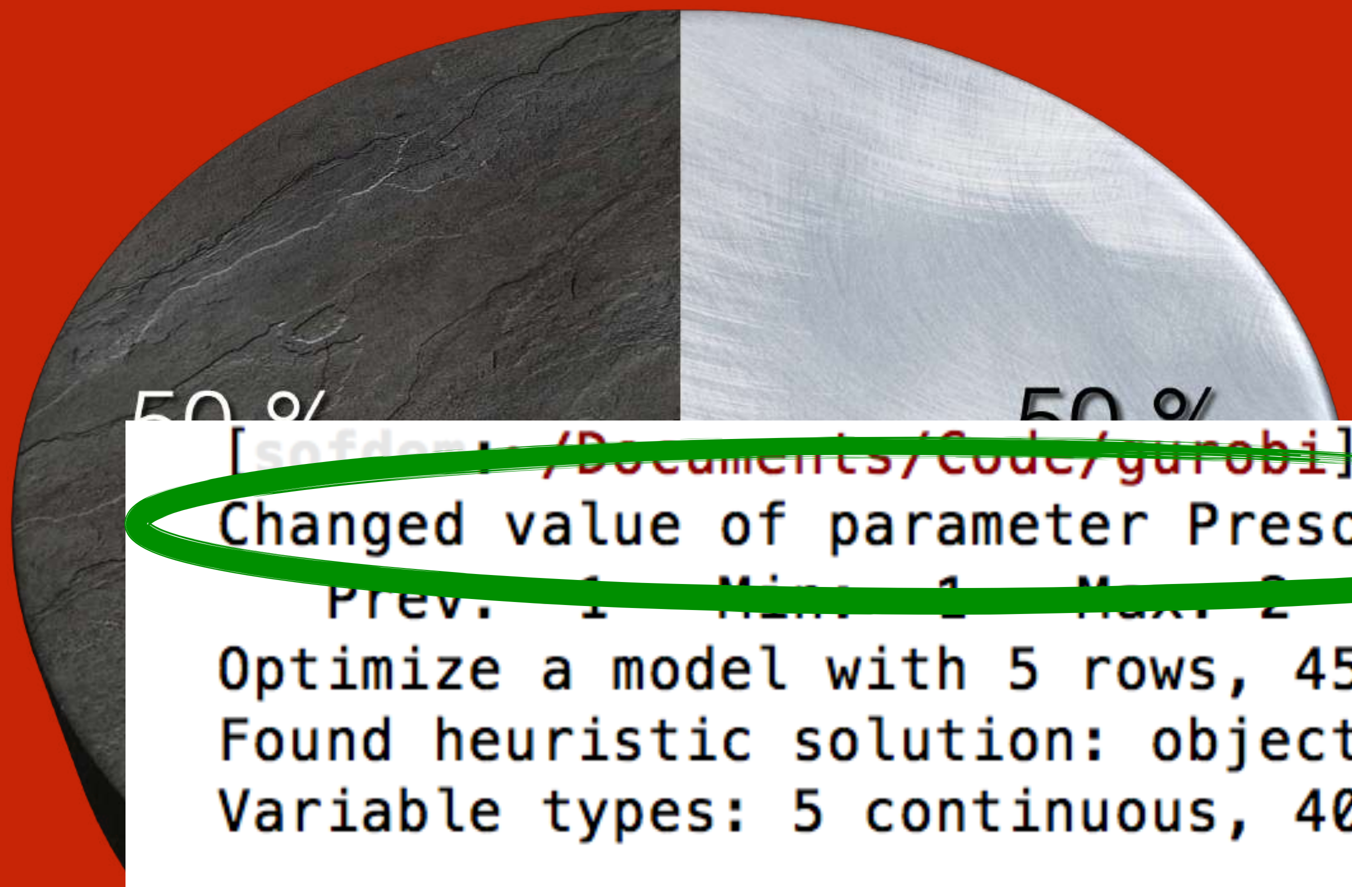
adjust bounds $2x+y\leq 1$, binaries $\Rightarrow x=0$
lift coefficients $2x-y\leq 1$, binaries $\Rightarrow x-y\leq 1$

identify/exploit properties

detect implied integer $3x+y=7$, x int $\Rightarrow y$ int
build the conflict graph
detect disconnected components
remove symmetries

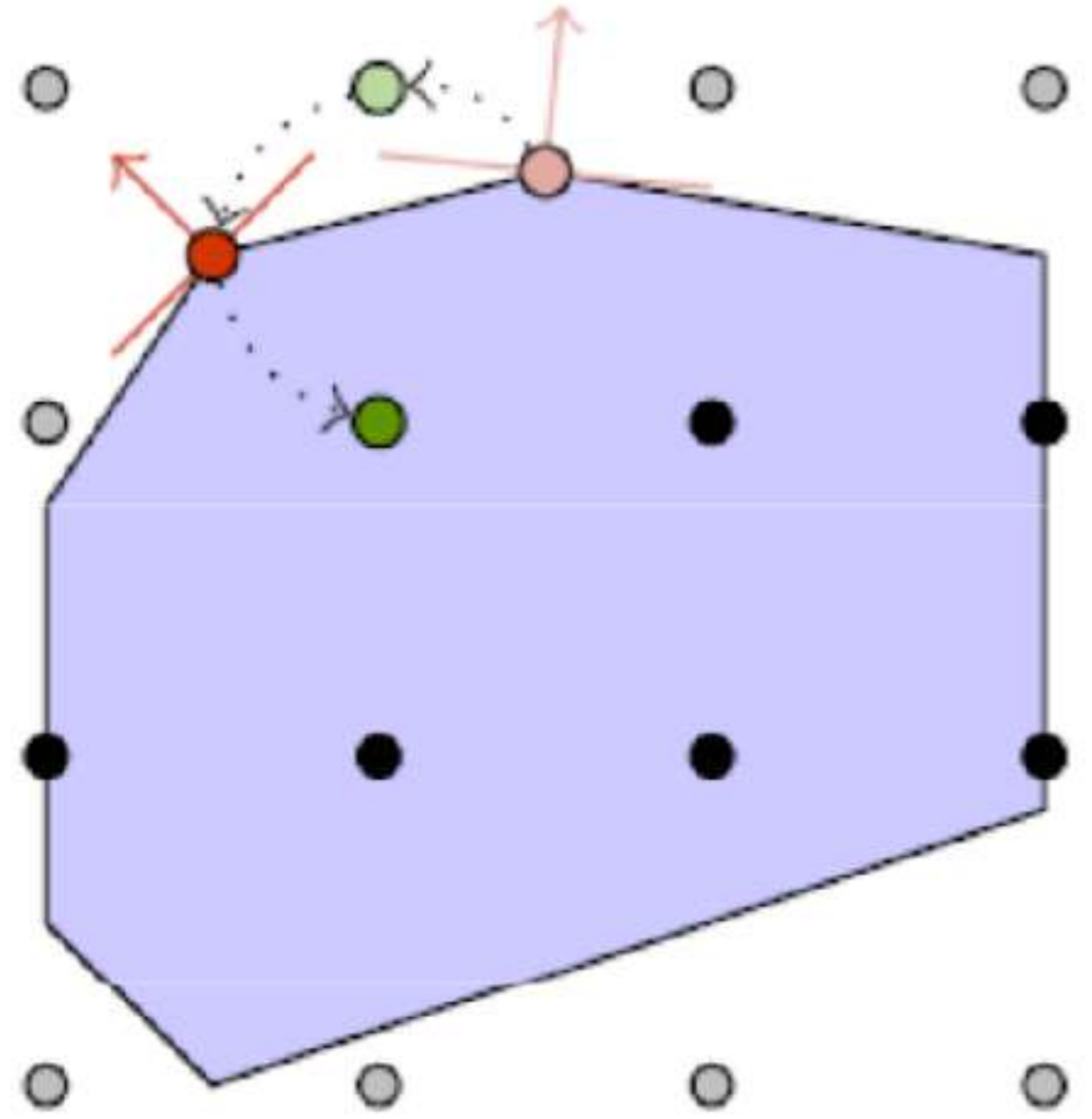


MIPLIB markshare_5_0




```
[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 3883.5 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
```

```
[sofdem:~/Documents/Code/gurobi]$ gurobi.sh mymip.py markshare_5_0.mps.gz
Optimize a model with 5 rows, 45 columns and 203 nonzeros
Found heuristic solution: objective 5335
Presolve time: 0.00s
Presolved: 5 rows, 45 columns, 203 nonzeros
Variable types: 0 continuous, 45 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
H    0     0                      320.0000000    0.00000   100%     -    0s
     0     0    0.00000    0     6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0     5  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0     6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0     5  320.00000    0.00000   100%     -    0s
H    0     0                      239.0000000    0.00000   100%     -    0s
     0     0    0.00000    0     5  239.00000    0.00000   100%     -    0s
*   36     0               29      96.0000000    0.00000   100%   2.7    0s
*   99    32               34      58.0000000    0.00000   100%   2.1    0s
H  506   214                      53.0000000    0.00000   100%   1.9    0s
H30682   442                       1.0000000    1.00000  0.00%   2.1    0s

Cutting planes:
  Cover: 26

Explored 30682 nodes (65348 simplex iterations) in 0.70 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
```



rounding LP solution
diving at some nodes

Primal Heuristics

search feasible solutions locally around the LP solution

accelerate the search a little
appeal to the practitioner a lot

limits of branch&cut

- all-purpose vs tailored solver
- highly heuristic (branching decisions, cut generation)
- floating-point errors and optimality tolerance (0.01%)
- less effective on integers vs binaries (ex: scheduling)
- MILP approximations for nonlinearities are either large or loose
- NP-hard problems

how to tune modern solvers

play with Gurobi

https://www.gurobi.com/wp-content/uploads/intro_tuning.pdf

(or read the manual)

```
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
H    0     0                      320.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
H    0     0                      239.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    5  239.00000    0.00000   100%     -    0s
*   36     0               29      96.0000000    0.00000   100%   2.7    0s
*   99    32               34      58.0000000    0.00000   100%   2.1    0s
H  506   214                       53.0000000    0.00000   100%   1.9    0s
H30682   442                        1.0000000    1.00000  0.00%   2.1    0s
```

use as a heuristic

set a time limit or loose gap

```
MIPFocus = 1
ImproveStartGap = 0.1
TimeLimit = 600
MIPGap = 1e-1
```

```
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
H    0     0                     320.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
H    0     0                     239.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    5  239.00000    0.00000   100%     -    0s
*   36     0               29      96.0000000    0.00000   100%   2.7    0s
*   99    32               34      58.0000000    0.00000   100%   2.1    0s
H  506   214                      53.0000000    0.00000   100%   1.9    0s
H30682   442                       1.0000000    1.00000  0.00%   2.1    0s
```

change the LP solver

if nbIter(node) ≥ nbIter(root)/2

NodeMethod=2

```
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
H    0     0                     320.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
H    0     0                     239.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    5  239.00000    0.00000   100%     -    0s
*   36     0              29      96.0000000    0.00000   100%   2.7    0s
*   99    32              34      58.0000000    0.00000   100%   2.1    0s
H  506   214                      53.0000000    0.00000   100%   1.9    0s
H30682   442                       1.0000000    1.00000  0.00%   2.1    0s
```

init with a feasible solution

if built-in heuristics fail

`PumpPasses,MinRelNodes,ZeroObjNodes`

`model.read("initSol.mst")`

`model.cbSetSolution(vars, newSol)`

```
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
H    0     0                       320.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    6  320.00000    0.00000   100%     -    0s
     0     0    0.00000    0    5  320.00000    0.00000   100%     -    0s
H    0     0                       239.0000000    0.00000   100%     -    0s
     0     0    0.00000    0    5  239.00000    0.00000   100%     -    0s
*   36     0                29      96.0000000    0.00000   100%   2.7    0s
*   99    32                34      58.0000000    0.00000   100%   2.1    0s
H  506   214                        53.0000000    0.00000   100%   1.9    0s
H30682   442                         1.0000000    1.00000  0.00%   2.1    0s
```

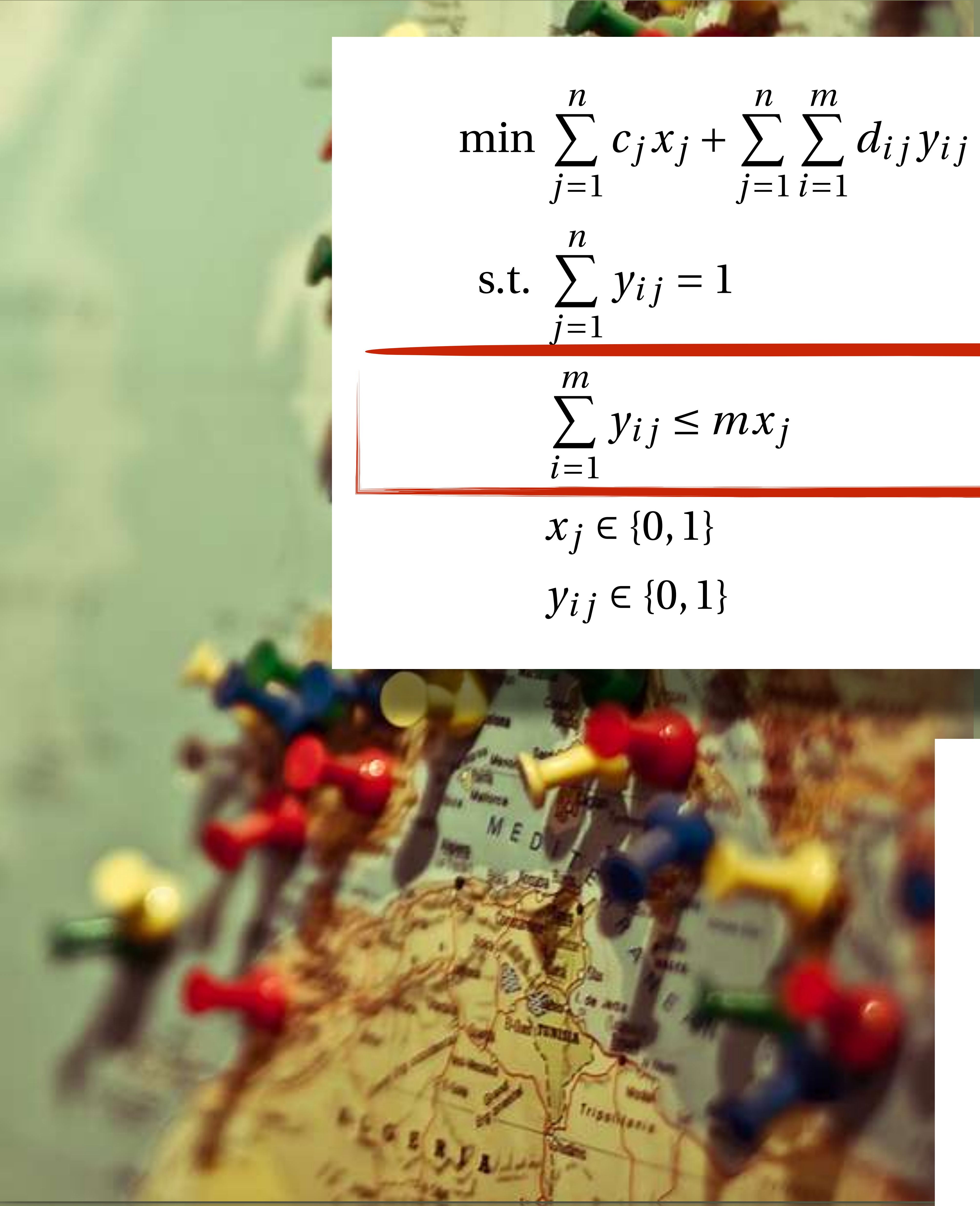
tighten the model

if the LP bound stagnates

```
Cuts = 3
Presolve = 3
model.cbCut(lhs, sense, rhs)
```

you know your problem better
than your solver does

improve the model



Uncapacitated Facility Location Problem

14 hours

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

2 seconds

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

Input n facility locations, m
pen
serve
y j
ng and
t of
.

m=n=40

Uncapacitated Lot Sizing Problem

$$\min \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t$$

$$\begin{aligned}
\text{s.t. } & s_{t-1} + x_t = d_t + s_t && t = 1..n \\
& x_t \le M y_t && t = 1..n \\
& y_t \in \{0,1\} && t = 1..n \\
& s_t, x_t \ge 0 && t = 1,\dots,n \\
& s_0 = 0 &&
\end{aligned}$$

ods, fixed
, unit production
age cost h_t,
period t
(production and
duction plan to

z_{it} production in period i to satisfy demand of period t

Uncapacitated Lot Sizing Problem

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

LP=ILP

ds, fixed
, unit production
age cost h_t,
period t
(production and
uction plan to

z_{it} production in period i to satisfy demand of period t

Bin Packing Problem

Input n containers, m items, capacity c for all containers, weight w_j for each item j
Output a packing of all items in a mimimum number of containers

Bin Packing Problem

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

tainers, m items,
for all containers,
or each item j
cking of all items
m number of

how to manage the exponential number of variables ?

Bin Packing Problem

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

Dantzig-Wolfe decomposition

tainers, m items,
for all containers,
or each item j
cking of all items
m number of

$\mathcal{S}$ all the possible arrangements of items in a bin

delayed column generation

$min\{c_B x_B + c_N x_N \,|\, A_B x_B + A_N x_N = b\}$ without (c_N, A_N) i.e. $x_N = 0$:

1/ solve the restricted LP with the primal simplex algorithm where the omitted columns N are implicitly non-basic
2/ find $j \in N$ that can profitably enter the basis $\bar{c}_j < 0$, stop if none

= dual cut generation: (cut separation = pricing problem)

$\min cx$			$\max ub$	
$A_i x \geq b_i,$	$\forall i$		$uA_j \leq c_j,$	$\forall j$
$x_j \geq 0,$	$\forall j$		$u_i \geq 0,$	$\forall i$

given a basic dual solution u find j such that $\bar{c}_j = c_j - uA_j < 0$

application to Bin Packing

$\mathcal{S} \subseteq 2^m$ all the possible arrangements of items in a bin

start with a feasible subset S covering all the items:

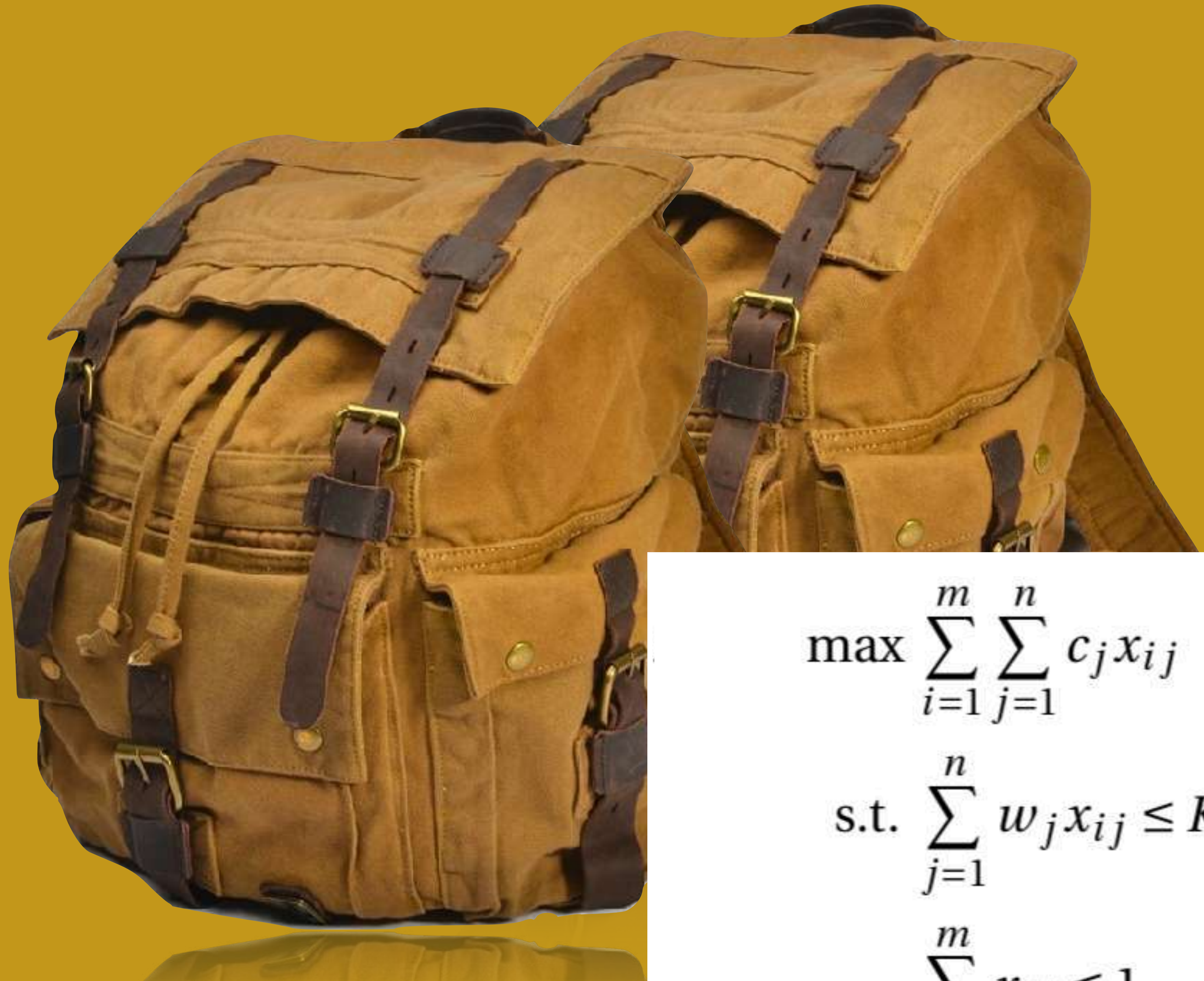
1. solve the restricted LP:
   $\min\{\sum_{s \in S} x_s \mid \sum_{s \in S} a_{js} x_s = 1\ \forall j,\ x_s \geq 0\ \forall s \in S\}$
   get the corresponding dual solution $\bar{u} \in \mathbb{R}^m$
2. look for an improving basic direction
   = some $s \in \mathcal{S} \setminus S$ with $\bar{c}_s = 1 - \sum_j a_{js} \bar{u}_j < 0$
   e.g. by solving $\max\{\sum_j a_j \bar{u}_j \mid \sum_j w_j a_j \leq K, a \in \{0,1\}^m\}$
3. if $\sum_j a_j^* \bar{u}_j > 1$ add column $(1, a^*)$ to S then 1
   otherwise STOP: $(\bar{x}_S, 0)$ solves the LP-relaxation

Branch-and-Price

- branch-and-bound for ILP with large number of variables where the LP relaxation is solved by **column generation**
- the branching strategy should keep the **search tree balanced** without altering the LP relaxation structure, ex (bin packing): branch by fixing to 0 either all $x_s \mid \{i,j\} \subseteq s$ or all $x_s \mid \{i,j\} \nsubseteq s$ for some pair of items (i,j) s.t. $0 < \sum_s a_{is}a_{js}x_s < 1$
- the pricing problem can be seen as an optimization problem but does not need to be **solved at optimality**, except for the convergence proof.
- convenient decomposition method when **additional constraints** only appear in the pricing problem, ex (conflicts in bin packing): $\sum_{j \in C} a_j \leq 1$

Multi 0-1 Knapsack Problem

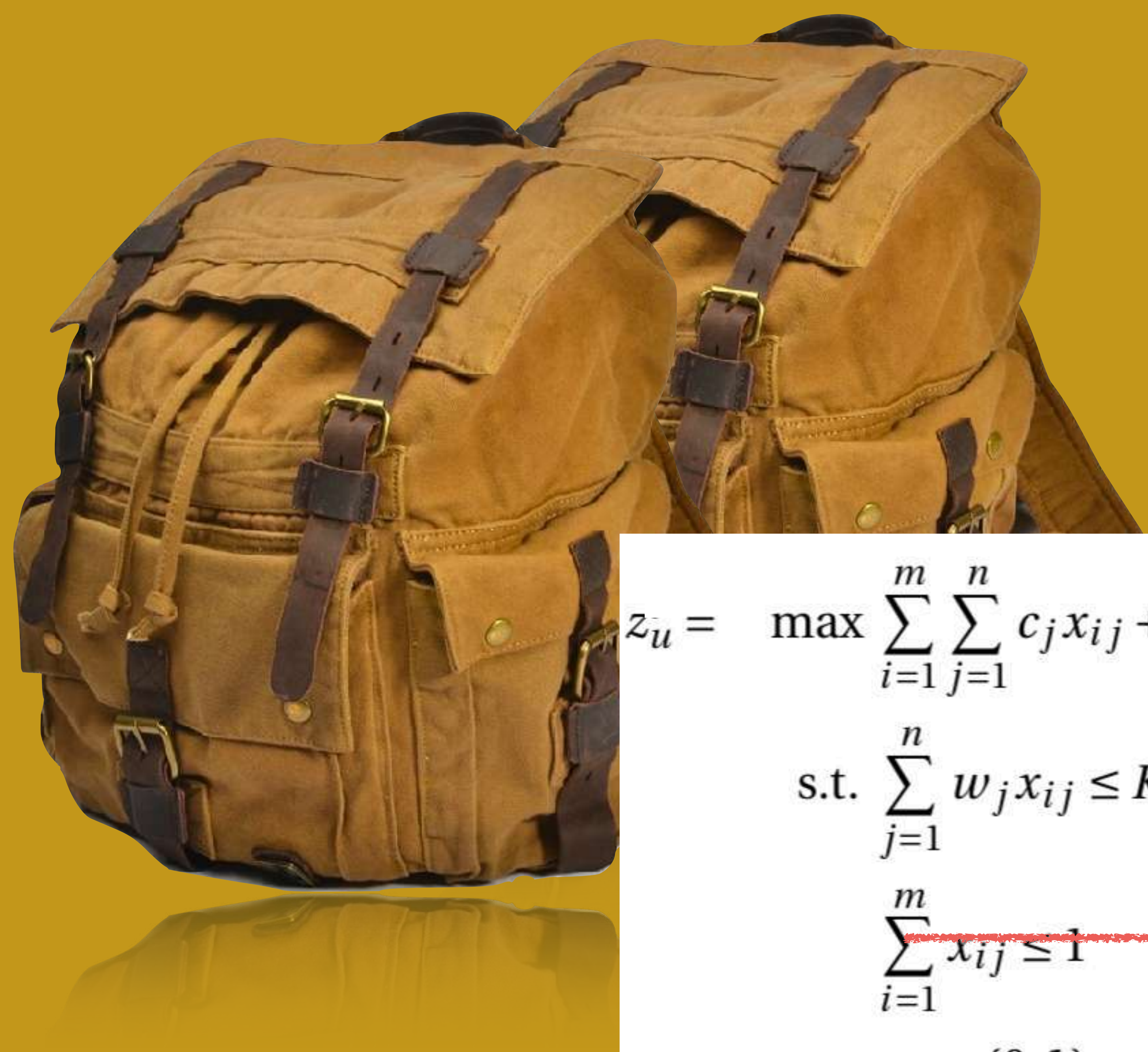
Input n items, m bins, value c_j and weight w_j for each item j, capacity K_i for each bin i. Output a maximum value subset of items packed in the bins.



Multi 0-1 Knapsack Problem

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

m bins, value
for each item
or each bin i.
m value subset
in the bins.



Multi 0-1 Knapsack Problem

$$u \in \mathbb{R}^n_+$$

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

m bins, value
for each item
or each bin i.
m value subset
in the bins.

find the smallest upper bound

lagrangian relaxation

Lagrangian Relaxation

dualize the complicating or coupling constraints of an ILP:

$$
\begin{aligned}
(P) : z = \max \sum_k c_k x_k \\
\sum_k D_k x_k \leq e_k \\
A_k x_k \leq b_k, \qquad & \forall k \\
x_k \in \mathbb{Z}^p \times \mathbb{R}^n, \qquad & \forall k
\end{aligned}
$$

$$
\begin{aligned}
(D) : w = \min_{u \geq 0} l(u) \\
l(u) = ue + \sum_k z_k^u \\
(P_u) : z_u^k = \max c_k x_k - u D_k x_k \\
A_k x_k \leq b_k \\
x_k \in \mathbb{Z}^p \times \mathbb{R}^n
\end{aligned}
$$

(D) is the lagrangian dual problem

(P_u) is the lagrangian suproblem with multipliers u

strong duality may not hold if p>0, ie the dual only provides an upper bound $w \geq z$

solving the lagrangian dual

$$
\begin{aligned}
(P) : z = \max \sum_k c_k x_k & \\
\sum_k D_k x_k \leq e_k & \\
A_k x_k \leq b_k, & \qquad \forall k \\
x_k \in \mathbb{Z}^p \times \mathbb{R}^n, & \qquad \forall k
\end{aligned}
$$

$$
\begin{aligned}
(D) : w &= \min_{u \geq 0} l(u) \\
l(u) &= ue + \sum_k z_k^u \\
(P_k^u) : z_k^u &= \max c_k x_k - u D_k x_k \\
& A_k x_k \leq b_k \\
& x_k \in \mathbb{Z}^p \times \mathbb{R}^n
\end{aligned}
$$

- function l is convex and a subgradient at $u \geq 0$ is $e - \sum D_k x_k^u$ where x_k^u an optimal solution of (P_k^u)
- minimize l with a subgradient, bundle, or cutting-plane method
- almost feasible solutions computed at each iteration: repair violations heuristically to get feasible solutions and lower bounds

MILP perks

performance
sophisticated algorithms

declarative
models, not algorithms

large-scale
decomposition methods

certification
primal-dual bounds

versatile
covers many problems

flexible
general-purpose solvers

logic & constraint programming

integer nonlinear programming

graph algorithms

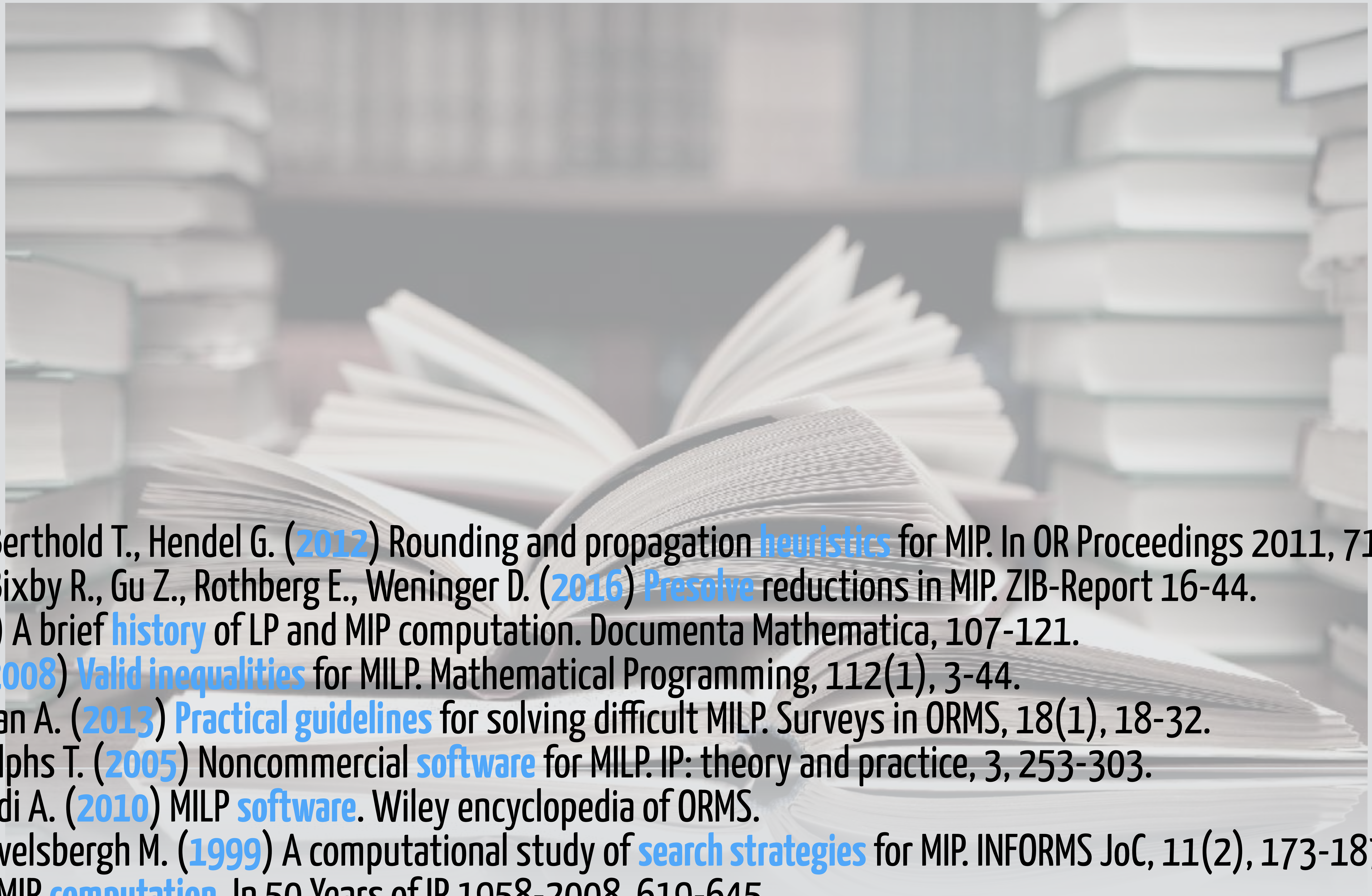
combinatorial optimization beyond MILP

machine learning

dynamic programming

metaheuristics

Matteo Fischetti (2019) Introduction to Mathematical Optimization
Wolsey L. (1998) Integer Programming. Wiley
Bertsimas D., Tsitsiklis J. (1997) Introduction to Linear Optimization. Athena Scientific



Achterberg T., Berthold T., Hendel G. (2012) Rounding and propagation heuristics for MIP. In OR Proceedings 2011, 71-76.
Achterberg T., Bixby R., Gu Z., Rothberg E., Weninger D. (2016) Presolve reductions in MIP. ZIB-Report 16-44.
Bixby R. (2012) A brief history of LP and MIP computation. Documenta Mathematica, 107-121.
Cornuéjols G. (2008) Valid inequalities for MILP. Mathematical Programming, 112(1), 3-44.
Klotz E., Newman A. (2013) Practical guidelines for solving difficult MILP. Surveys in ORMS, 18(1), 18-32.
Linderoth J., Ralphs T. (2005) Noncommercial software for MILP. IP: theory and practice, 3, 253-303.
Linderoth J., Lodi A. (2010) MILP software. Wiley encyclopedia of ORMS.
Linderoth J., Savelsbergh M. (1999) A computational study of search strategies for MIP. INFORMS JoC, 11(2), 173-187.
Lodi A. (2010) MIP computation. In 50 Years of IP 1958-2008, 619-645.
Newman A., Weiss M. (2013) A survey of linear and mixed-integer optimization tutorials. INFORMS ToE, 14(1) 26-38.
Savelsbergh M. (1994) Preprocessing and probing techniques for MIP. ORSA Journal on Computing, 6(4), 445-454.
Vanderbeck F., Wolsey L. (2010) Reformulation and decomposition of IP. In 50 Years of IP 1958-2008, 431-502.