15-780: Grad AI
Lecture 11: Optimization, Duality

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Sam Ganzfried, Byron Boots

Admin

Alexandra Marilyn Gordon

Project proposals

- *Due today*

HW3

- *Questions?*

LPs, MILPs, and their ilk

Recall

- *Linear program:*

$$\min\ 3x + 2y \ \ s.t.$$

$$x + 2y \leq 3$$

$$x \leq 2$$

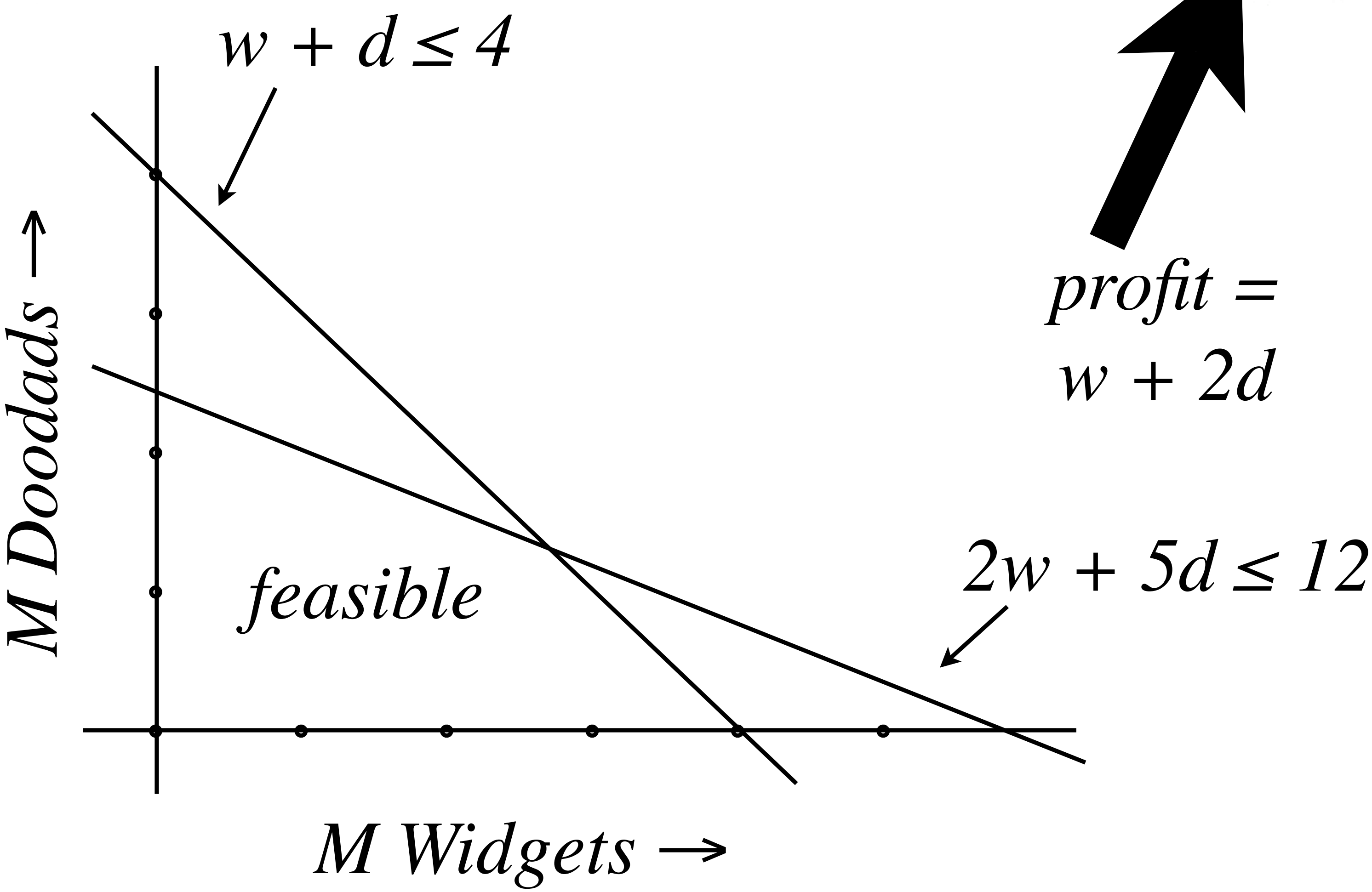
$$x, y \geq 0$$

- *Integer linear program: add* $x, y \in \mathbb{Z}$
- *Mixed ILP:* $x \in \mathbb{Z}, y \in \mathbb{R}$

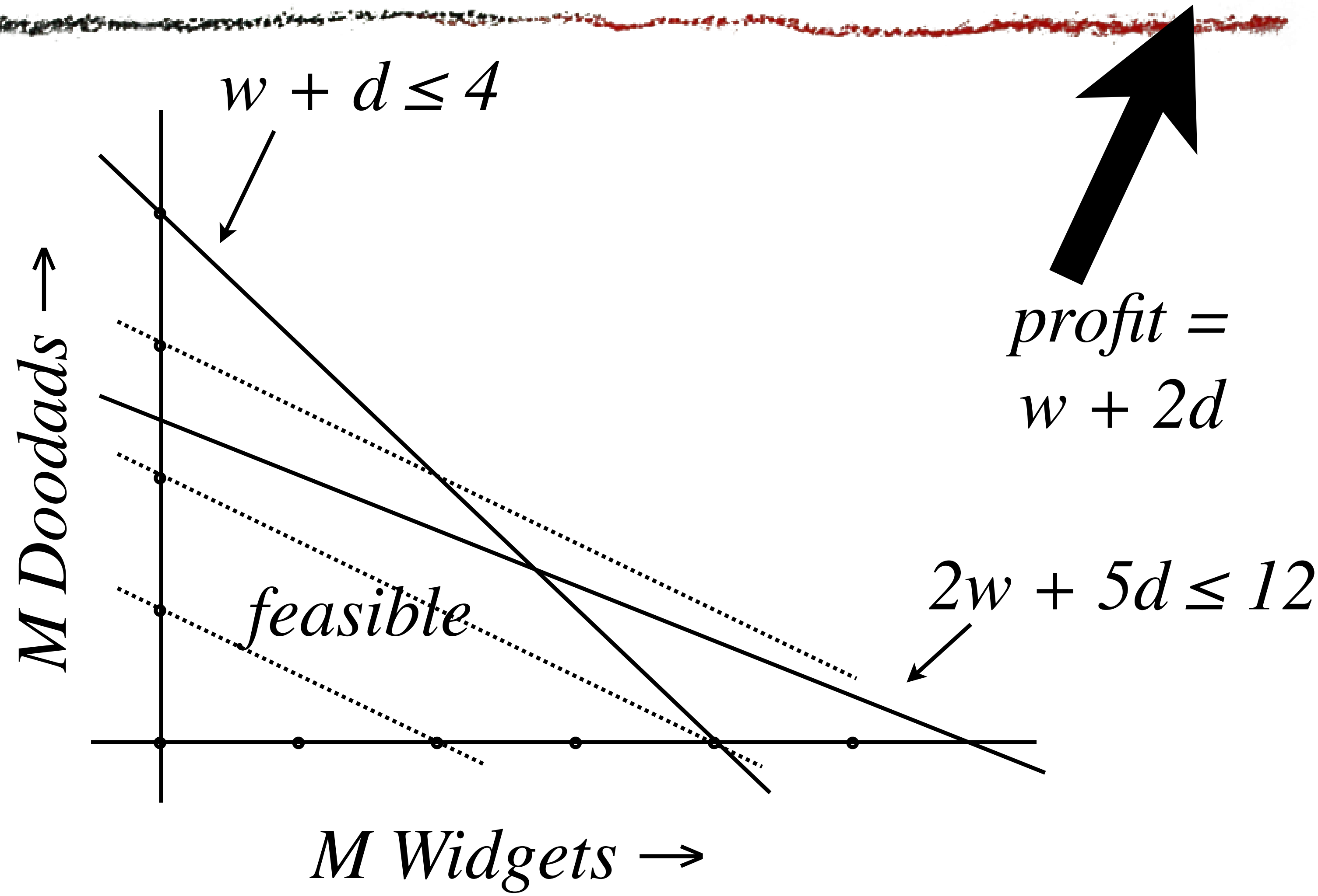
Example LP

- *Factory makes widgets and doodads*
- *Each widget takes 1 unit of wood and 2 units of steel to make*
- *Each doodad uses 1 unit wood, 5 of steel*
- *Have 4M units wood and 12M units steel*
- *Maximize profit: each widget nets \$1, each doodad nets \$2*

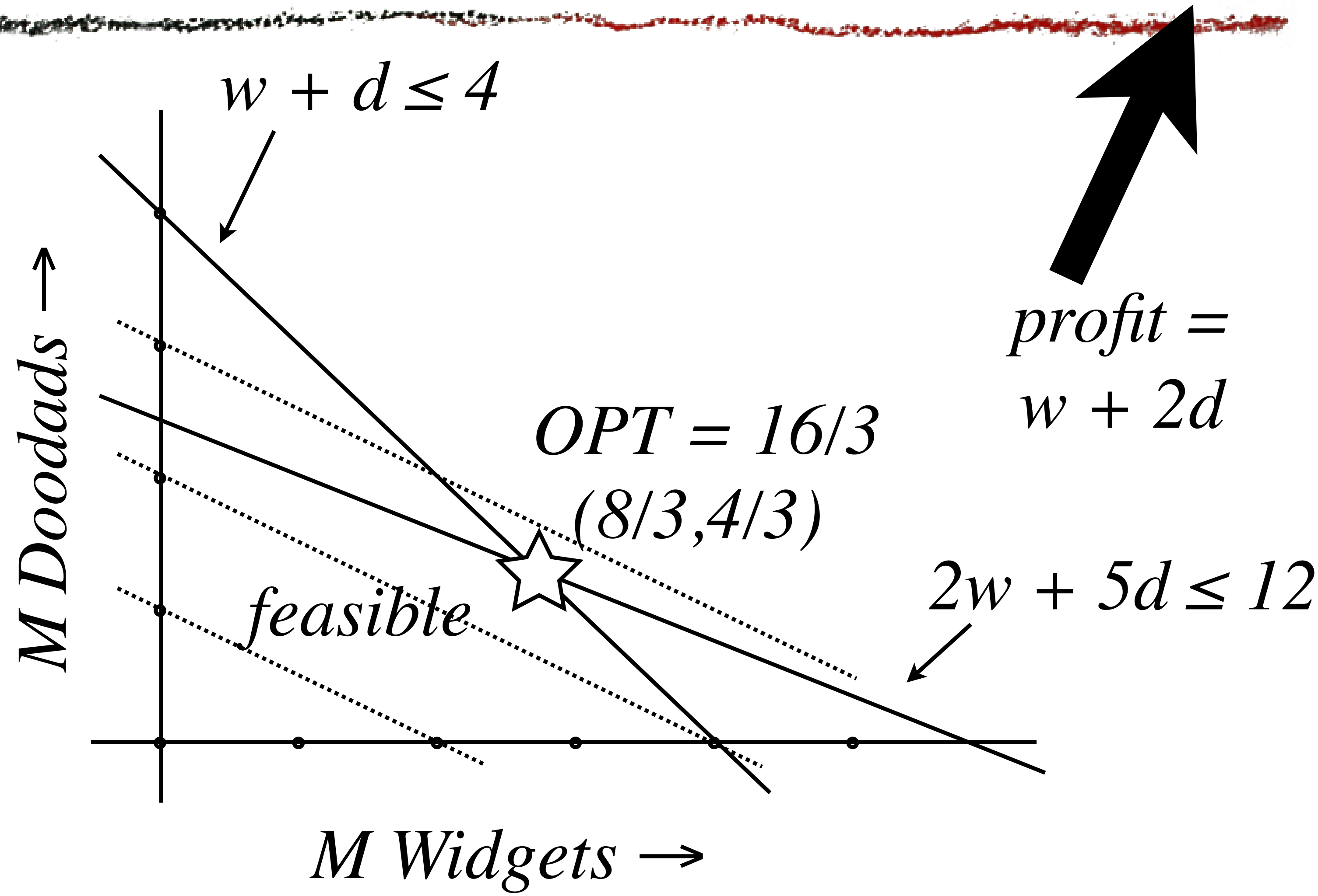
Factory LP



Factory LP



Factory LP

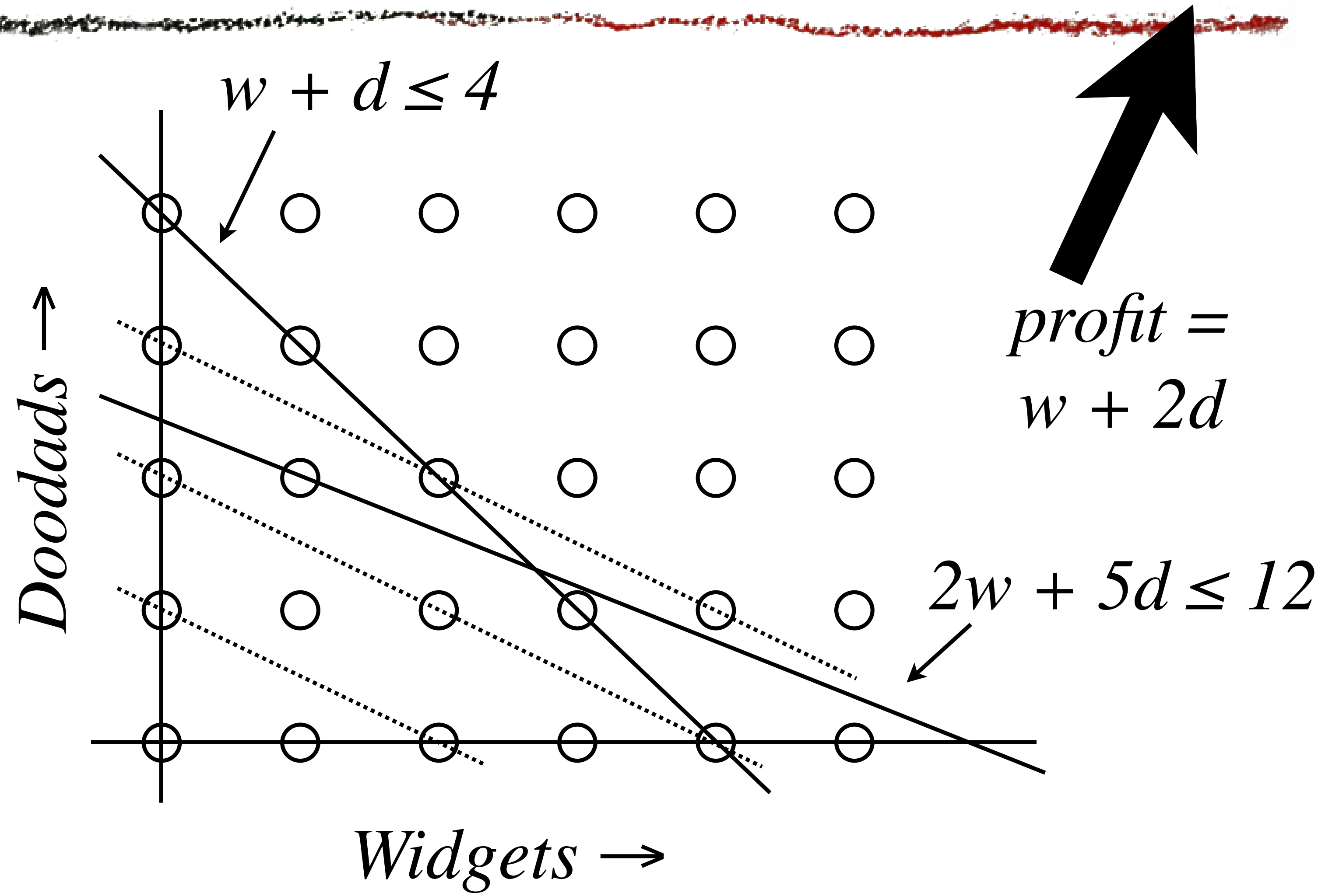


w + d ≤ 4
profit =
w + 2d
OPT = 16/3
(8/3,4/3)
2w + 5d ≤ 12
feasible
M Doodads →
M Widgets →

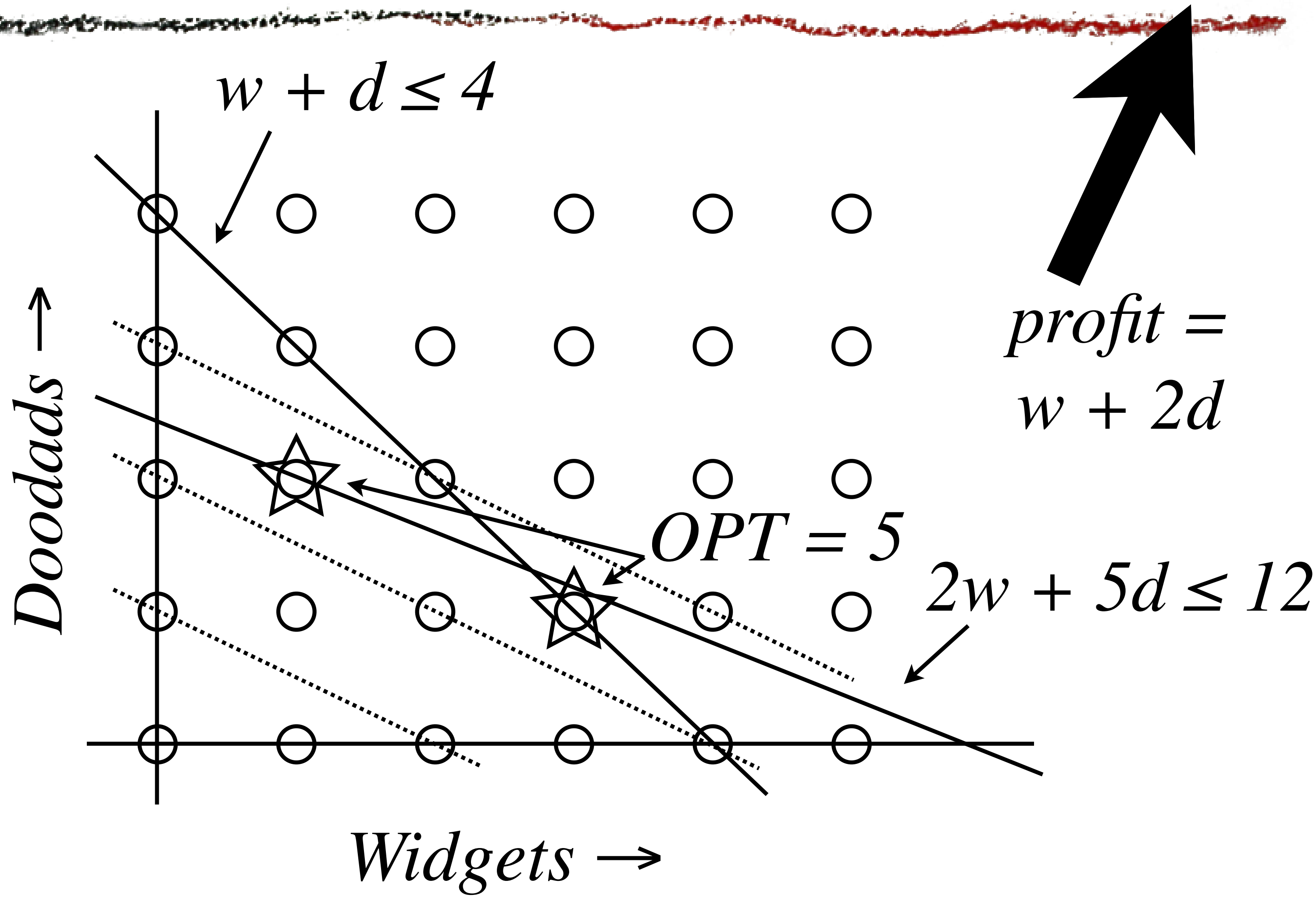
Example ILP

- *Instead of 4M units of wood, 12M units of steel, have 4 units wood and 12 units steel*

Factory example



Factory example

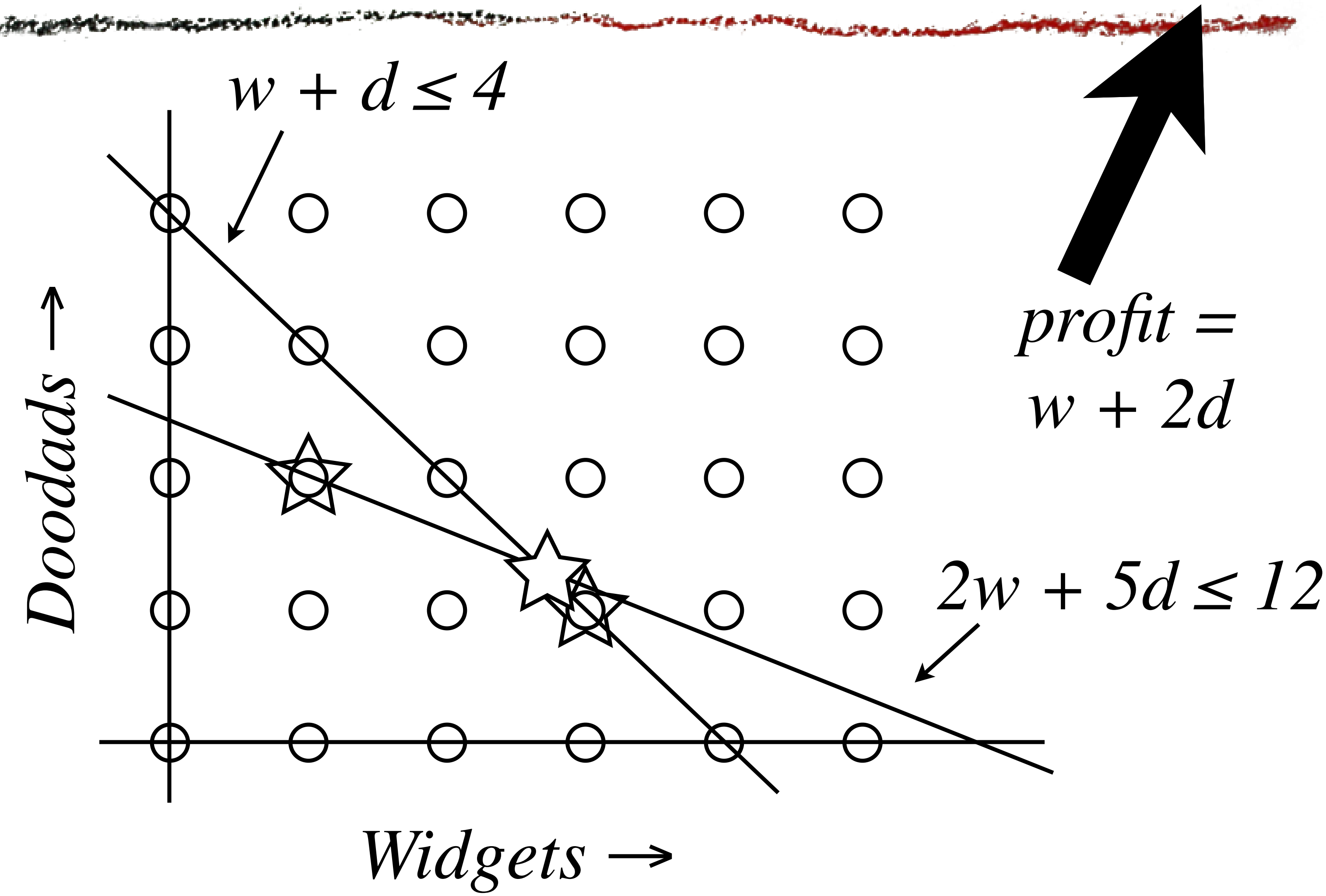


LP relaxations

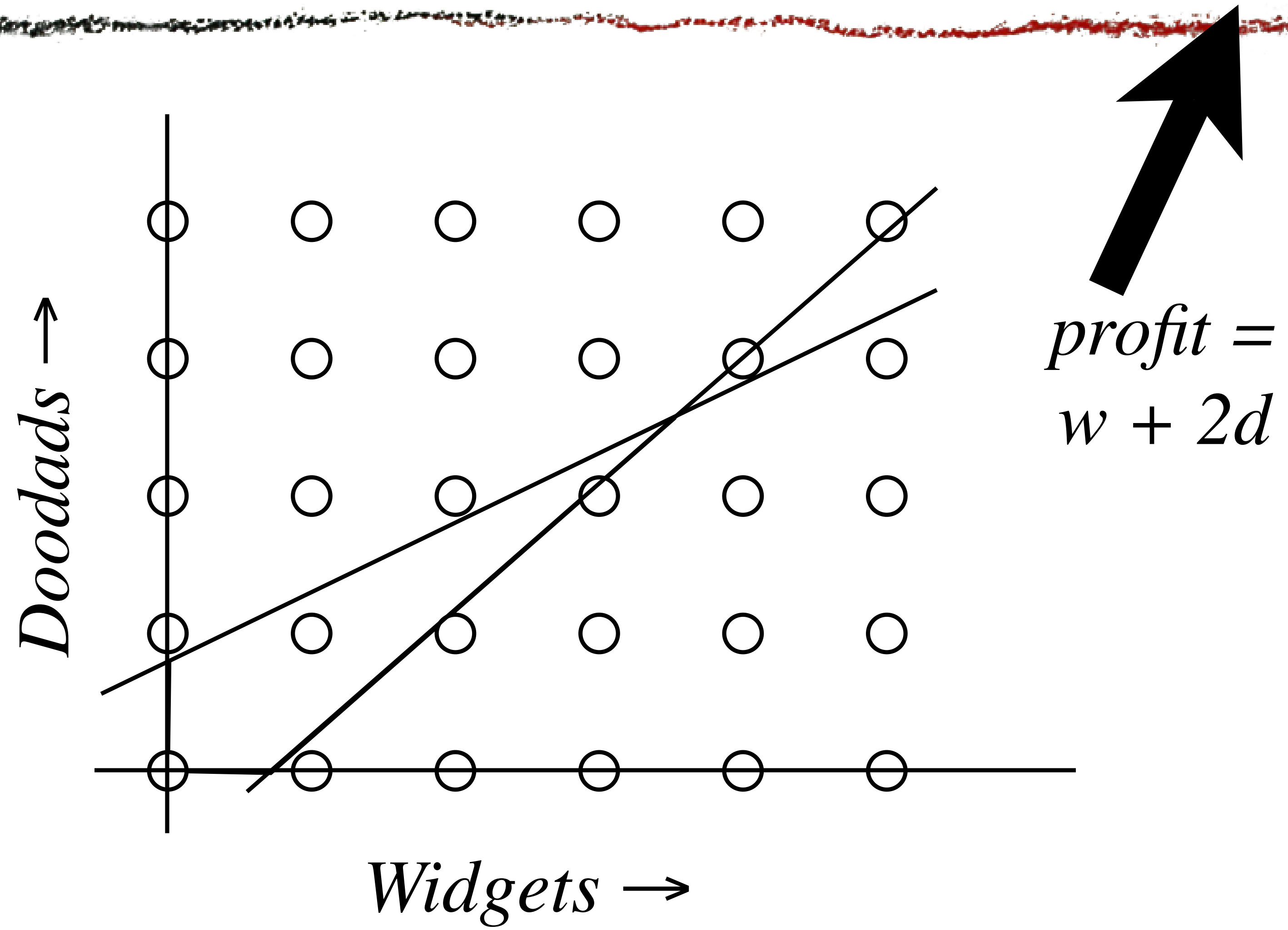
- *Above LP and ILP are the same, except for constraint $w, d \in \mathbb{Z}$ (in ILP)*
- *LP is a **relaxation** of ILP*
- *Adding any constraint makes optimal value **same or worse***
- *So, $OPT(LP) \geq OPT(ILP)$*

  (in a maximization problem)

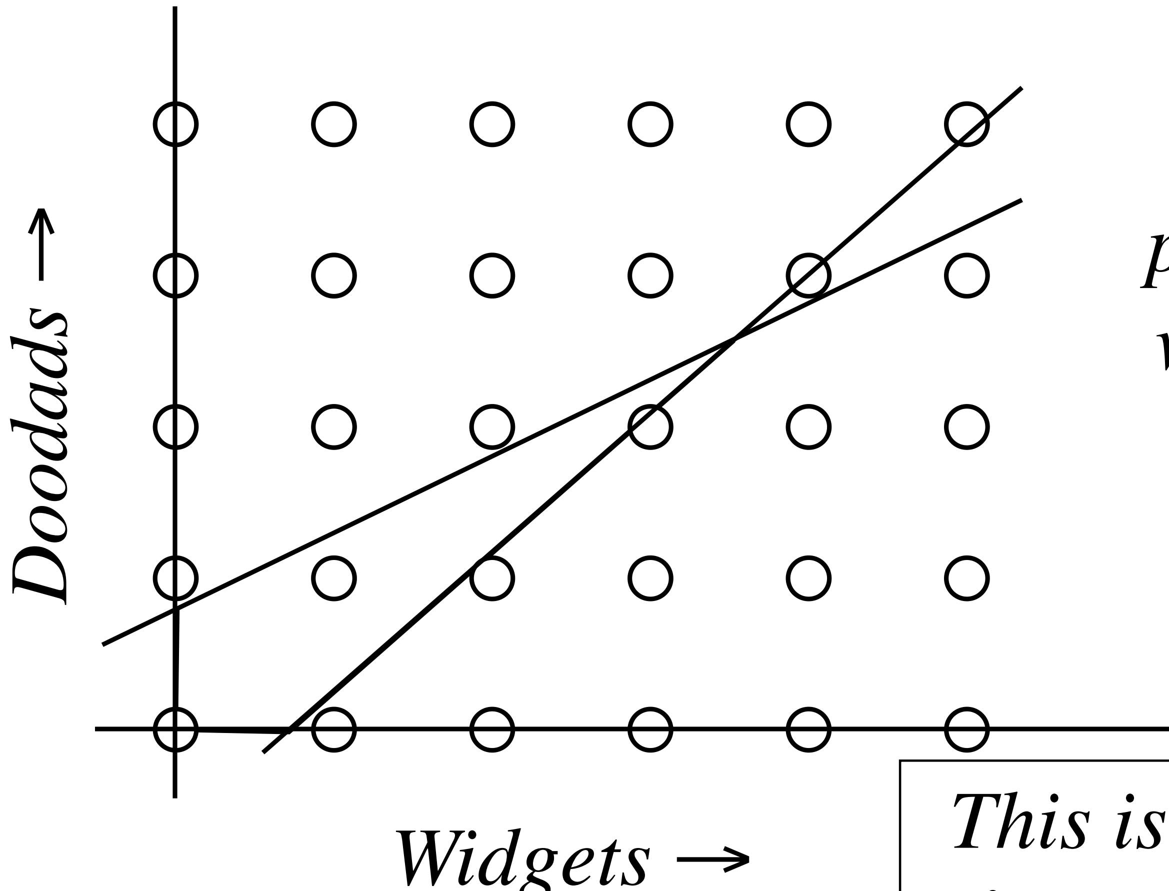
Factory relaxation is pretty close



Unfortunately…



Unfortunately…



profit =
w + 2d

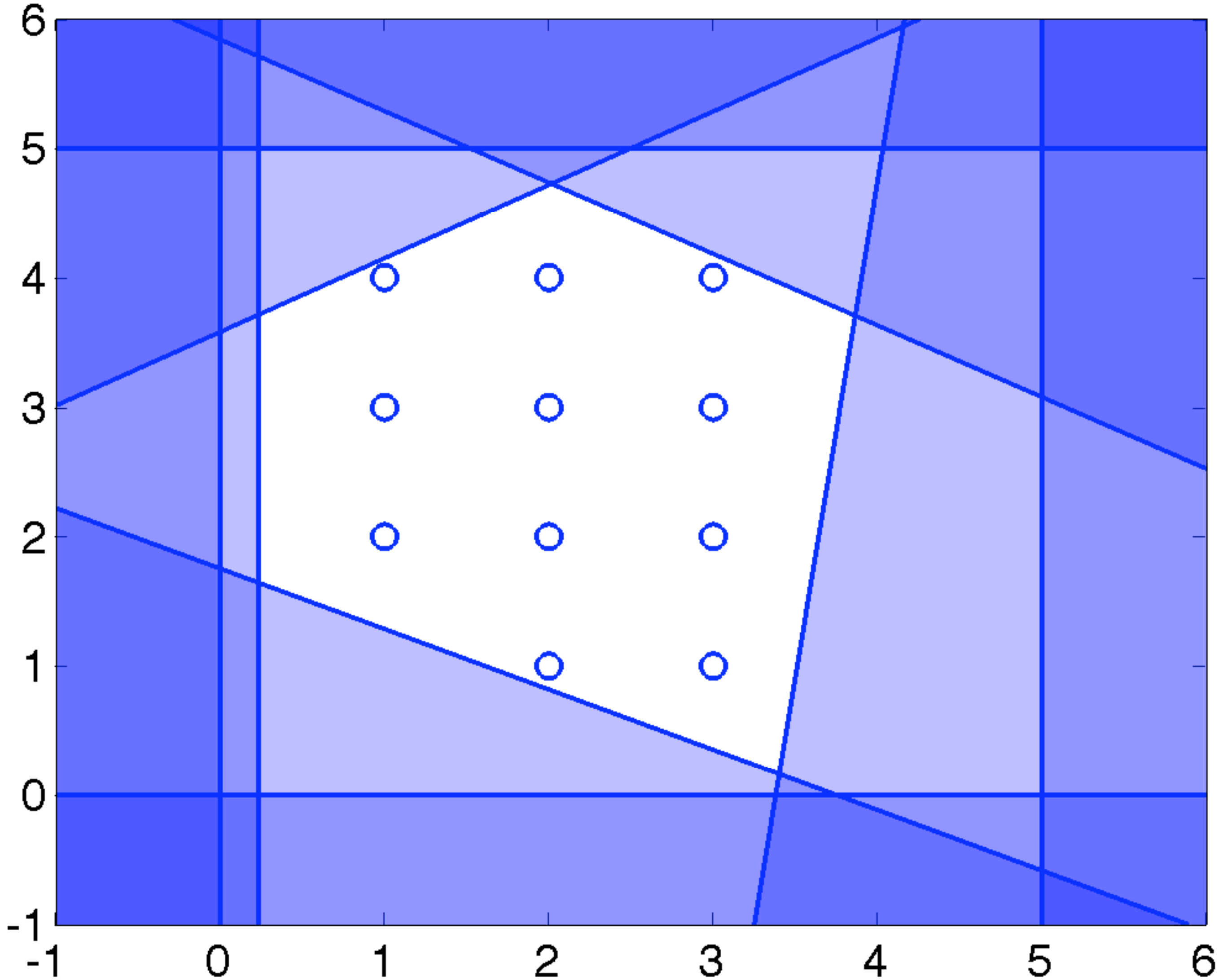
This is called an ***integrality gap***

15

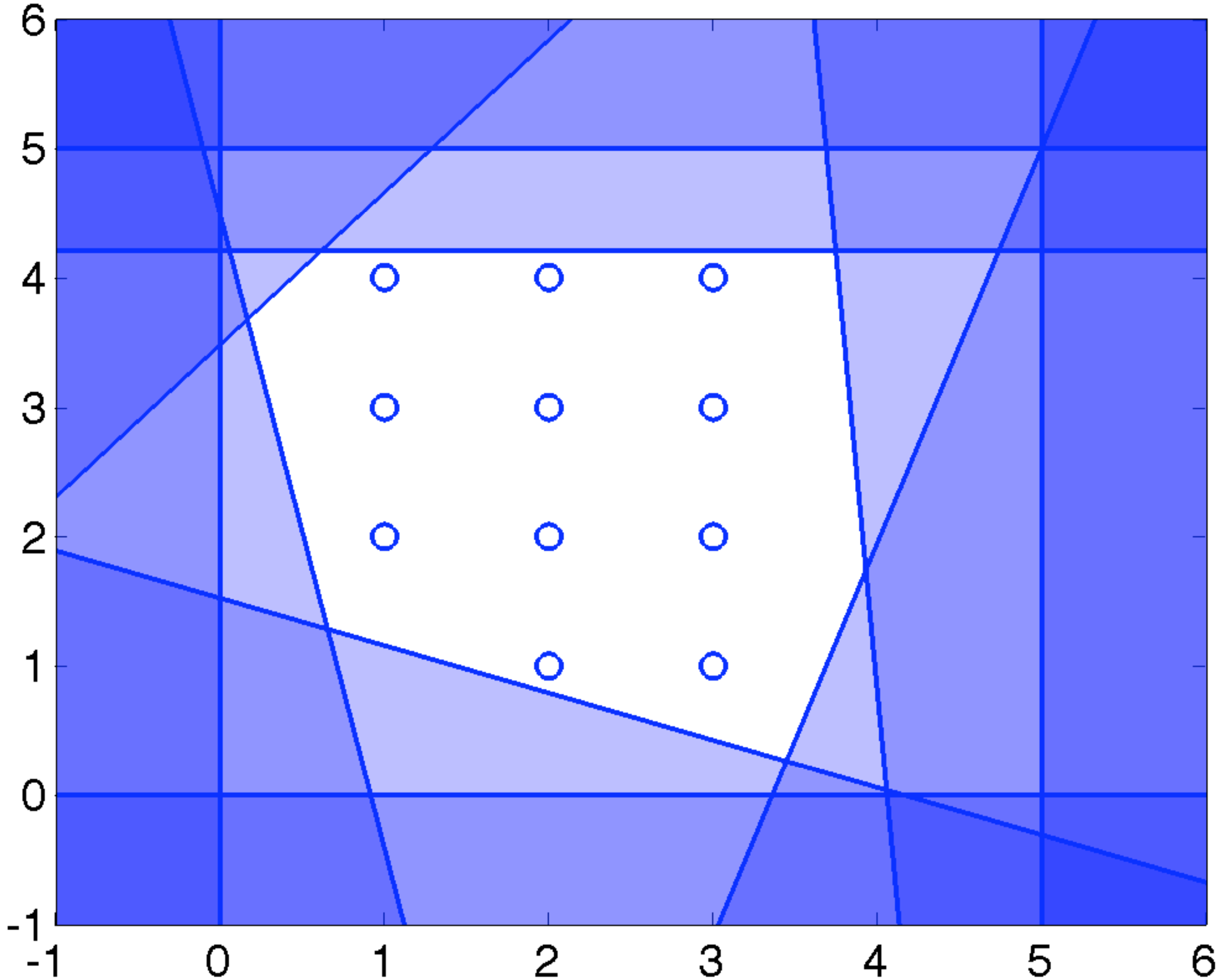
Bad gap

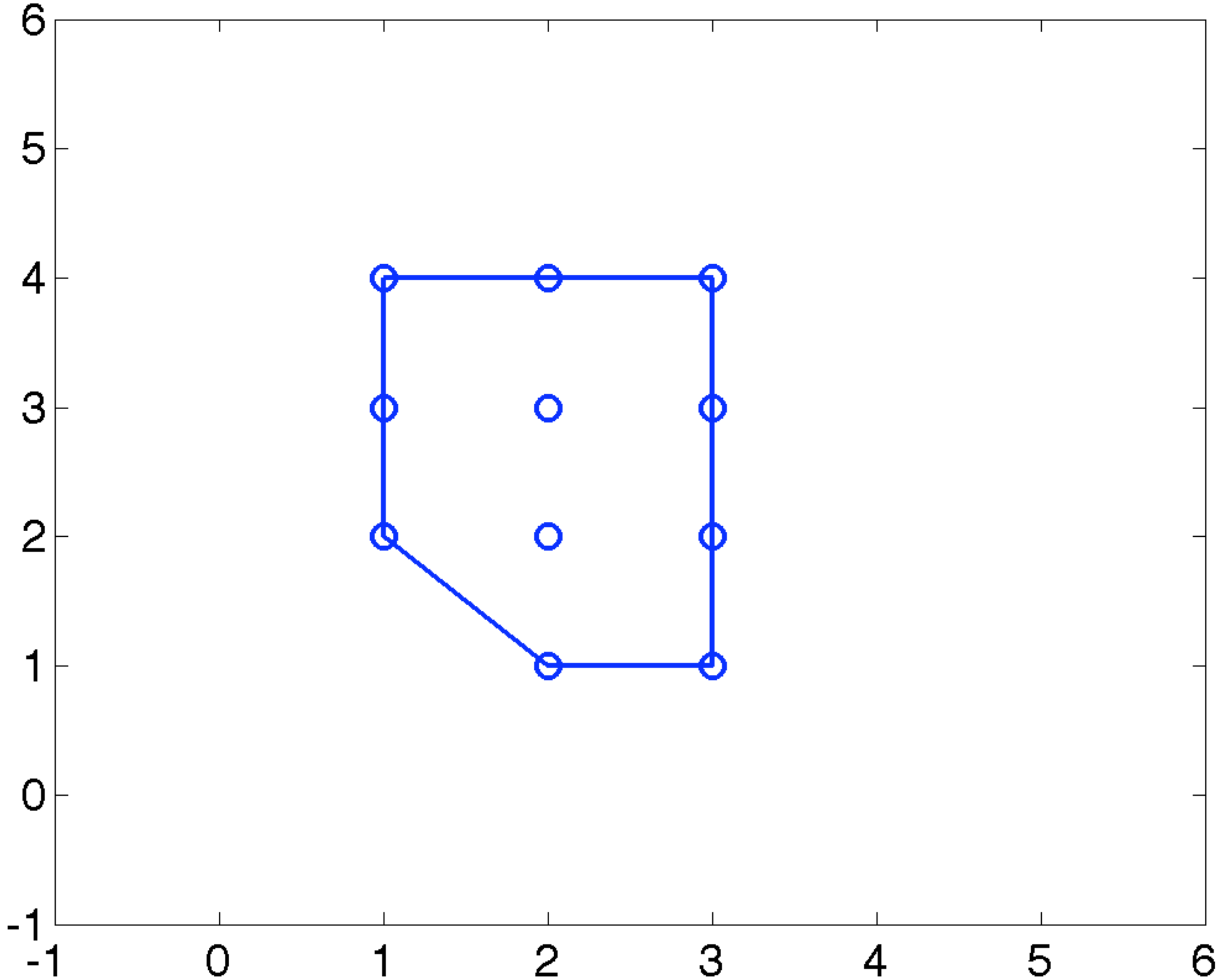
- *In this example, gap is 3 vs 8.5, or about a ratio of 0.35*
- *Ratio can be arbitrarily bad*
  - *but, can sometimes bound it for classes of ILPs*

16



6
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
6


6
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
6



6
5
4
3
2
1
0
-1
-1
0
1
2
3
4
5
6

3D LP example

$$max\ 3z + x - 2y\ s.t.$$

$$|x| + |y| + |z| \le 1$$

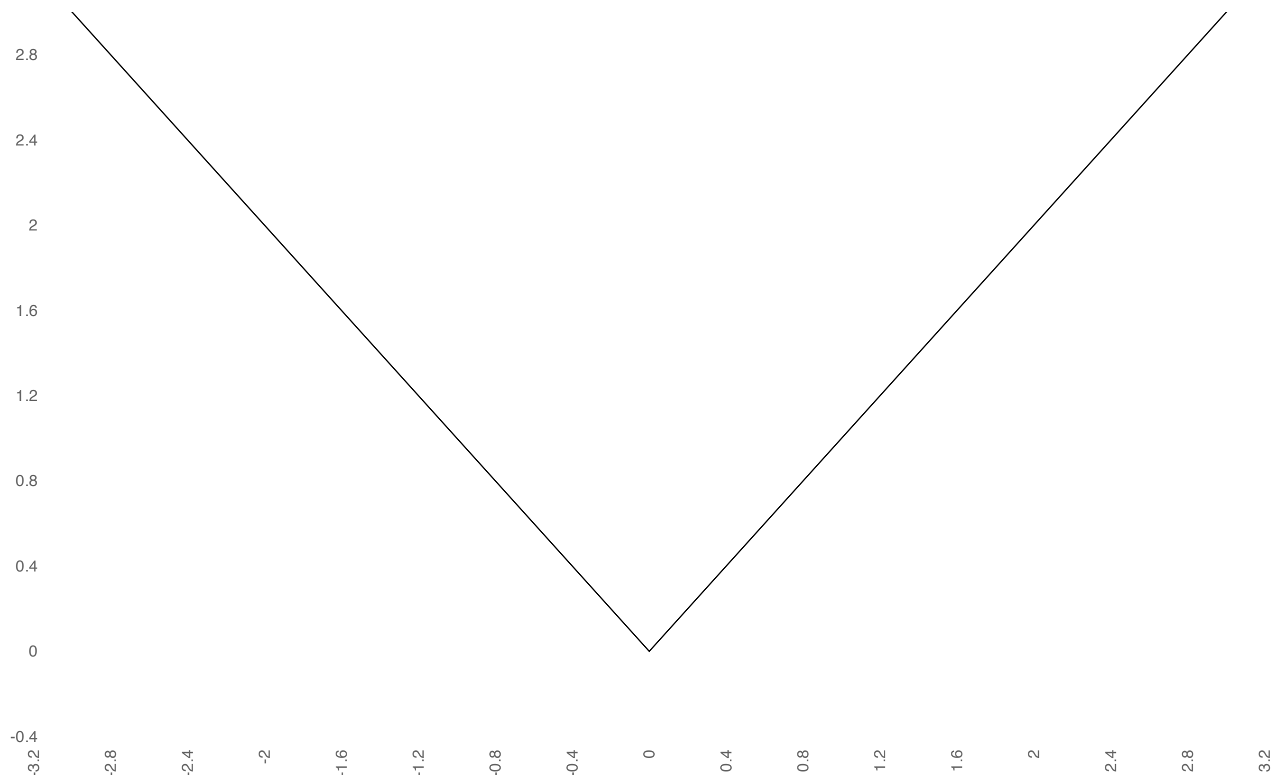
3D LP example

$$max\ 3z + x - 2y\ s.t.$$

$$|x| + |y| + |z| \leq 1$$

… not an LP! But …

Absolute value function



- $|x|$ *is always equal to either x or –x*

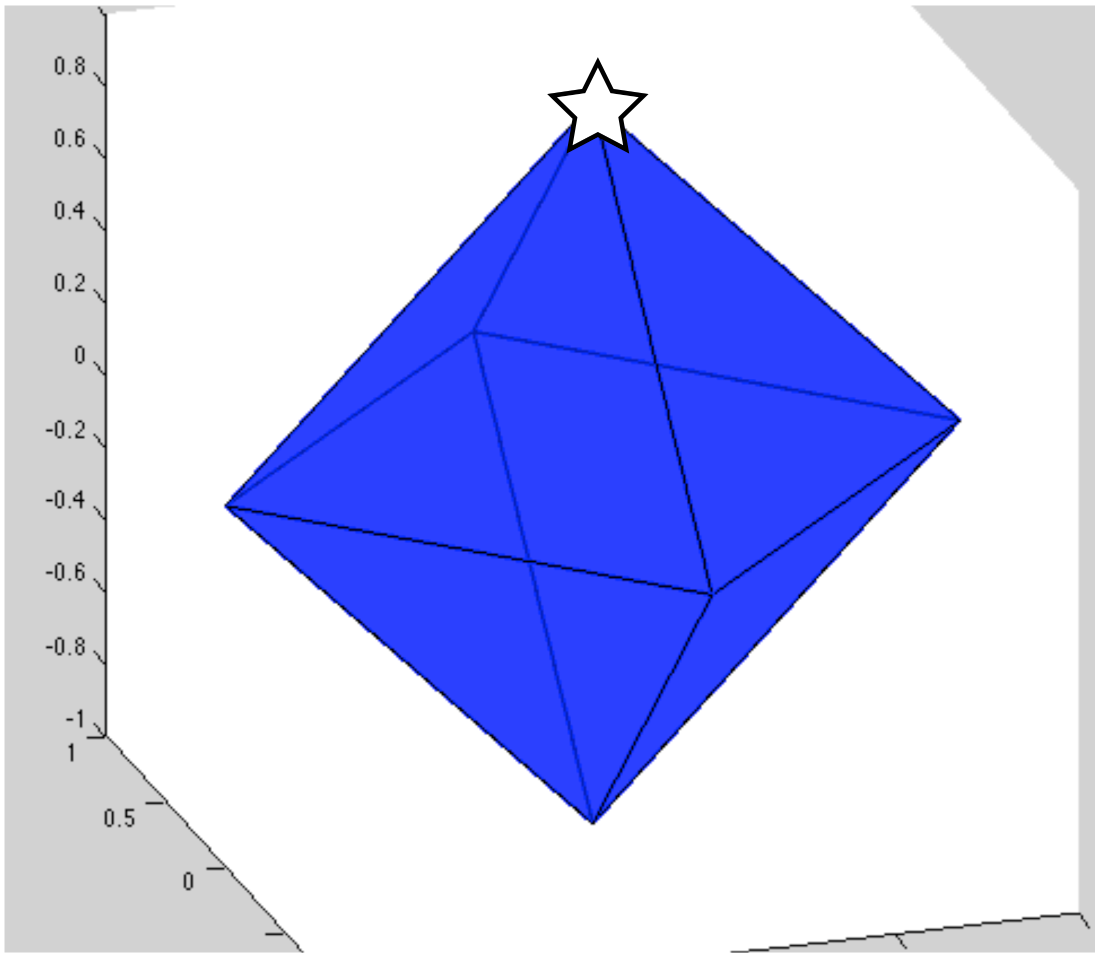
3D LP example

$max\ 3z + x - 2y\ s.t.$

$|x| + |y| + |z| \leq 1$

$$\Leftrightarrow \quad \begin{array}{ll} \multicolumn{2}{l}{max\ 3z + x - 2y\ s.t.} \\ x + y + z \leq 1 & -x + y + z \leq 1 \\ x + y - z \leq 1 & -x + y - z \leq 1 \\ x - y + z \leq 1 & -x - y + z \leq 1 \\ x - y - z \leq 1 & -x - y - z \leq 1 \end{array}$$

3D LP example



Notation: vector inequalities

- *For a vector of variables x and a constant matrix A and vector b,*

  $$Ax \leq b$$

  is interpreted componentwise

Vector inequalities

- *E.g.,* $Av \le b \iff$ $x + y + z \le 1$, $x + y - z \le 1$, $x - y + z \le 1$, $\ldots$

  if we define

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad v = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Complexity

- *There exist poly-time algorithms for LPs*
  - *e.g., ellipsoid, logarithmic barrier*
  - *rough estimate: n vars, m constraints ⇒ ~50–200 × cost of $n \times m$ regression*
- *No **strongly polynomial** LP algorithms known—interesting open question*
  - *simplex is "almost always" polynomial*

Complexity

- *ILPs and MILPs are complete for NP-opt*
  - *so, no poly-time algos unless P=NP*
- *Typically solved by search + smart techniques for ordering & pruning nodes*
- *E.g., branch & cut*

Branch & bound (& cut)

$[schema, value] = bb(F, sch, bnd)$

$[v_{rx}, sch_{rx}] = relax(F, sch)$

if $integer(sch_{rx})$: return $[sch_{rx}, v_{rx}]$

if $v_{rx} \geq bnd$: return $[sch, v_{rx}]$

Pick variable x_i

$[sch^0, v^0] = bb(F, sch/(x_i: 0), bnd)$

$[sch^1, v^1] = bb(F, sch/(x_i: 1), min(bnd, v^0))$

if $(v^0 \leq v^1)$: return $[sch^0, v^0]$

else: return $[sch^1, v^1]$

Branch & bound (& cut)

[schema, value] = bb(F, sch, bnd)

  [v_{rx}, sch_{rx}] = relax(F, sch)

  if integer(sch_{rx}): return [sch_{rx}, v_{rx}]

  if $v_{rx} \geq$ bnd: return [sch, v_{rx}]

  Pick variable x_i

  [sch^0, v^0] = bb(F, sch/(x_i: 0), bnd)

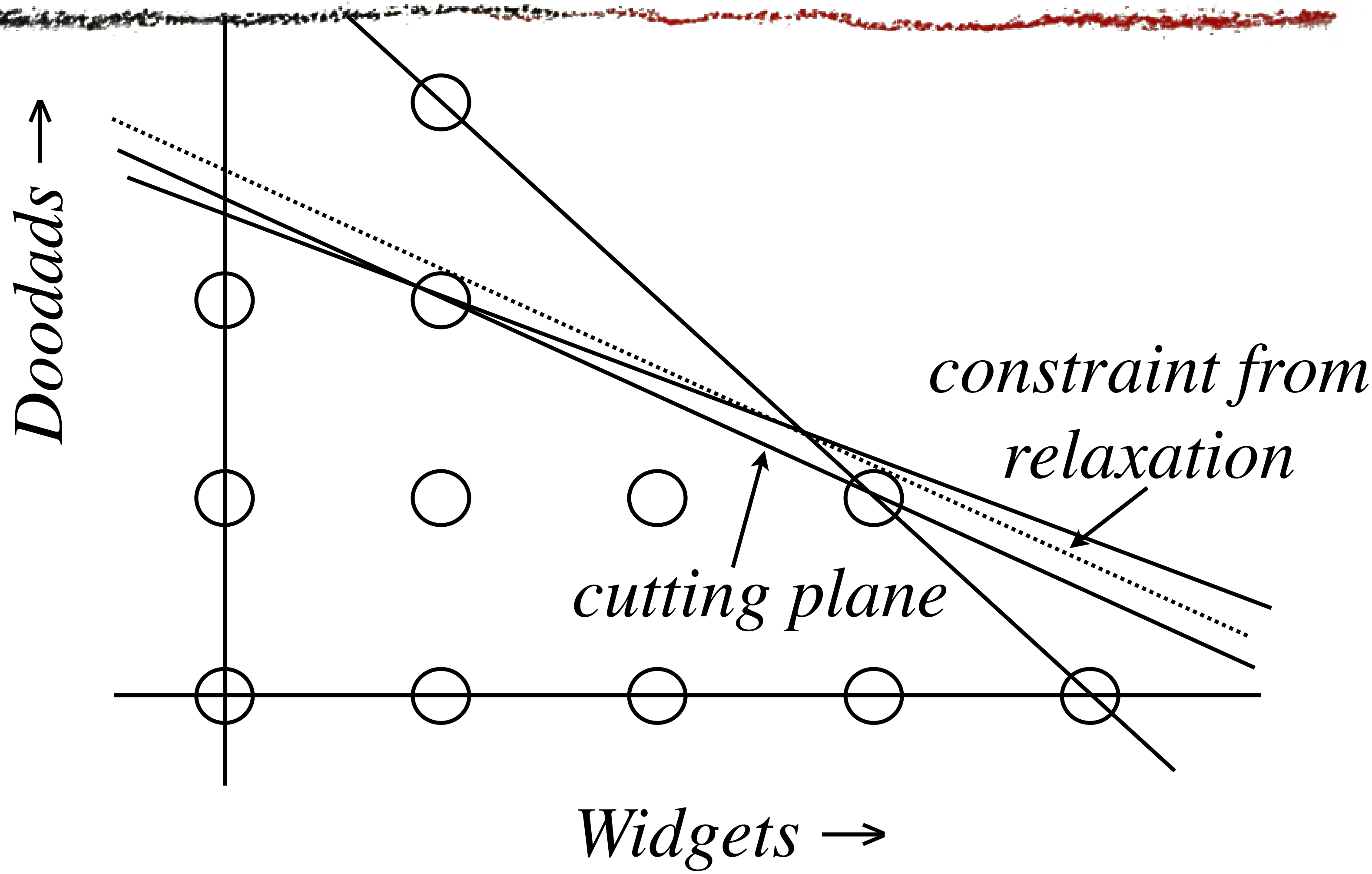
  [sch^1, v^1] = bb(F, sch/(x_i: 1), min(bnd, v^0))

  if ($v^0 \leq v^1$): return [sch^0, v^0]

  else: return [sch^1, v^1]

for branch & cut: add cuts as desired here, re-solve relaxation

Gomory cut example



Tension of cutting v. branching

- *After a branch it may become easier to generate more cuts*
    - *so easier as we go down the tree*
- *Cuts at a node N are valid at N's children*
    - *so it's worth spending more effort higher in the search tree*

ILPs and SAT

From ILP to SAT

- *0-1 ILP: all variables in {0, 1}*
- *SAT: 0-1 ILP, objective = constant, all constraints of form*

  $x + (1-y) + (1-z) \geq 1$
- *MAXSAT: 0-1 ILP, constraints of form*

  $x + (1-y) + (1-z) \geq s_j$

  maximize $s_1 + s_2 + \ldots$

DPLL+CL vs. branch & cut

- *Both are DFS + propagation + learning*
  - *DFS nodes = partial assignments*
  - *DFS neighborhood = branch on a question (e.g., assign a variable)*
  - *propagation = unit resolution / LP*
  - *learning = clause learning / cut generation*

Propagation

Propagation

- *Unit clauses (e.g., $\neg x$, y) translate to*

Propagation

- *Unit clauses (e.g., $\neg x$, y) translate to*
  - $(1-x) \geq 1 \iff x \leq 0$

Propagation

- *Unit clauses (e.g., $\neg x$, y) translate to*
  - $(1-x) \geq 1 \;\Leftrightarrow\; x \leq 0$
  - $y \geq 1$

Propagation

- *Unit clauses (e.g., ¬x, y) translate to*
  - $(1-x) \geq 1 \;\Leftrightarrow\; x \leq 0$
  - $y \geq 1$
- *Combined with* $0 \leq x \leq 1$, $0 \leq y \leq 1$, *unit clause constraints allow LP to completely determine x and y*

Propagation

- *Unit clauses (e.g., $\neg x$, y) translate to*
  - $(1-x) \geq 1 \;\Leftrightarrow\; x \leq 0$
  - $y \geq 1$
- *Combined with $0 \leq x \leq 1$, $0 \leq y \leq 1$, unit clause constraints allow LP to completely determine x and y*
- *So, LP is strictly stronger than unit resolution*

LP and resolution

- *What about more general resolutions?*
  - $(x \vee \neg y \vee \neg z) \wedge (z \vee a)$
  - $(x \vee \neg y \vee \neg z) \wedge (z \vee \neg y \vee a)$

Cuts and clause learning

- *So, LP + Gomory can duplicate any resolution*
- *In particular, some sequence of Gomory cuts can give us any learnable clause*
  - *DPLL+CL for SAT is just a special case of branch & cut*

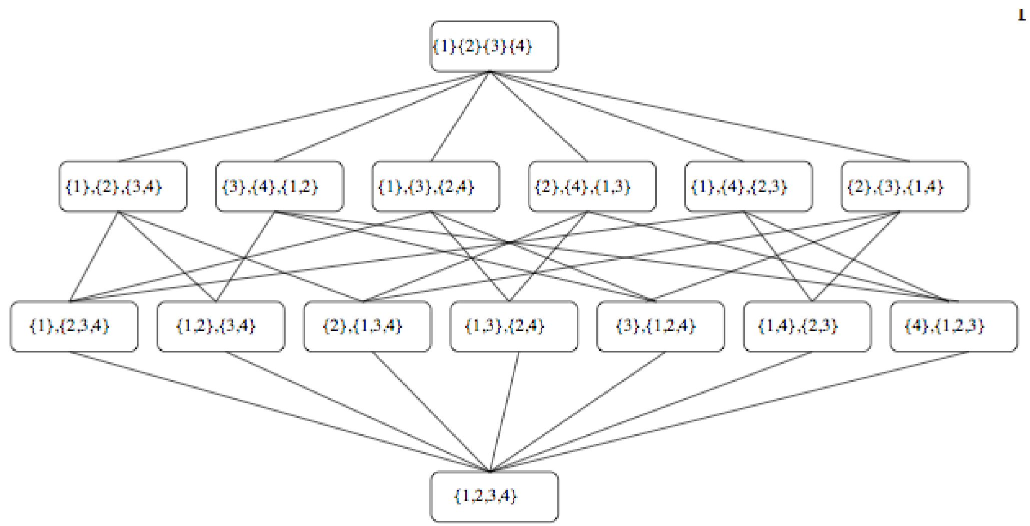
LP bounds in SAT

- *What would be pros and cons of using LP relaxation to get bounds in DPLL for SAT?*

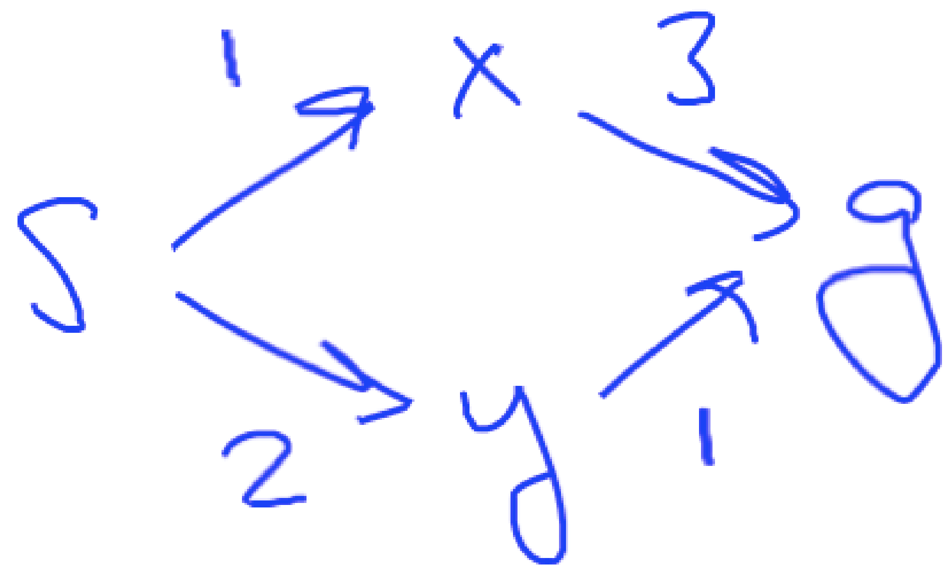
Examples

Examples

- *Any problem in NP, since "does MILP have solution of value z?" is NP-complete*
- *E.g., allocation problems like clearing combinatorial auctions*



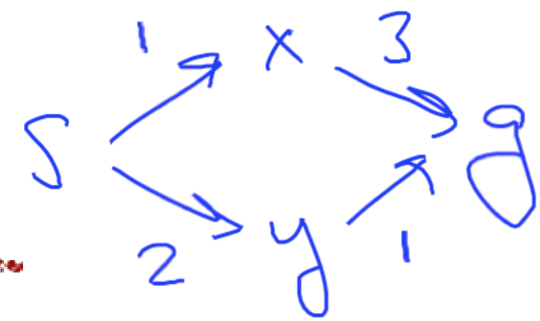
Path planning



- *Find the min-cost path: 0-1 variables*

$$P_{sx}, P_{sy}, P_{xg}, P_{yg} \geq 0$$

Path planning



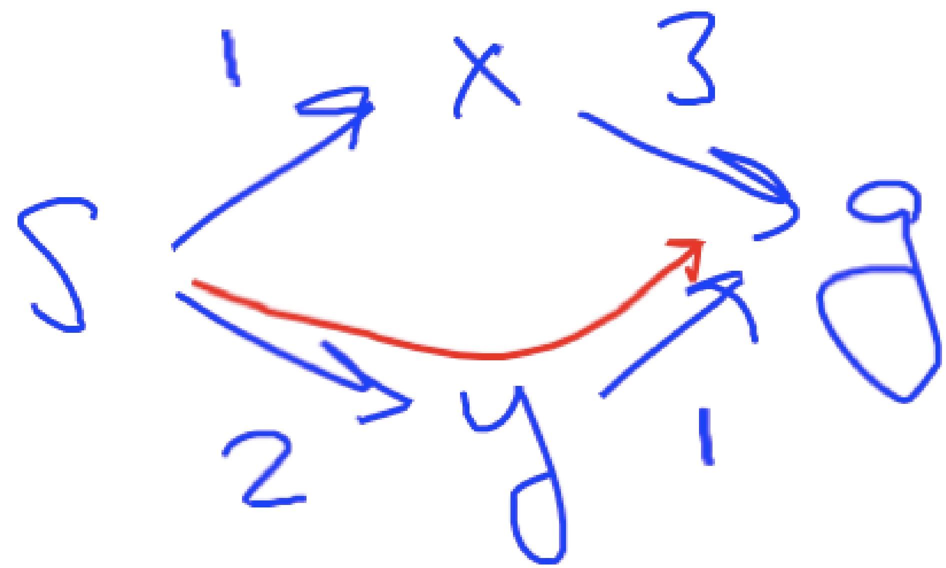
min

$$p_{sx} + 3p_{xg} + 2p_{sy} + p_{yg}$$

st

$$\begin{aligned}
p_{sx} \quad\quad\quad + p_{sy} \quad\quad\quad &= 1 \\
-p_{sx} + p_{xg} \quad\quad\quad\quad\quad\quad &= 0 \\
-p_{sy} + p_{yg} &= 0 \\
-p_{xg} \quad\quad - p_{yg} &= -1
\end{aligned}$$

Optimal solution



$$p_{sy} = p_{yg} = 1, \quad p_{sx} = p_{xg} = 0, \quad cost\ 3$$

Matrix form

$$\min \ (1\ 3\ 2\ 1)\, p$$

st

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

$$p \geq 0$$

Matrix form

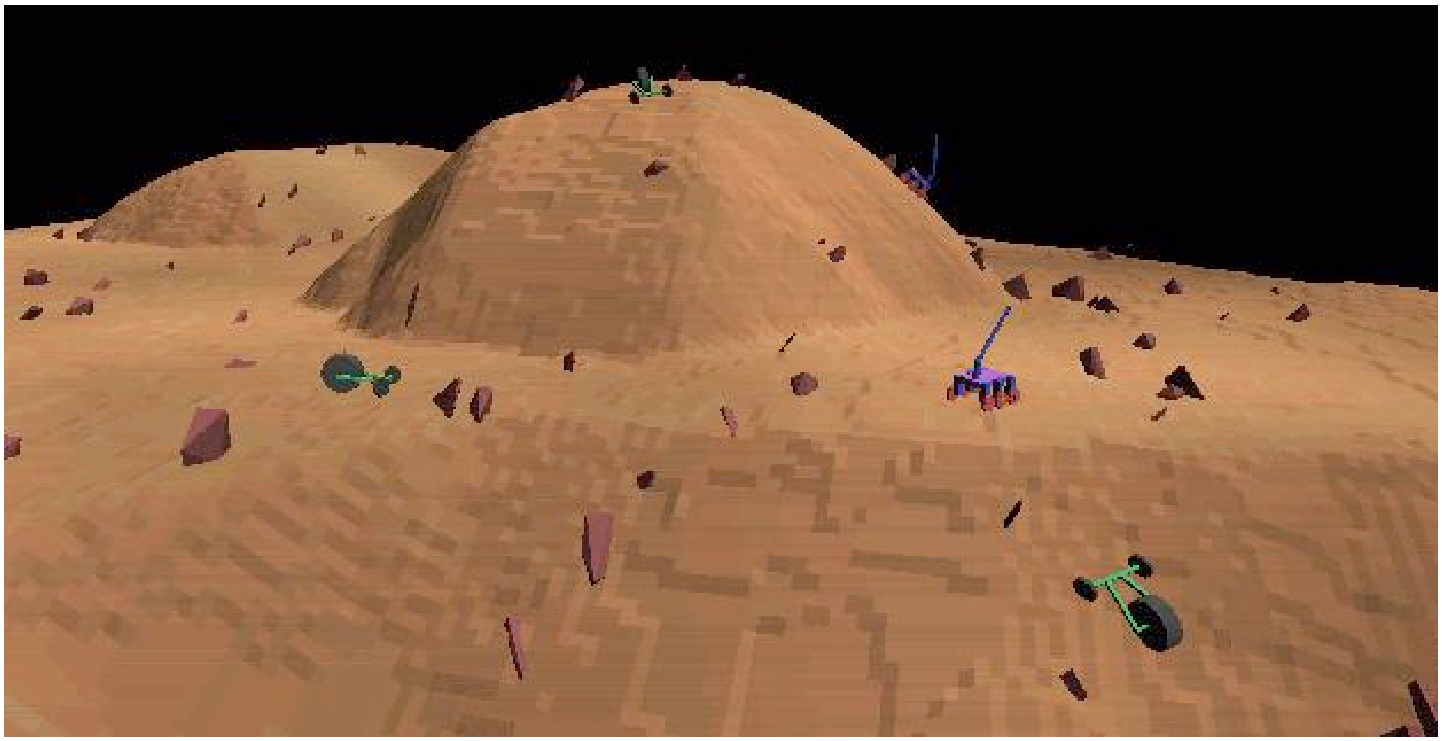
$$\min \begin{pmatrix} 1 & 3 & 2 & 1 \end{pmatrix} p$$

st

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

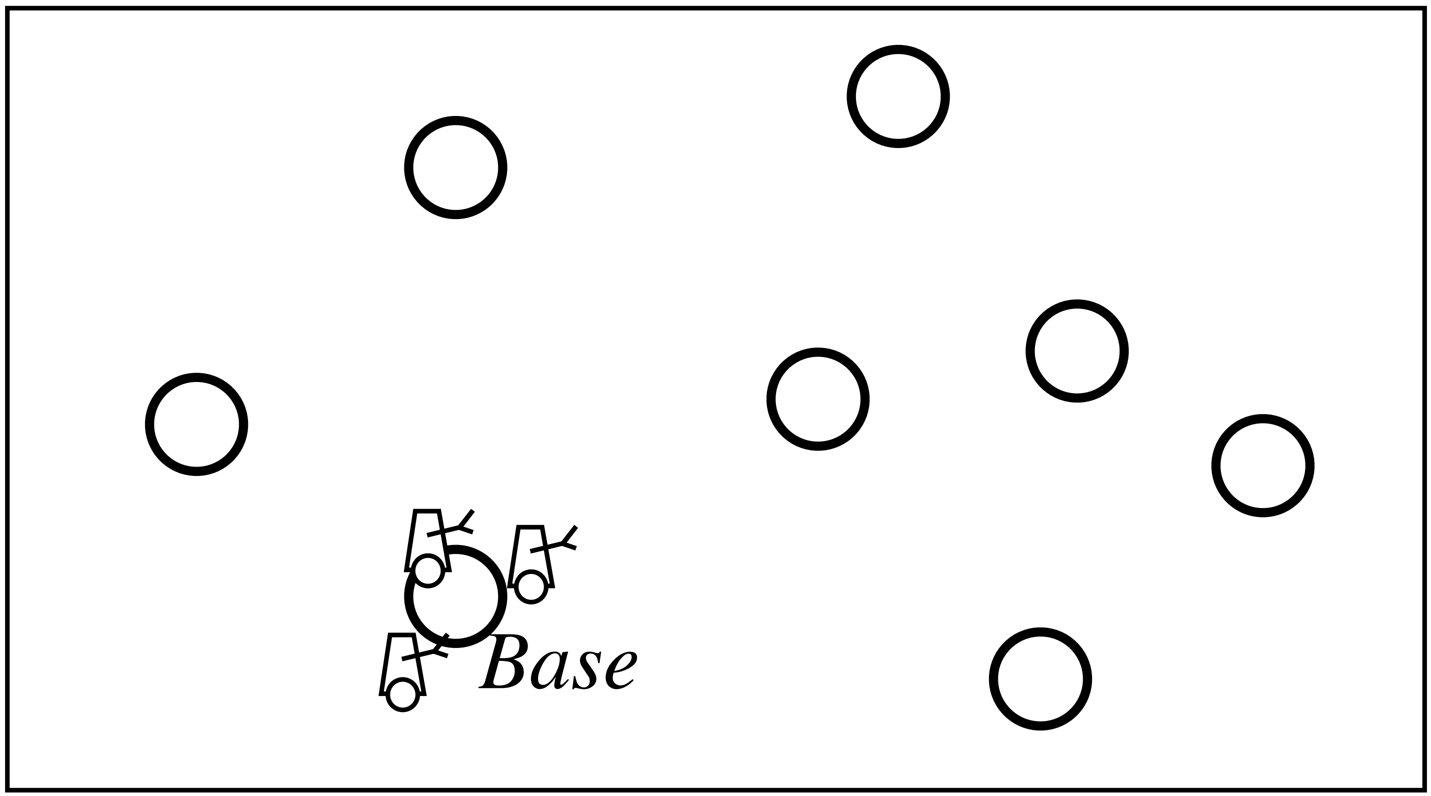
?? $p \in \{0,1\}^4$ $\quad p \geq 0$

Example: robot exploration task assignment

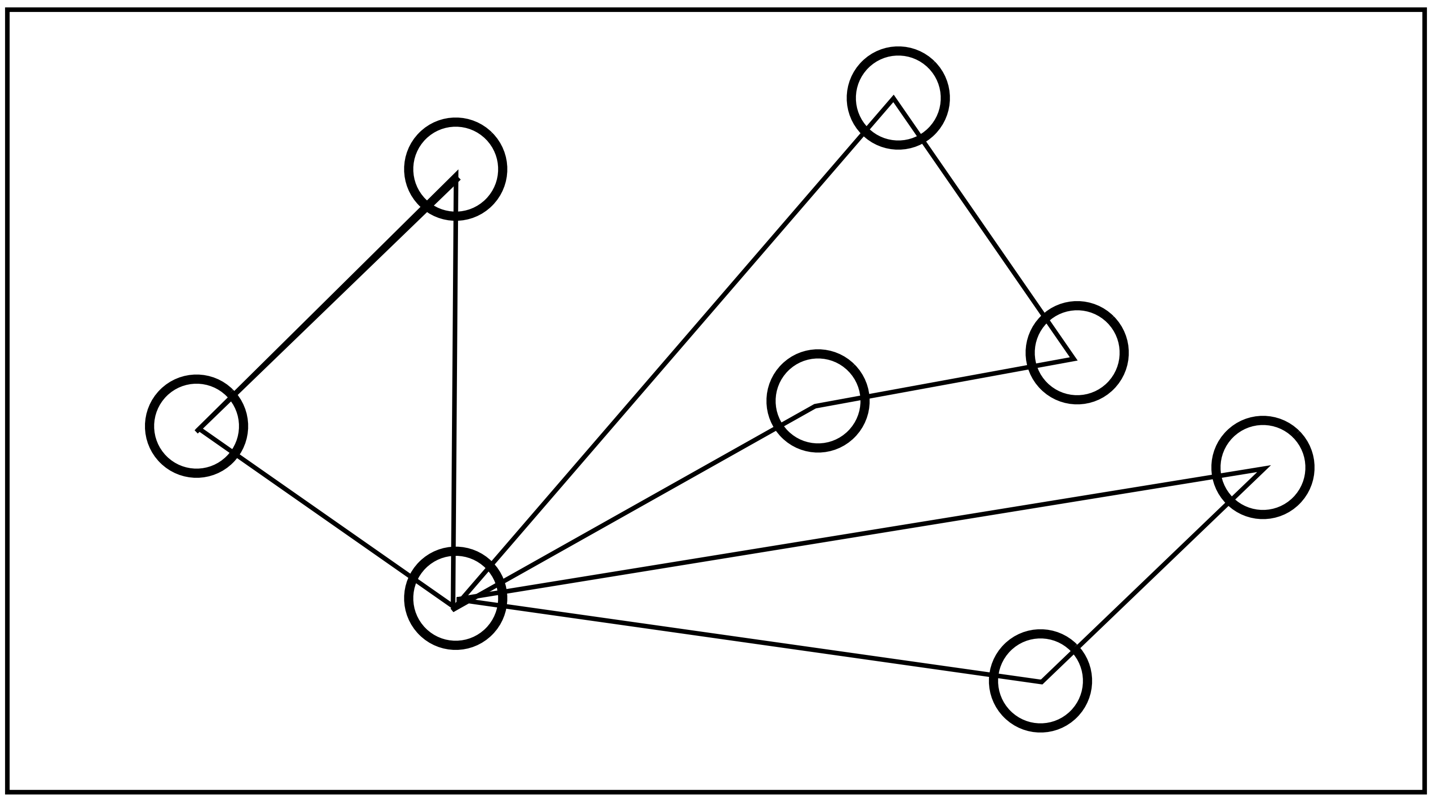


- *Team of robots must explore unknown area*

Points of interest



Exploration plan



ILP

- *Variables (all 0/1):*

  z_{ri} = *robot r does task i*

  x_{rijt} = *robot r uses edge ij at step t*

- *Minimize cost = [path cost – task bonus]*

$$\sum_{rijt} x_{rijt}\, c_{rijt} - \sum_{ri} z_{ri}\, b_{ri}$$

r indexes robots, i&j index tasks, t indexes steps

Constraints

- *Assigned tasks:* $\forall r, j, \sum_{it} x_{rijt} \geq z_{rj}$
- *One edge per step:* $\forall r, t, \sum_{ij} x_{rijt} = 1$
  - *self-loops @ base to allow idling*
- *For each i, path forms a tour from base:*
  - $\forall r, i, t, \sum_j x_{rjit} = \sum_j x_{rij(t+1)}$
  - *edges used into node = edges used out*
  - *except at times 0 and T*

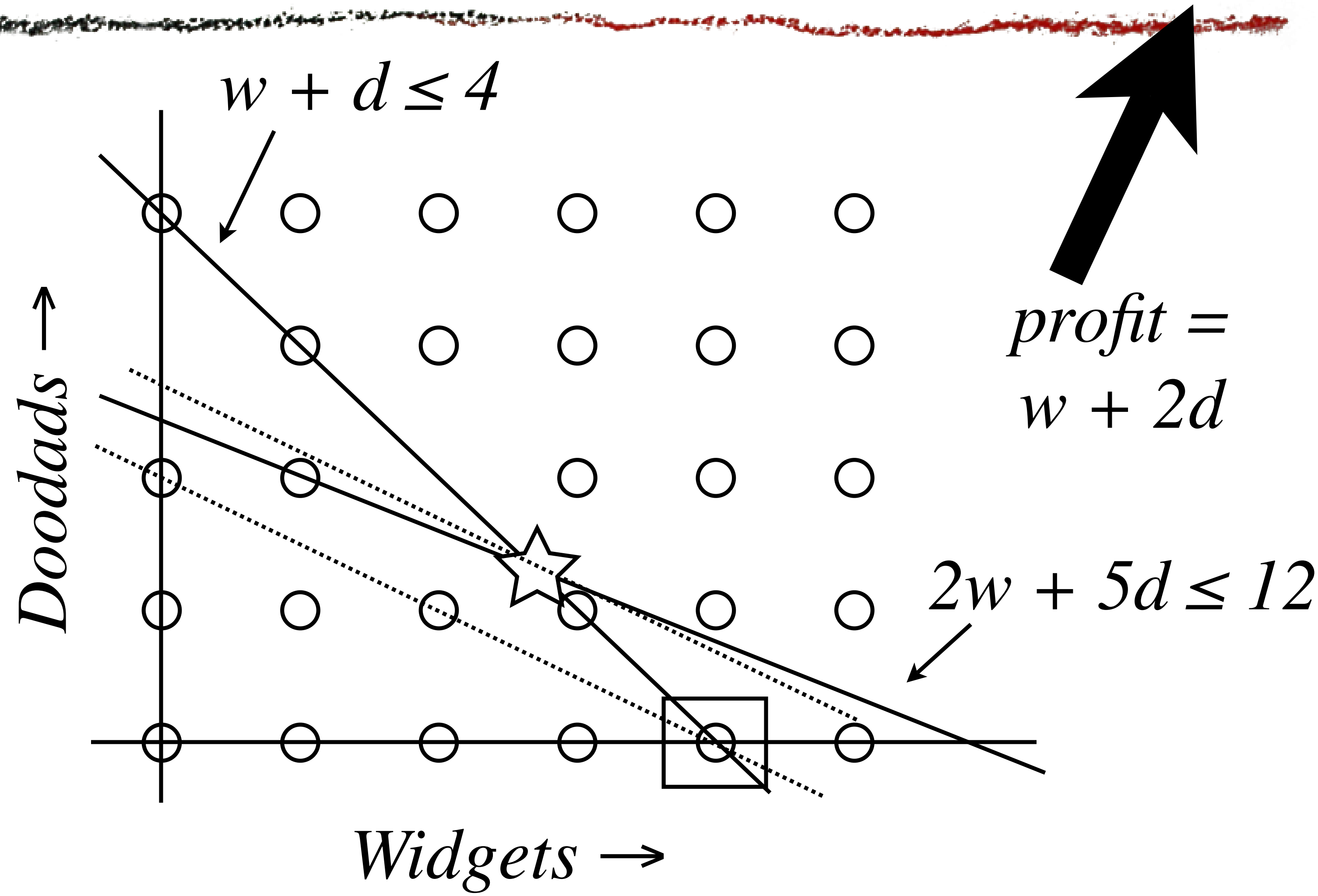
r indexes robots, i&j index tasks, t indexes steps

Duality

Branch & bound summary

- *Branch & bound idea 1: if we have a solution with profit 3, add a constraint "profit ≥ 3"*
  - *If we then find a solution with profit 4, replace constraint with "profit ≥ 4"*
- *B&B idea 2: use LP relaxations to get constraints like "profit ≤ 5 1/3"*

Factory example



Early stopping

- *So, we have a solution of profit $4*
- *And we know the best solution has profit no more than $5 1/3*
- *If we're lazy, we can stop now*
- *Can we get smarter? Or lazier?*

What if we're *really* lazy?

- *To get our bound: had to solve the LP and find its exact optimum*
- *Can we do less work?*
- *Idea: find a suboptimal solution to LP?*
  - *Sadly, a non-optimal feasible point in the LP relaxation gives us no useful bound*

A simple bound

- *Recall:*
  - *constraint* $w + d \le 4$ *(limit on wood)*
  - *profit* $w + 2d$
- *Since* $w, d \ge 0$,
  - $profit = w + 2d \le 2w + 2d$
- *And, doubling both sides of constraint,*
  - $2w + 2d \le 8 \quad \Rightarrow \quad profit \le 8$

The same trick works twice

- *Try other constraint (steel use)*
  - $2w + 5d \leq 12$
- $2 * profit = 2w + 4d \leq 2w + 5d \leq 12$
- *So* $profit \leq 6$

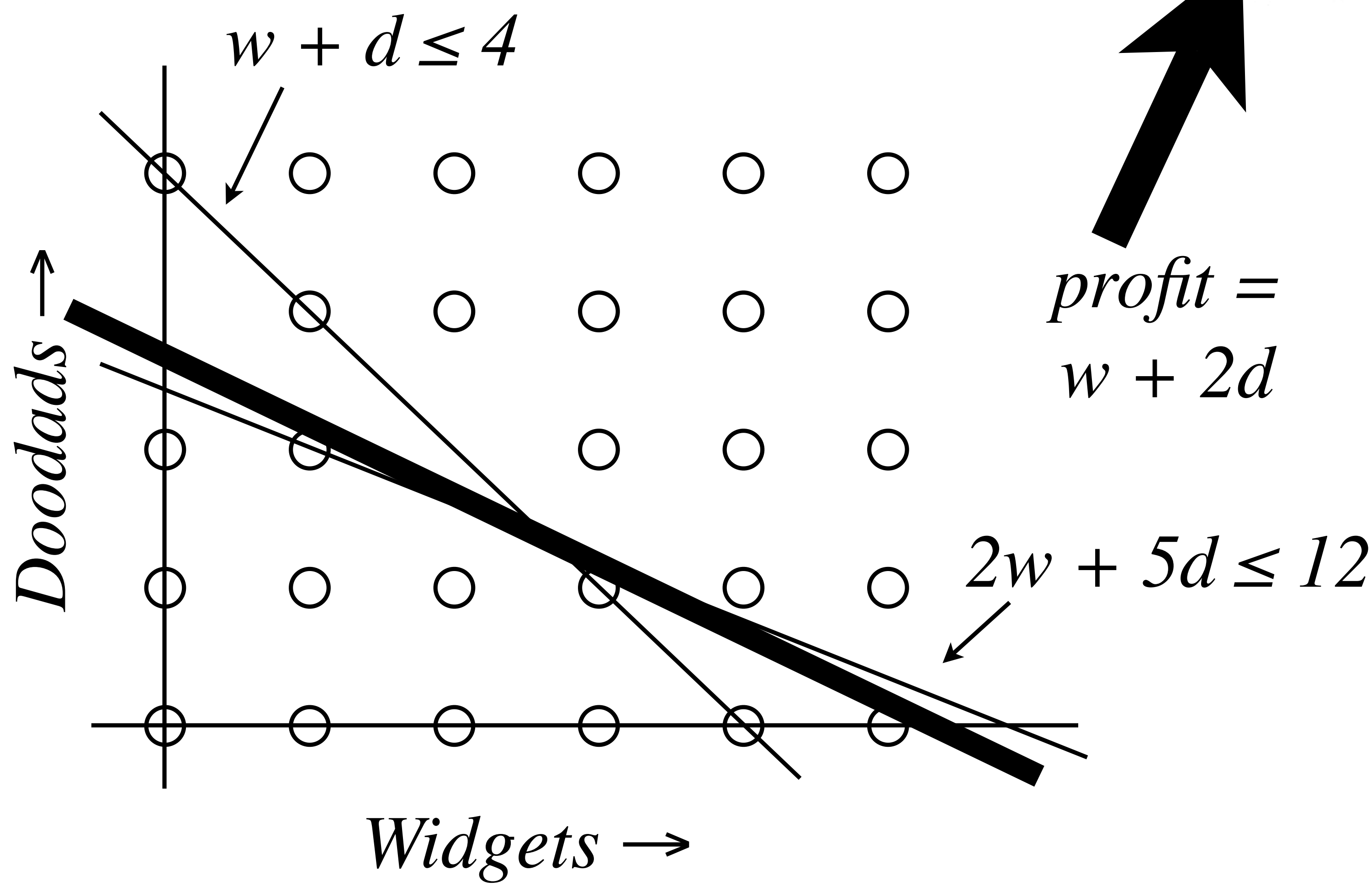
In fact it works infinitely often

- *Could take any positive-weight linear combination of our constraints*
  - *negative weights would flip sign*

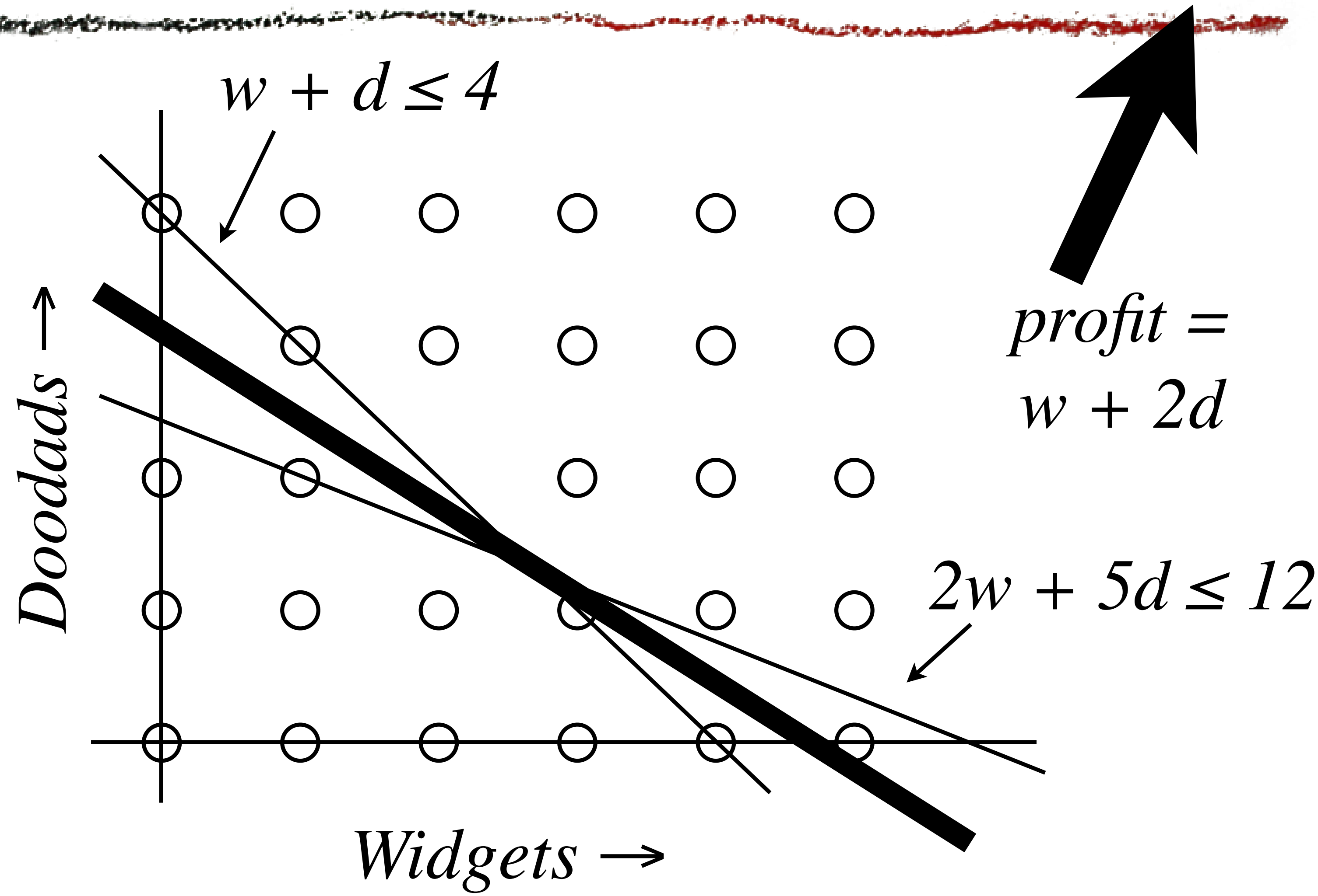
$$a\,(w + d - 4) + b\,(2w + 5d - 12) \le 0$$

$$(a + 2b)\,w + (a + 5b)\,d \le 4a + 12b$$

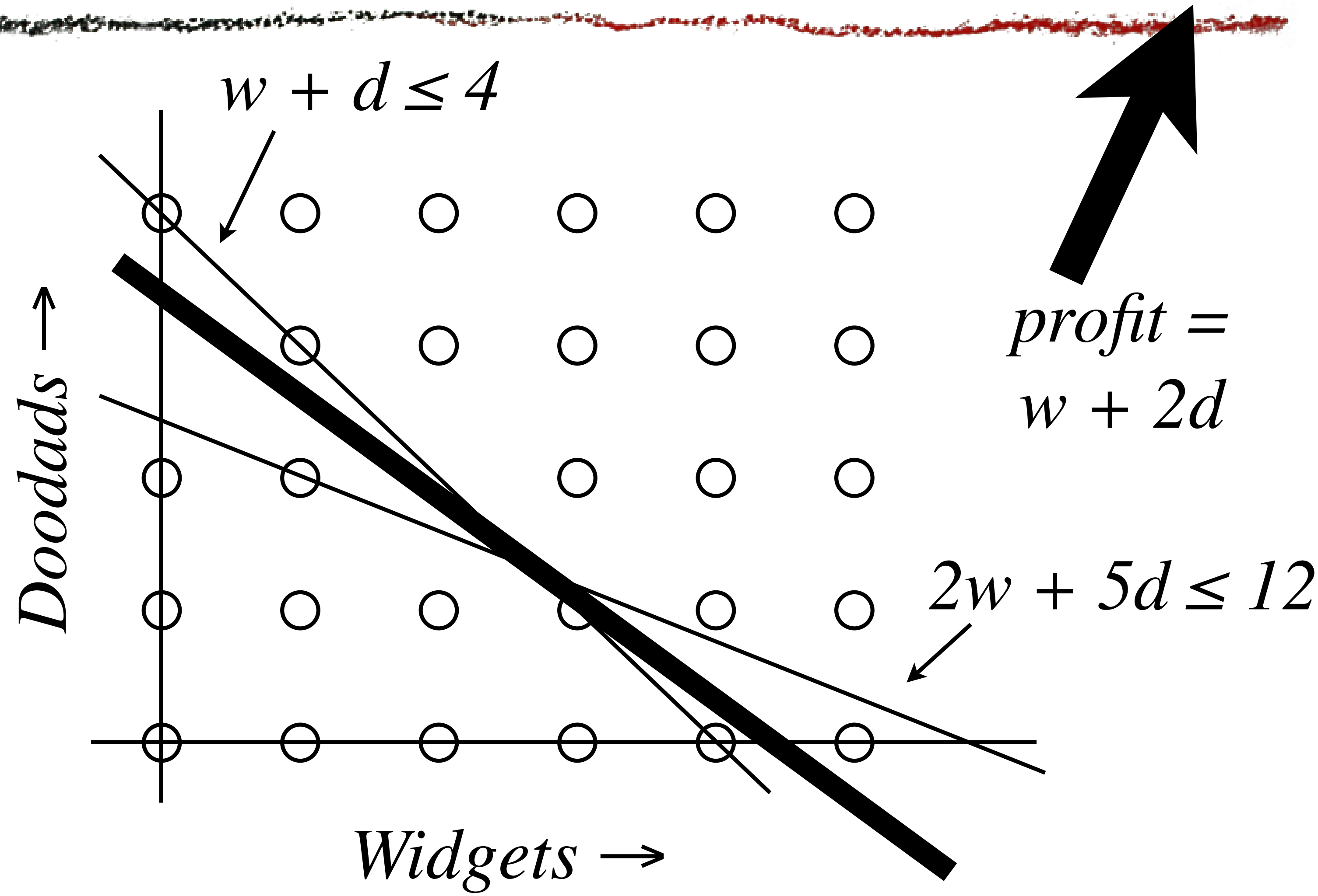
Geometrically



Geometrically



Geometrically



Bound

- $(a + 2b)\, w + (a + 5b)\, d \leq 4a + 12b$
- $profit = 1w + 2d$
- So, if $1 \leq (a + 2b)$ and $2 \leq (a + 5b)$, we know that $profit \leq 4a + 12b$

Bound

- $\boxed{(a + 2b)}\, w + (a + 5b)\, d \leq 4a + 12b$
- $profit = \boxed{1}\, w + 2d$
- *So, if* $1 \leq (a + 2b)$ *and* $2 \leq (a + 5b)$*, we know that* $profit \leq 4a + 12b$

Bound

- $(\boxed{a + 2b})\, w + (\boxed{a + 5b})\, d \le 4a + 12b$
- $profit = \boxed{1}w + \boxed{2}d$
- *So, if* $1 \le (a + 2b)$ *and* $2 \le (a + 5b)$*, we know that* $profit \le 4a + 12b$

The best bound

- *If we search for the tightest bound, we have an LP:*

  minimize 4a + 12b such that

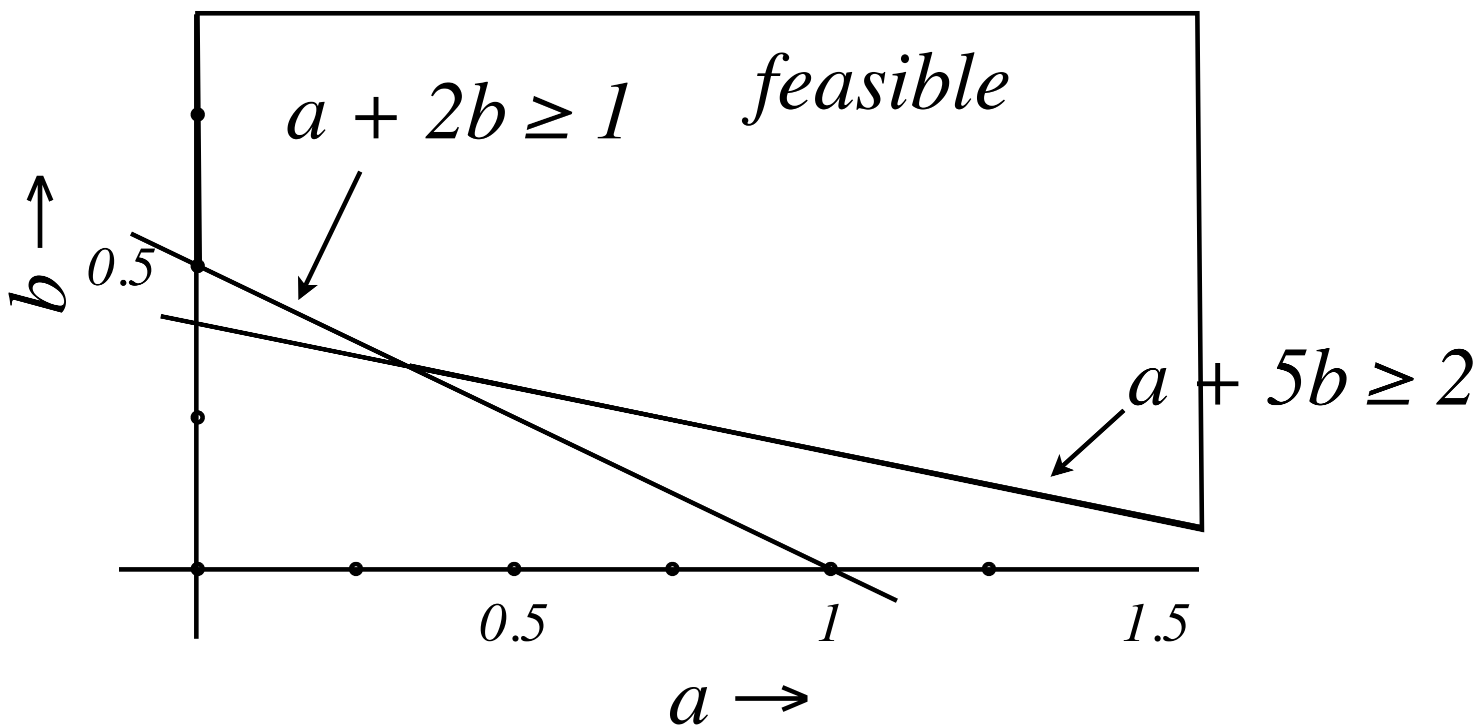
  $a + 2b \geq 1$

  $a + 5b \geq 2$

  $a, b \geq 0$

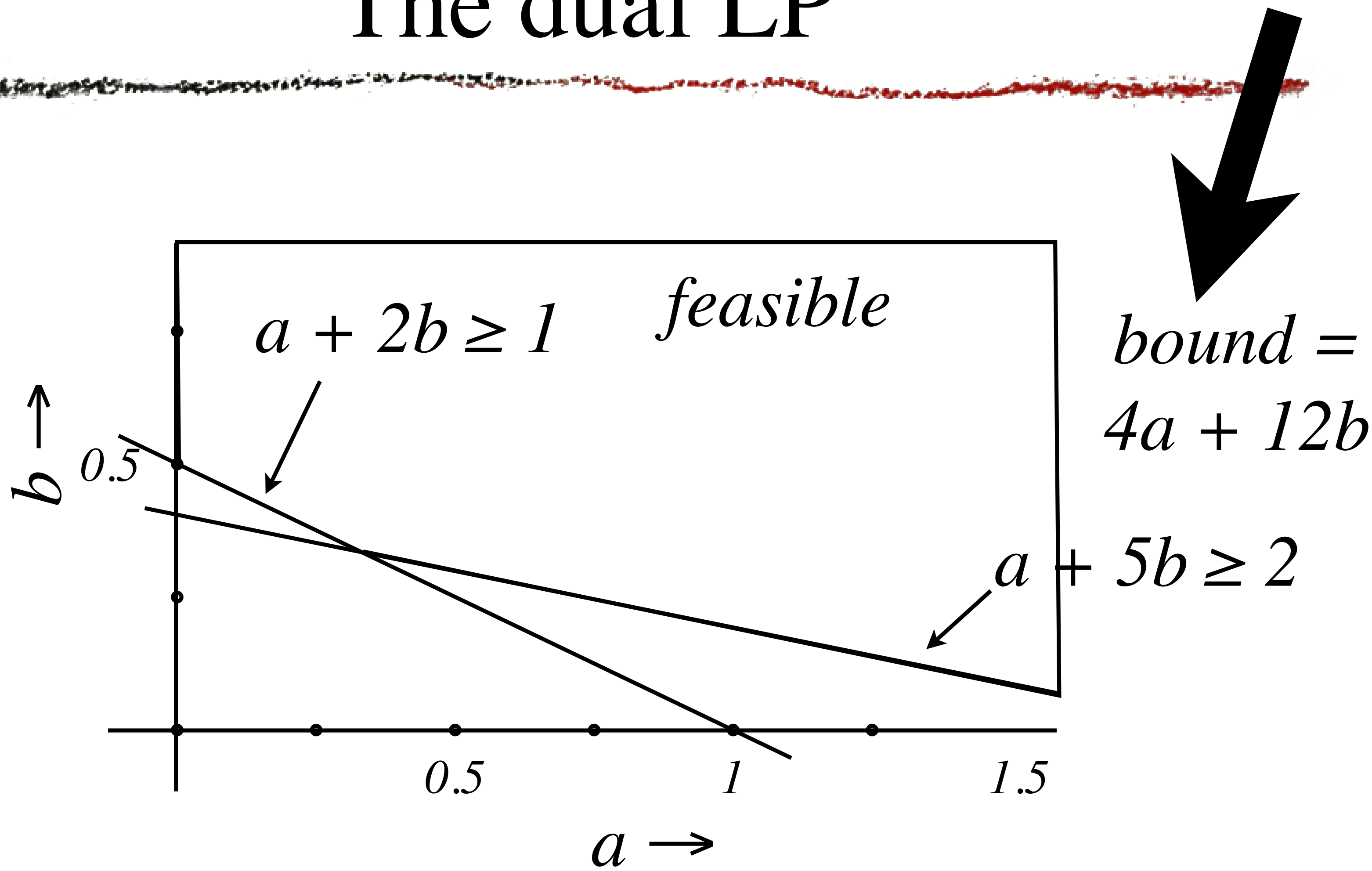
- *Called the* ***dual***

The dual LP



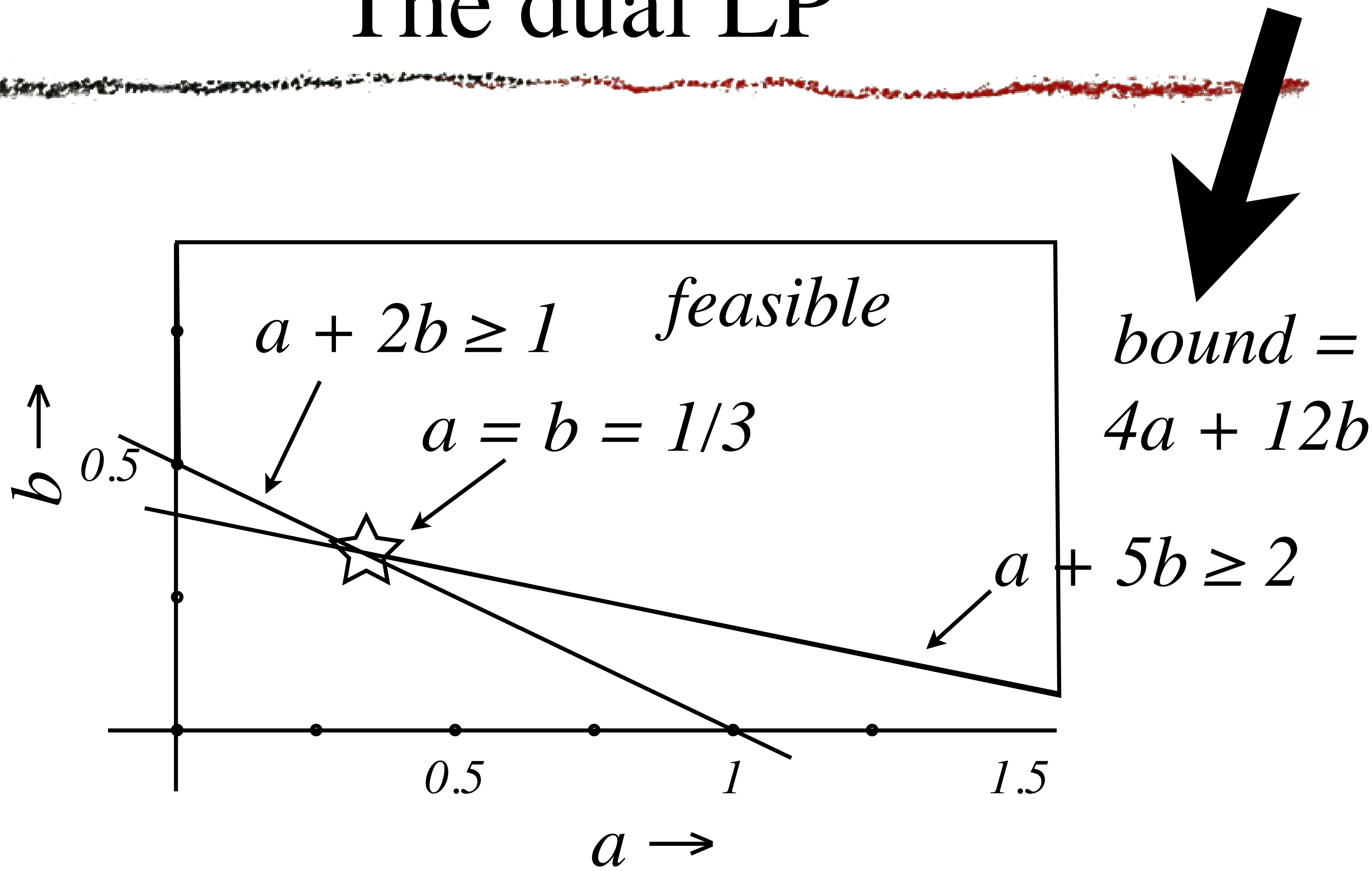
feasible
$a + 2b \geq 1$
$a + 5b \geq 2$
$b \rightarrow$
0.5
0.5
1
1.5
$a \rightarrow$

The dual LP

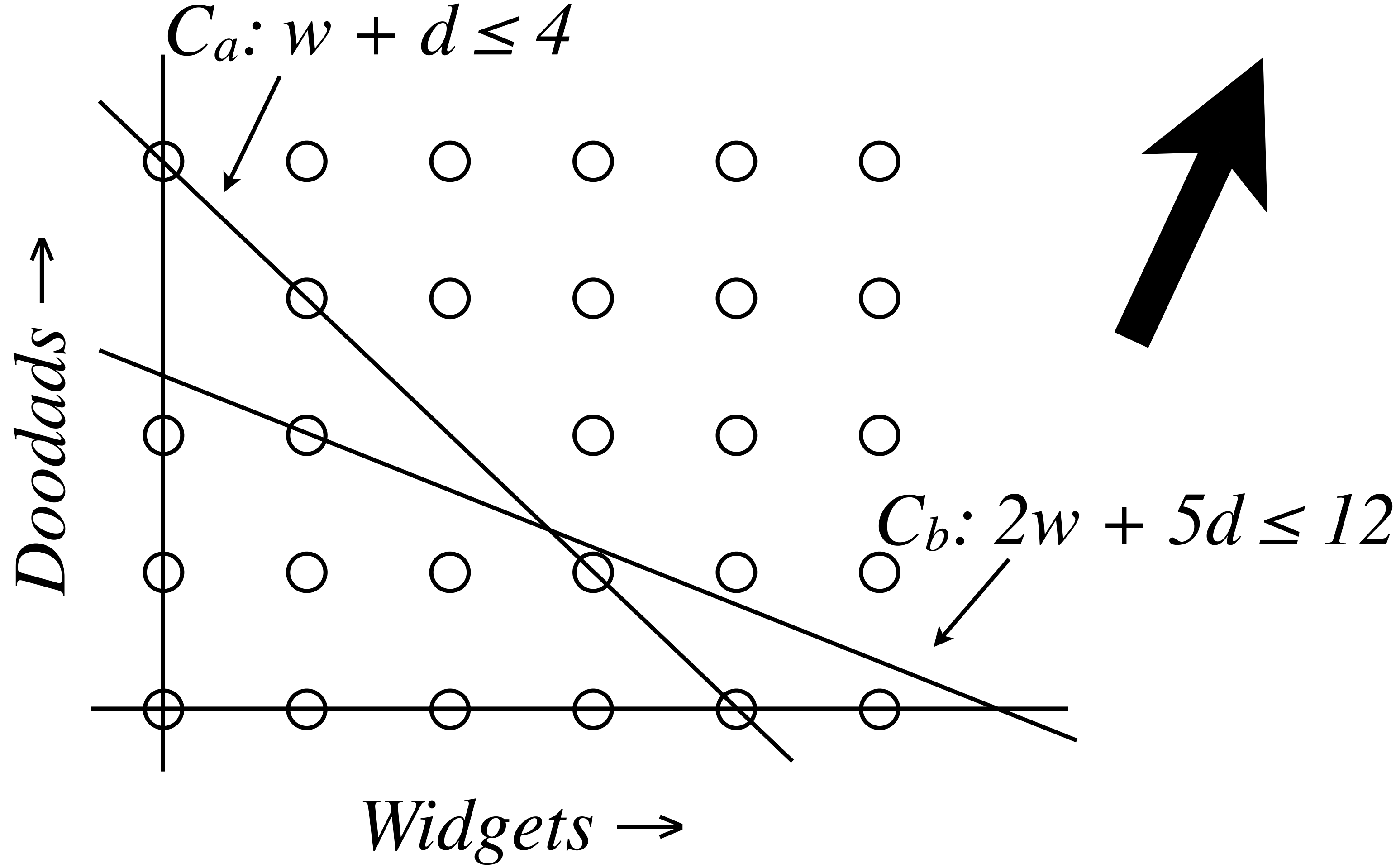


feasible
$a + 2b \geq 1$
$a + 5b \geq 2$
bound = $4a + 12b$
$b \rightarrow$
0.5
0.5
1
1.5
$a \rightarrow$

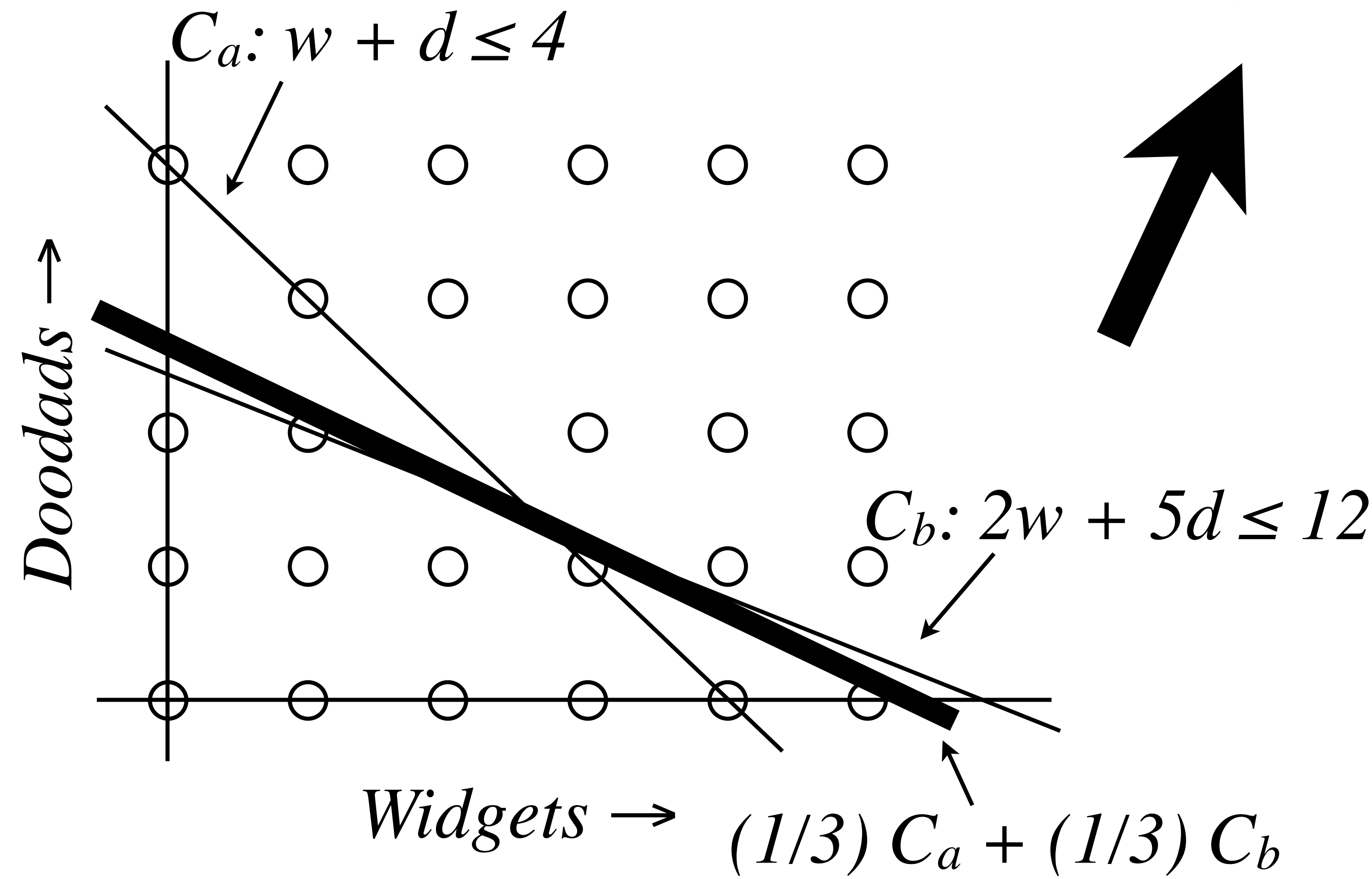
The dual LP



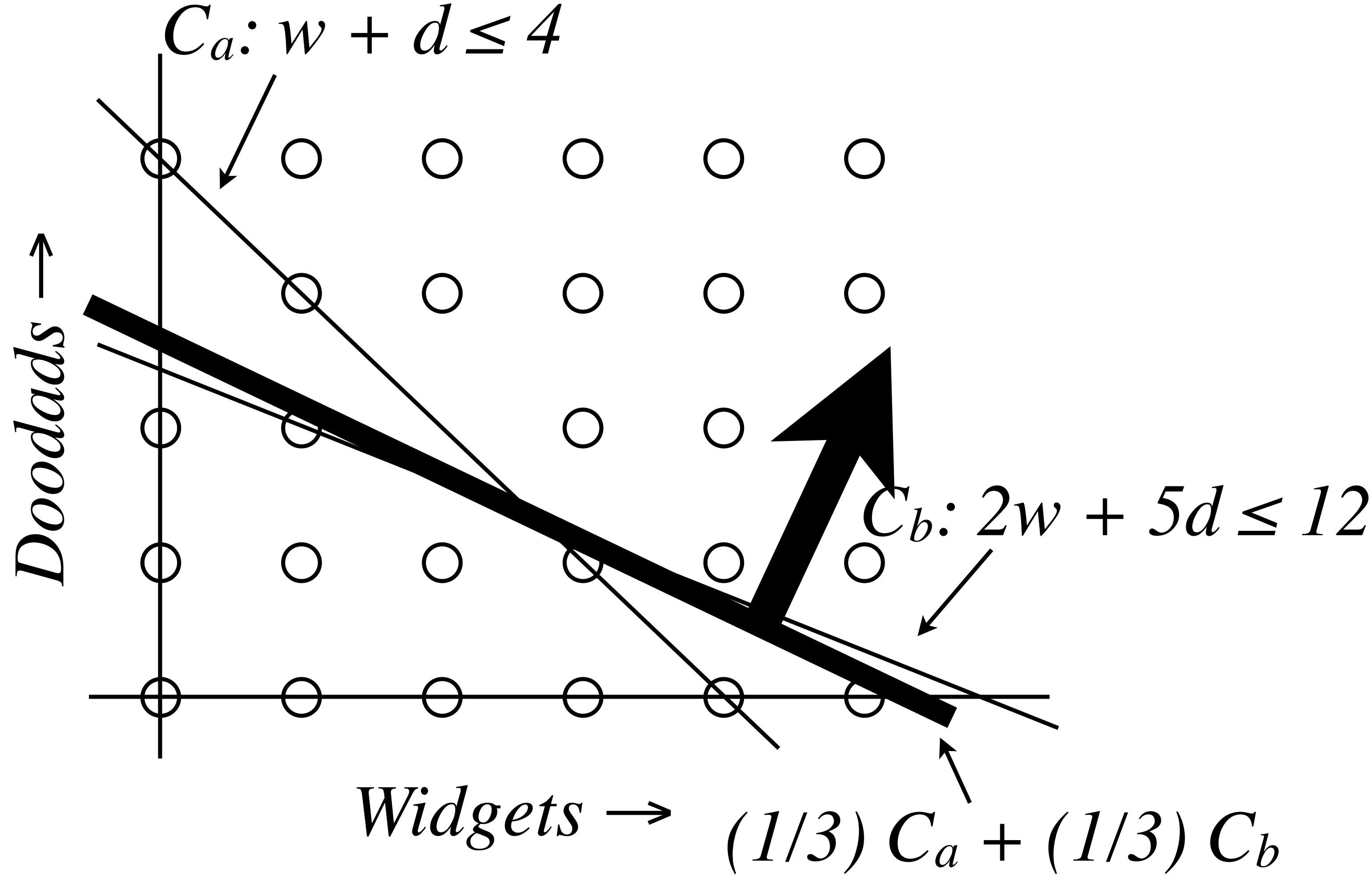
Best bound, as primal constraint



Best bound, as primal constraint



Best bound, as primal constraint



Bound from dual

- *$a = b = 1/3$ yields bound of*

  $4a + 12b = 16/3 = 5\ 1/3$
- *Same as bound from original relaxation!*
- *No accident: dual of an LP always* has same objective value*

So why bother?

- *Reason 1: any feasible solution to dual yields upper bound (compared with only optimal solution to primal)*
- *Reason 2: dual might be easier to work with*

Recap

- *Each feasible point of dual is an upper bound on objective*
- *Each feasible point of primal is a lower bound on objective*
  - *for ILP, each integral feasible point*

Recap

- *If search in primal finds a feasible point w/ objective 4*
- *And approximate solution to dual has value 6*
  - *approximate = feasible but not optimal*
- *Then we know we're ≥ 66% of best*

Duality w/ equality

Recall duality w/ inequality

- *Take a linear combination of constraints to bound objective*
- $(\boxed{a + 2b})\, w + (\boxed{a + 5b})\, d \leq 4a + 12b$
- $profit = \boxed{1}w + \boxed{2}d$
- *So, if* $1 \leq (a + 2b)$ *and* $2 \leq (a + 5b)$*, we know that* $profit \leq 4a + 12b$

Equality example

- *minimize* y *subject to*
- $x + y = 1$
- $2y - z = 1$
- $x, y, z \geq 0$

Equality example

- *Want to prove bound* $y \geq \ldots$
- *Look at 2nd constraint:*

  $$2y - z = 1 \quad \Rightarrow$$

  $$y - z/2 = 1/2$$

- *Since* $z \geq 0$*, dropping* $-z/2$ *can only increase LHS* $\Rightarrow$
  - $y \geq 1/2$

Duality w/ equalities

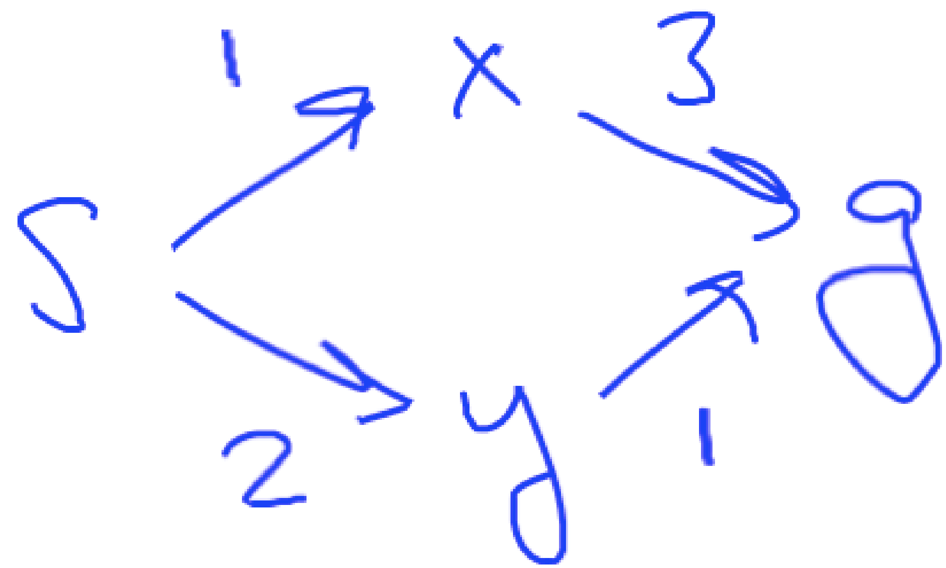
- *In general, could start from any linear combination of equality constraints*
  - *no need to restrict to +ve combination*
- $a\,(x + y - 1) + b\,(2y - z - 1) = 0$
- $a\,x + (a + 2b)\,y - b\,z = a + b$

Duality w/ equalities

- $a\,x + (a + 2b)\,y - b\,z = a + b$
- *As long as coefficients on LHS* $\leq (0, 1, 0)$,
  - *objective* $= 0\,x + 1\,y + 0\,z \geq a + b$
- *So, maximize* $a + b$ *subject to*
  - $a \leq 0$
  - $a + 2b \leq 1$
  - $-b \leq 0$

Duality example

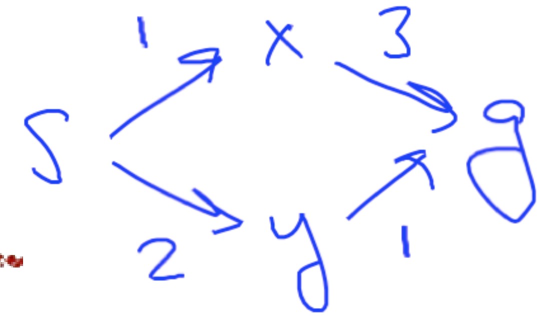
Path planning LP



- *Find the min-cost path: variables*

$P_{sx}, P_{sy}, P_{xg}, P_{yg} \geq 0$

Path planning LP



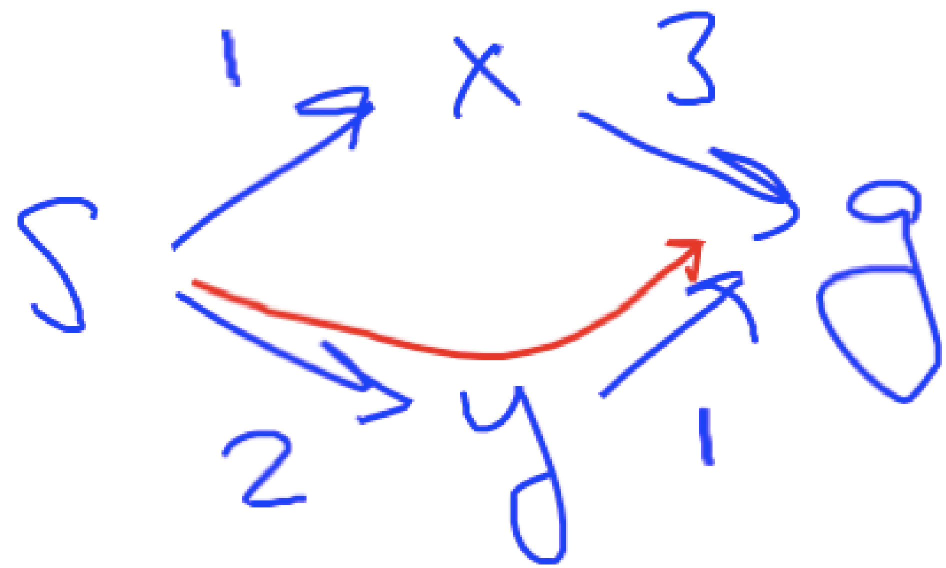
min

$$p_{sx} + 3p_{xg} + 2p_{sy} + p_{yg}$$

st

$$
\begin{aligned}
p_{sx} \qquad\qquad + p_{sy} \qquad\qquad &= 1 \\
-p_{sx} + p_{xg} \qquad\qquad\qquad &= 0 \\
- p_{sy} + p_{yg} &= 0 \\
- p_{xg} \qquad\qquad - p_{yg} &= -1
\end{aligned}
$$

Optimal solution



$p_{sy} = p_{yg} = 1, \quad p_{sx} = p_{xg} = 0, \quad cost\ 3$

Matrix form

$$\min \ (1\ 3\ 2\ 1)\, p$$

st

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

$$p \geq 0$$

Matrix form

$$\min \ (1\ 3\ 2\ 1)\, p$$

st

$$\begin{array}{c} \lambda_s \\ \lambda_x \\ \lambda_y \\ \lambda_g \end{array} \begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & -1 \end{pmatrix} p = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

$$p \geq 0$$

Dual

$$\max \quad \lambda_s - \lambda_g$$

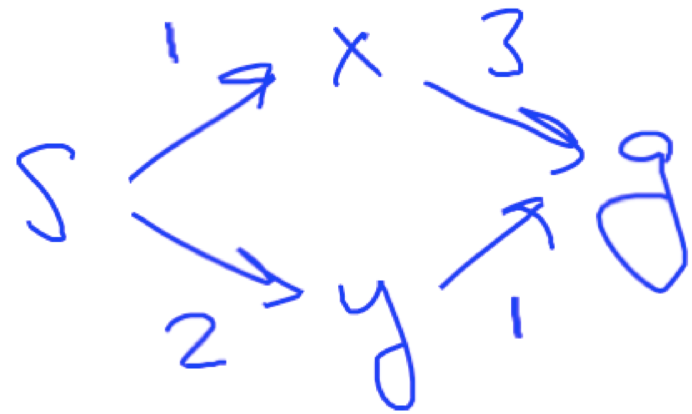
st

$$\lambda_s - \lambda_x \le 1$$

$$\lambda_x - \lambda_g \le 3$$

$$\lambda_s - \lambda_y \le 2$$

$$\lambda_y - \lambda_g \le 1$$



Optimal dual solution

max $\lambda_s - \lambda_g$

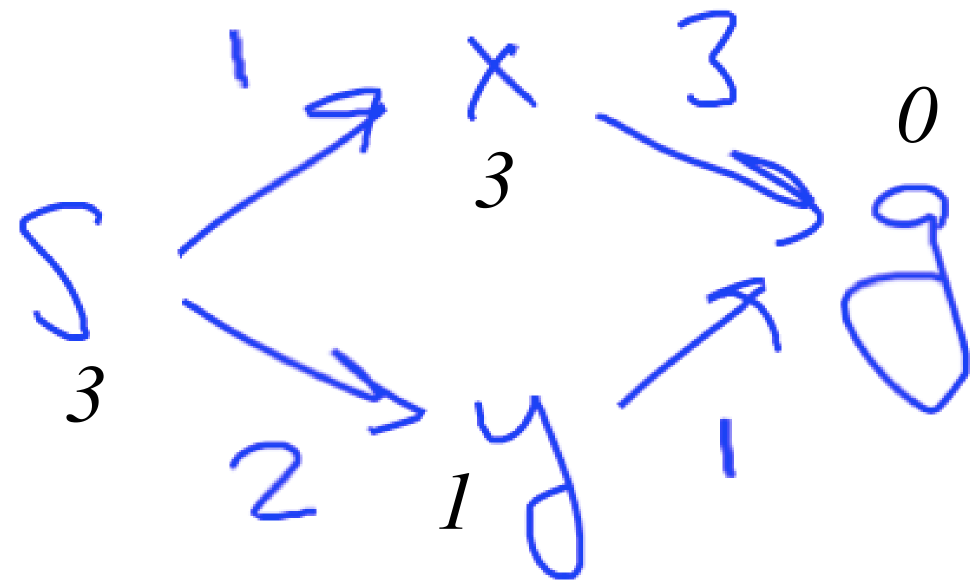
st

$\lambda_s - \lambda_x \le 1$

$\lambda_x - \lambda_y \le 3$

$\lambda_s - \lambda_y \le 2$

$\lambda_y - \lambda_g \le 1$



Any solution which adds a constant to all λs also works; $\lambda_x = 2$ also works

Interpretation

- *Dual variables are prices on nodes: how much does it cost to start there?*
- *Dual constraints are local price constraints: edge* xg *(cost 3) means that node x can't cost more than 3 + price of node g*

More about the dual

Dual dual

- *Take the dual of an LP twice, get the original LP back (called **primal**)*
- *Many LP solvers will give you both primal and dual solutions at the same time for no extra cost*

Recipe

- *If we have an LP in matrix form,*

  maximize c'x subject to

  $Ax \leq b$

  $x \geq 0$

- *Its dual is a similar-looking LP:*

  minimize b'y subject to

  $A'y \geq c$

  $y \geq 0$

$Ax \leq b$ means every component of Ax is $\leq$ corresponding component of b

Recipe with equalities

- *If we have an LP with equalities,*

  maximize $c'x$ *s.t.*

  $$Ax \le b$$
  $$Ex = f$$
  $$x \ge 0$$

- *Its dual has some unrestricted variables:*

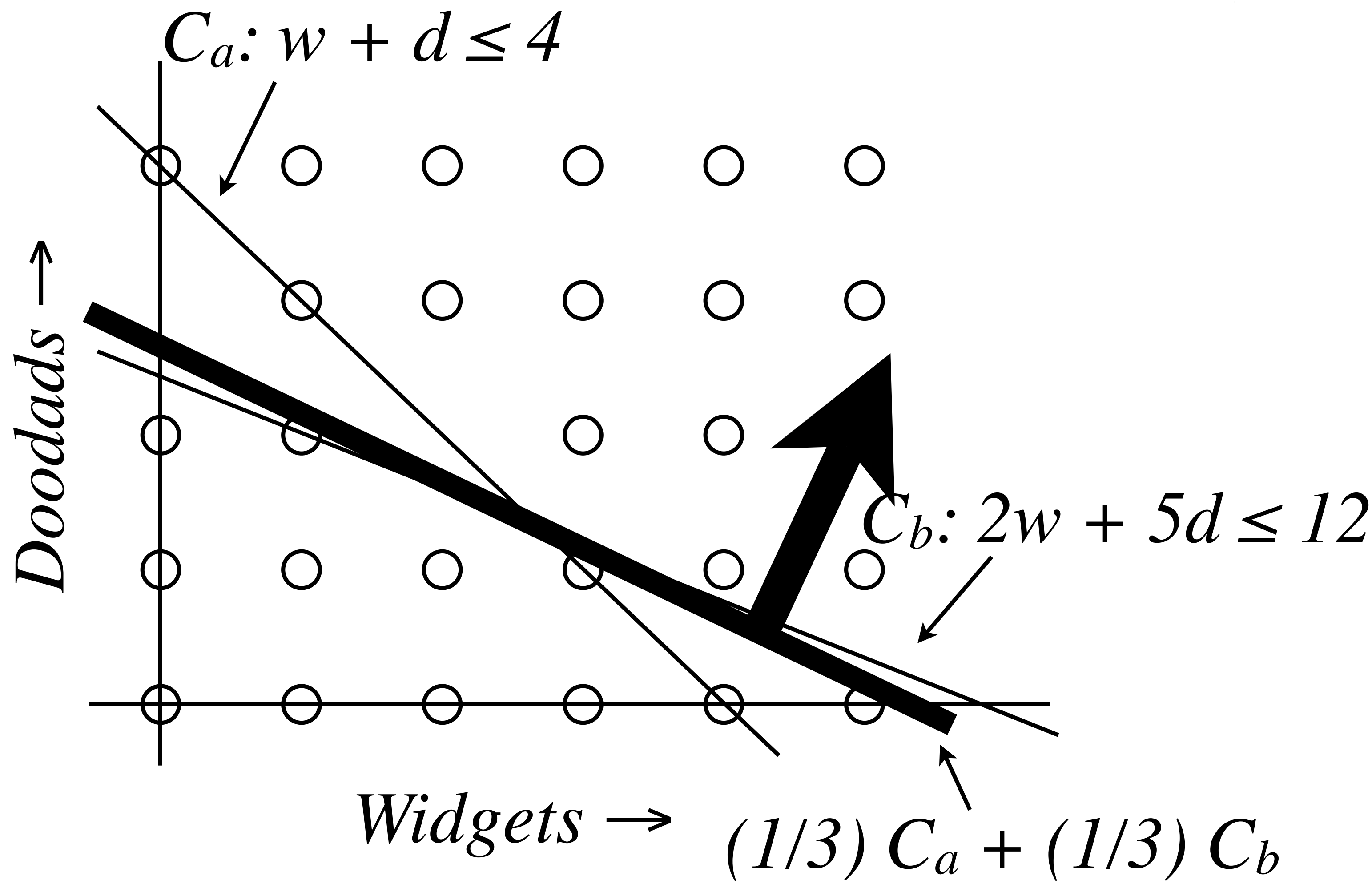
  minimize $b'y + f'z$ *s.t.*

  $$A'y + E'z \ge c$$
  $$y \ge 0$$
  $$z \text{ unrestricted}$$

Interpreting the dual variables

- *The primal variable variables in the factory LP were how many widgets and doodads to produce*
- *We interpreted dual variables as multipliers for primal constraints*

Dual variables as multipliers



Dual variables as prices

- *“Multiplier” interpretation doesn’t give much intuition*
- *It is often possible to interpret dual variables as **prices** for primal constraints*

Dual variables as prices

- *Suppose someone offered us a quantity ε of wood, loosening constraint to*

  $w + d \leq 4 + \varepsilon$

- *How much should we be willing to pay for this wood?*

Dual variables as prices

- *RHS in primal is objective in dual*
- *So, dual constraints stay same, previous solution a = b = 1/3 still dual feasible*
  - *still optimal if* ε *small enough*
- *Bound changes to (4 +* ε*) a + 12 b, difference of* ε ** 1/3*
- *So we should pay up to $1/3 per unit of wood (in small quantities)*