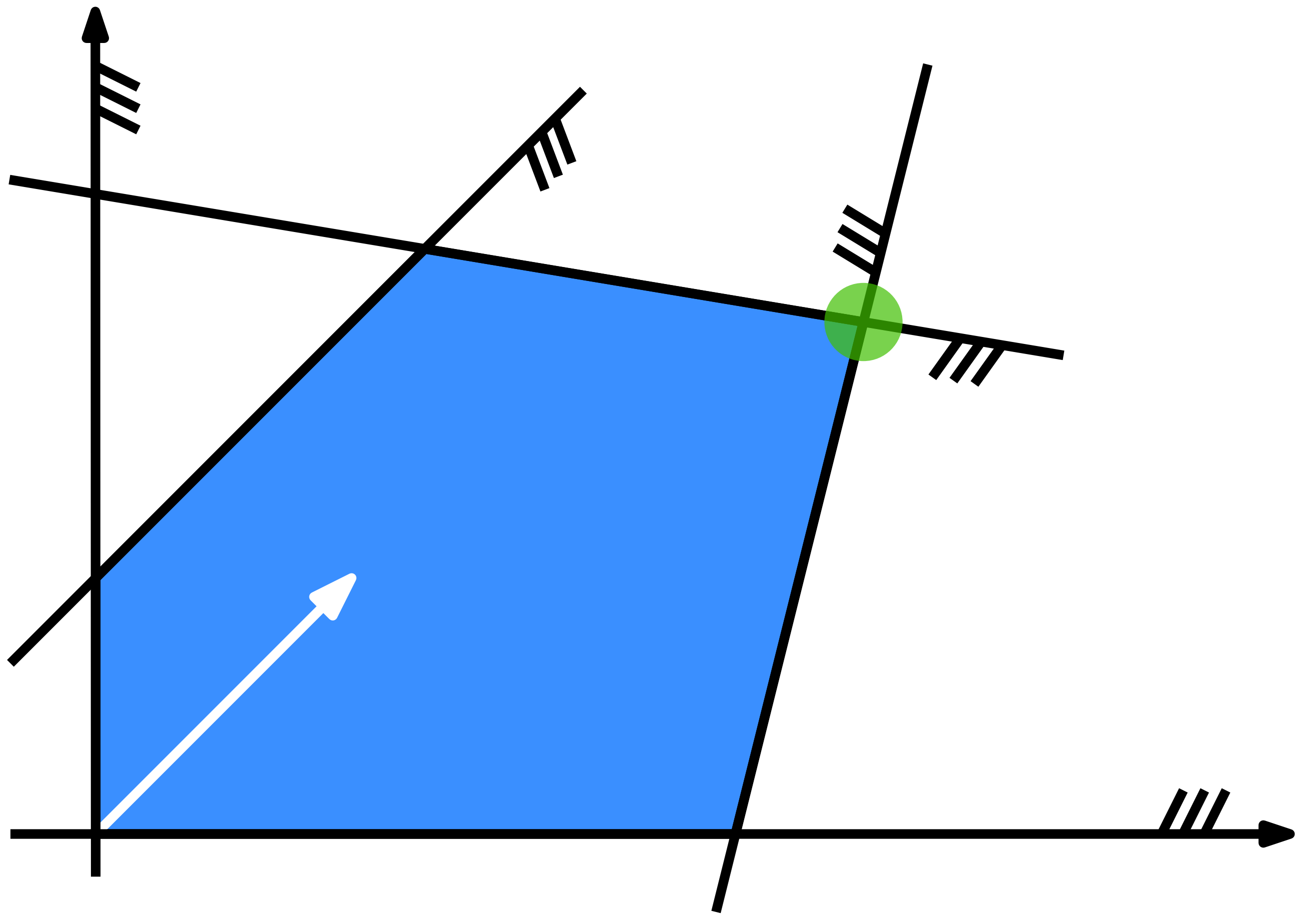
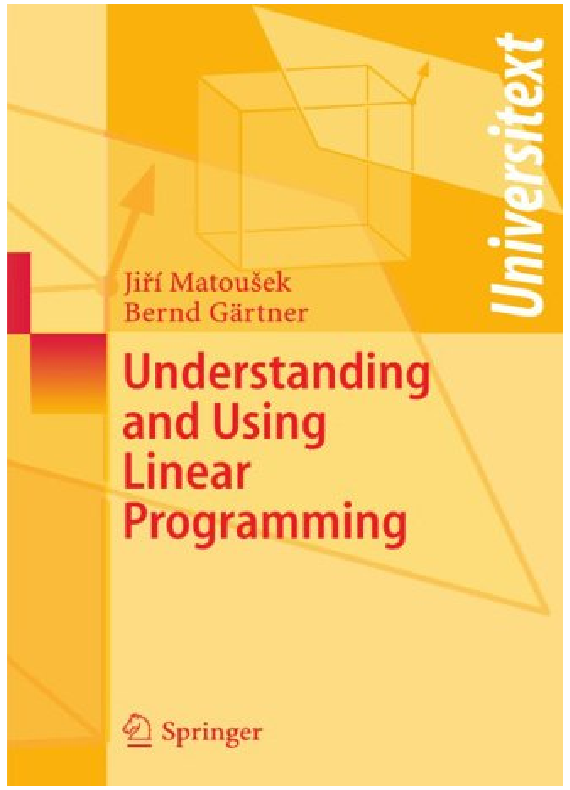
Linear Programming

Introduction and Examples



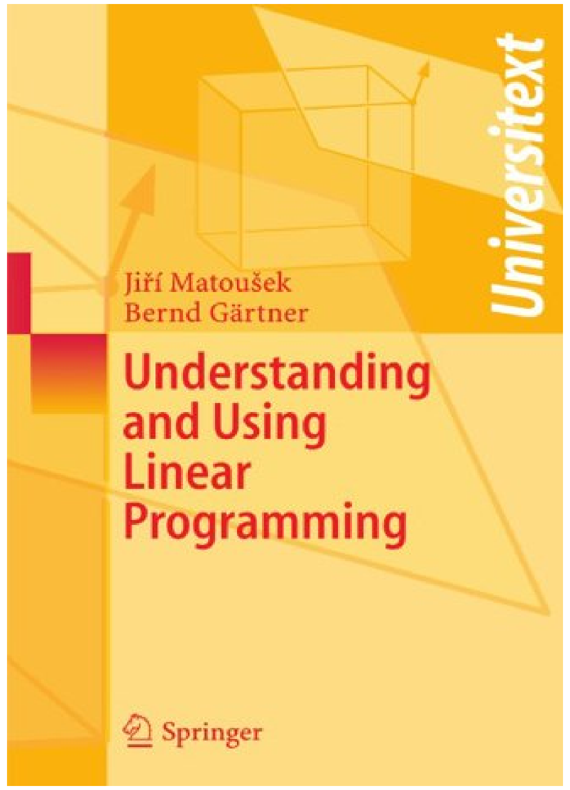
Background

Following "Understanding and Using Linear Programming" by Matoušek and Gärtner



Background

Following "Understanding and Using Linear Programming"
by Matoušek and Gärtner

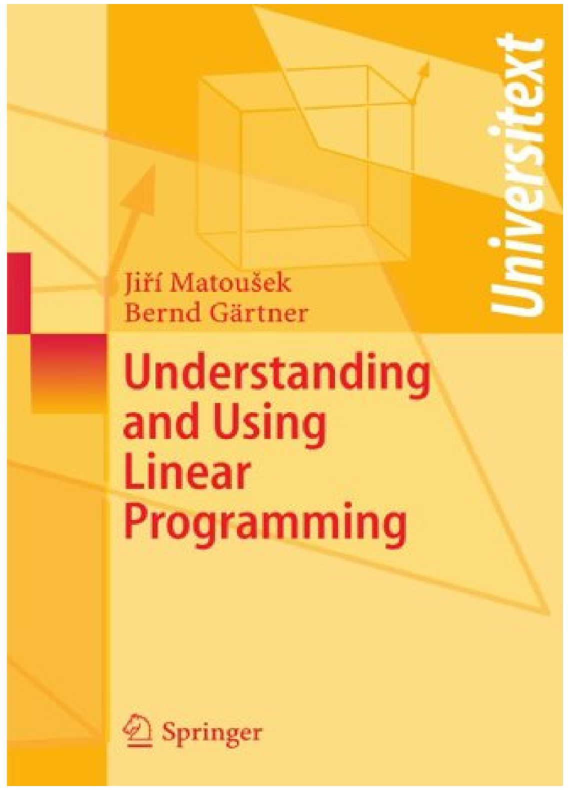


Etymology of Linear Programming:

- 1950's, programming is military term
  $\approx$ planning schedules, supply, deployment
- other name: linear optimization

Background

Following "Understanding and Using Linear Programming" by Matoušek and Gärtner



Etymology of Linear Programming:

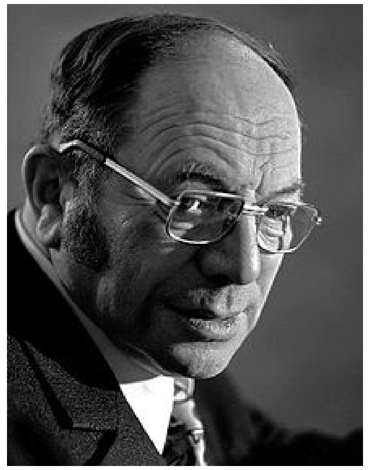
- 1950's, programming is military term
  $\approx$ planning schedules, supply, deployment
- other name: linear optimization

Motivation: many problems can be modelled and solved using linear programming

- in economics and industrial applications
- in computer science, in particular integer linear programs for combinatorial optimization problems
  $\rightarrow$ approximation algorithms (and more)

History

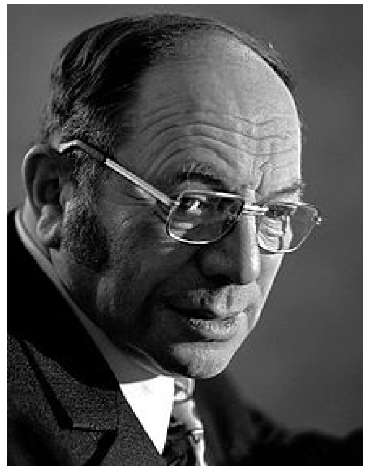
- Leonid Witaljewitsch Kantorowitsch (1939): linear programming for production planning (1975: nobel prize)



Leonid Kantorowitsch
(1912 – 1986)

History

- Leonid Witaljewitsch Kantorowitsch (1939): linear programming for production planning (1975: nobel prize)
- George B. Dantzig (1947): simplex algorithmus (planning problems)
  - continued development: George B. Dantzig, John von Neumann, Oscar Morgenstern, Tjalling Koopmans, . . .

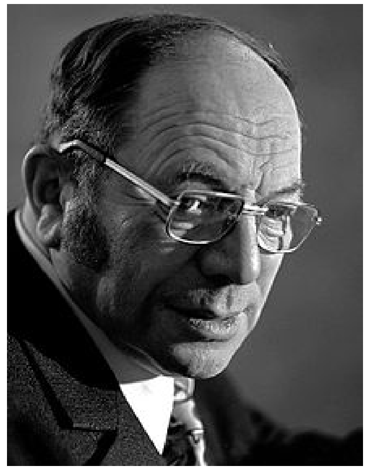


Leonid Kantorowitsch
(1912 – 1986)

George Dantzig
(1914 – 2005)

History

- Leonid Witaljewitsch Kantorowitsch (1939): linear programming for production planning (1975: nobel prize)
- George B. Dantzig (1947): simplex algorithmus (planning problems)
    - continued development: George B. Dantzig, John von Neumann, Oscar Morgenstern, Tjalling Koopmans, . . .
- 1950s: important for oil refineries
- 1970s: more and more industries use linear programming (e. g. airlines)



Leonid Kantorowitsch
(1912 – 1986)

George Dantzig
(1914 – 2005)

Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$

Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

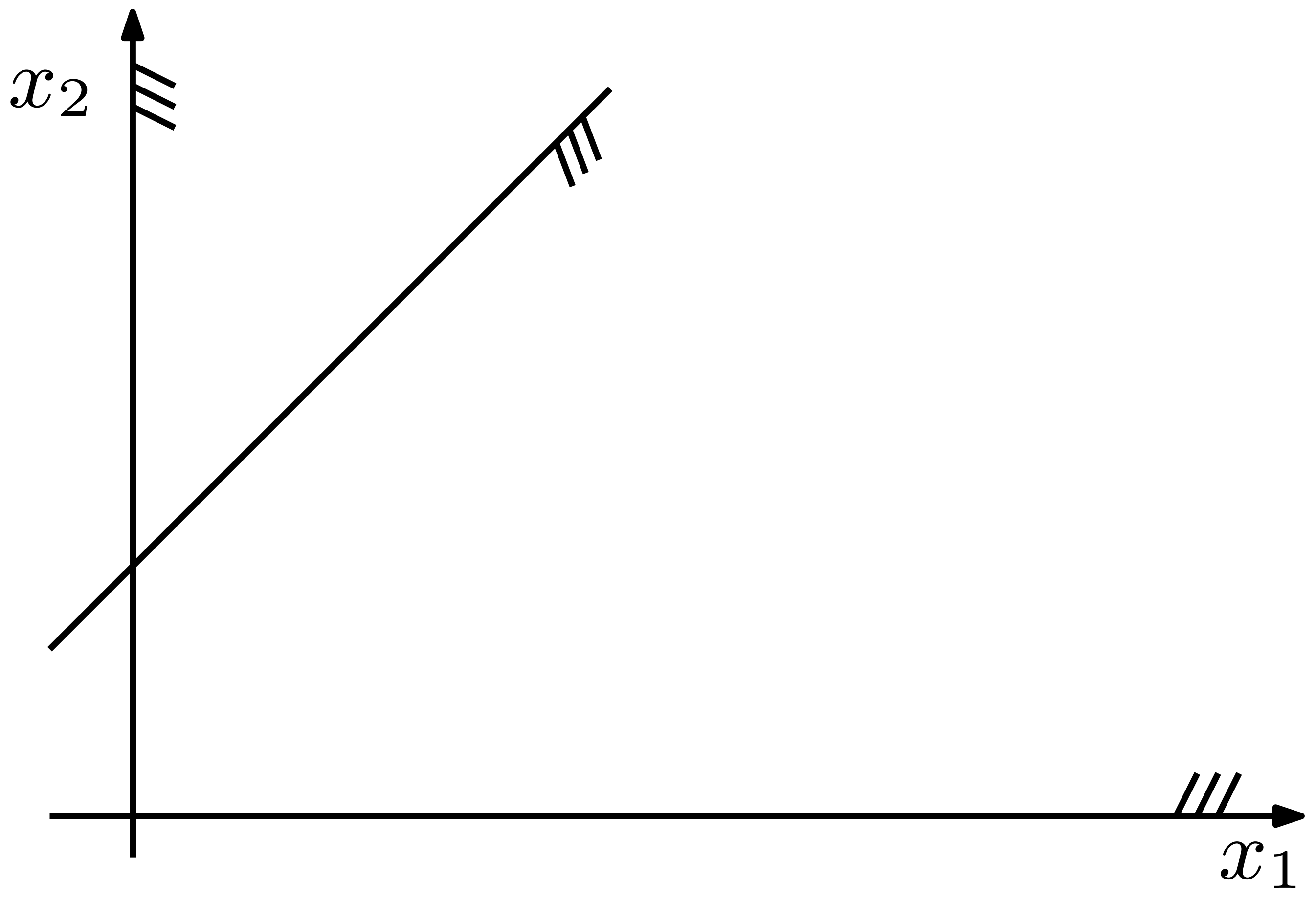
$$
\begin{aligned}
x_1 &\geq 0 \\
x_2 &\geq 0 \\
-x_1 + x_2 &\leq 1 \\
x_1 + 6x_2 &\leq 15 \\
4x_1 - x_2 &\leq 10
\end{aligned}
$$

Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$x_1 \geq 0$$
$$x_2 \geq 0$$
$$-x_1 + x_2 \leq 1$$
$$x_1 + 6x_2 \leq 15$$
$$4x_1 - x_2 \leq 10$$

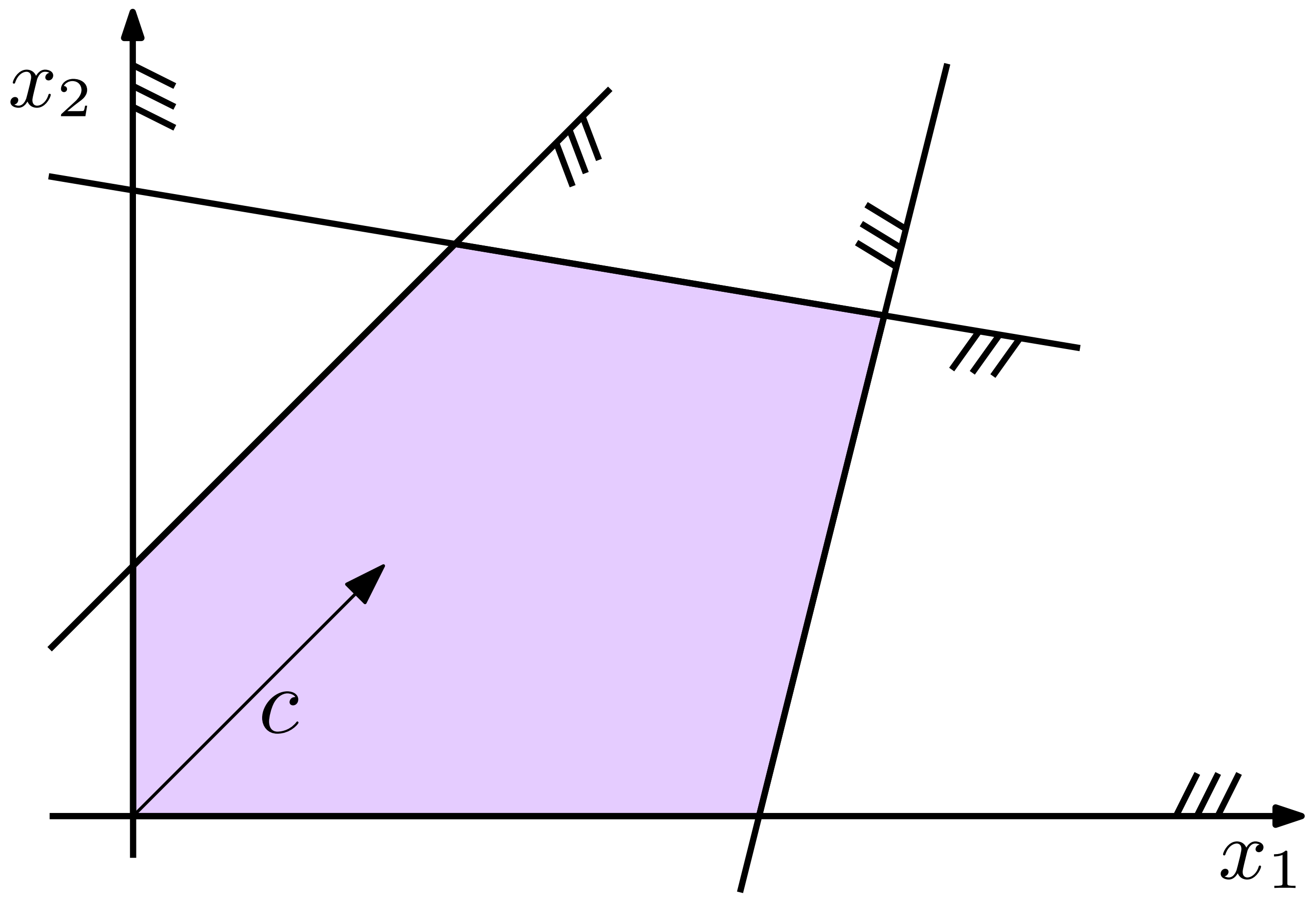


Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$

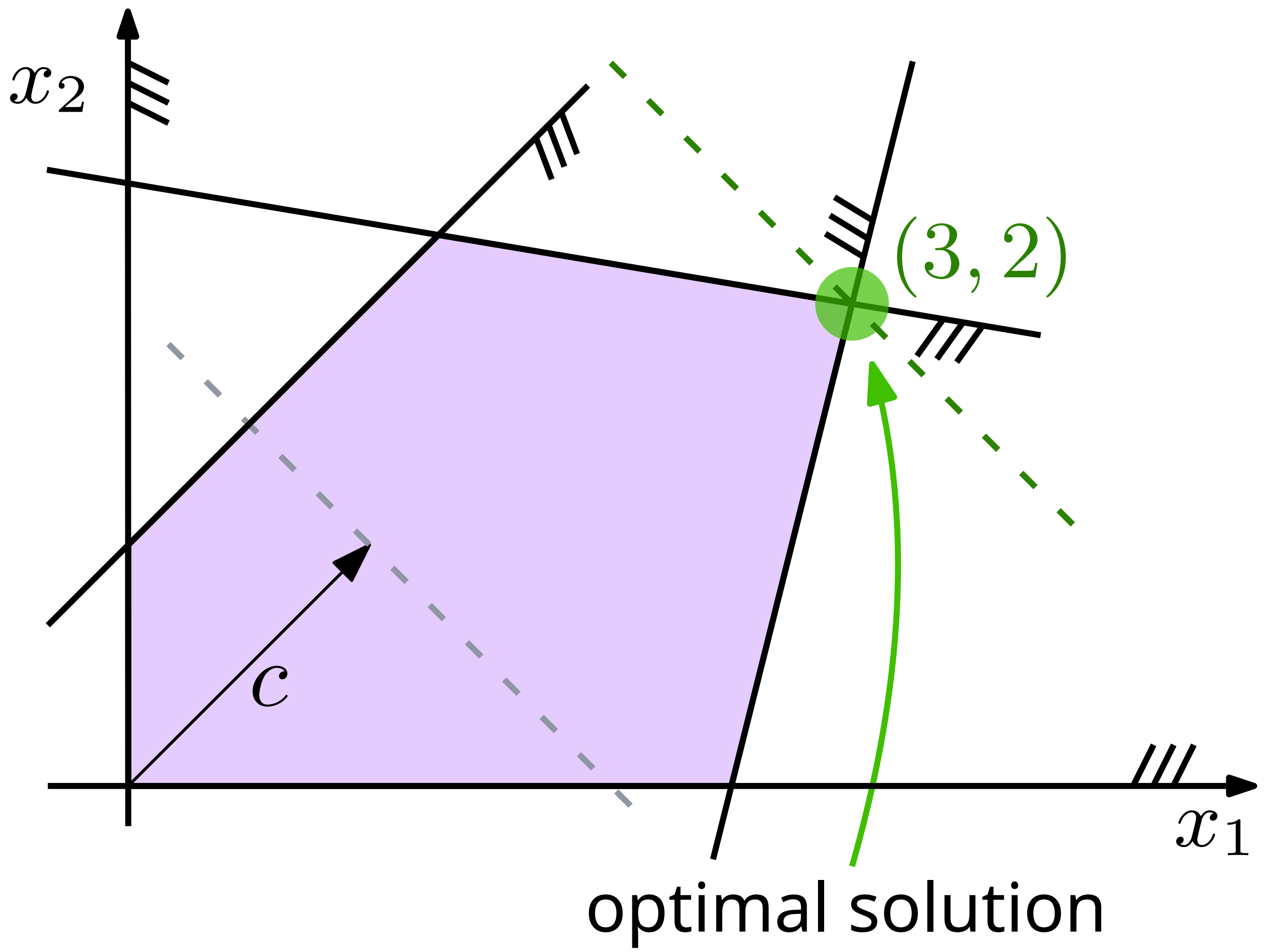


Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$

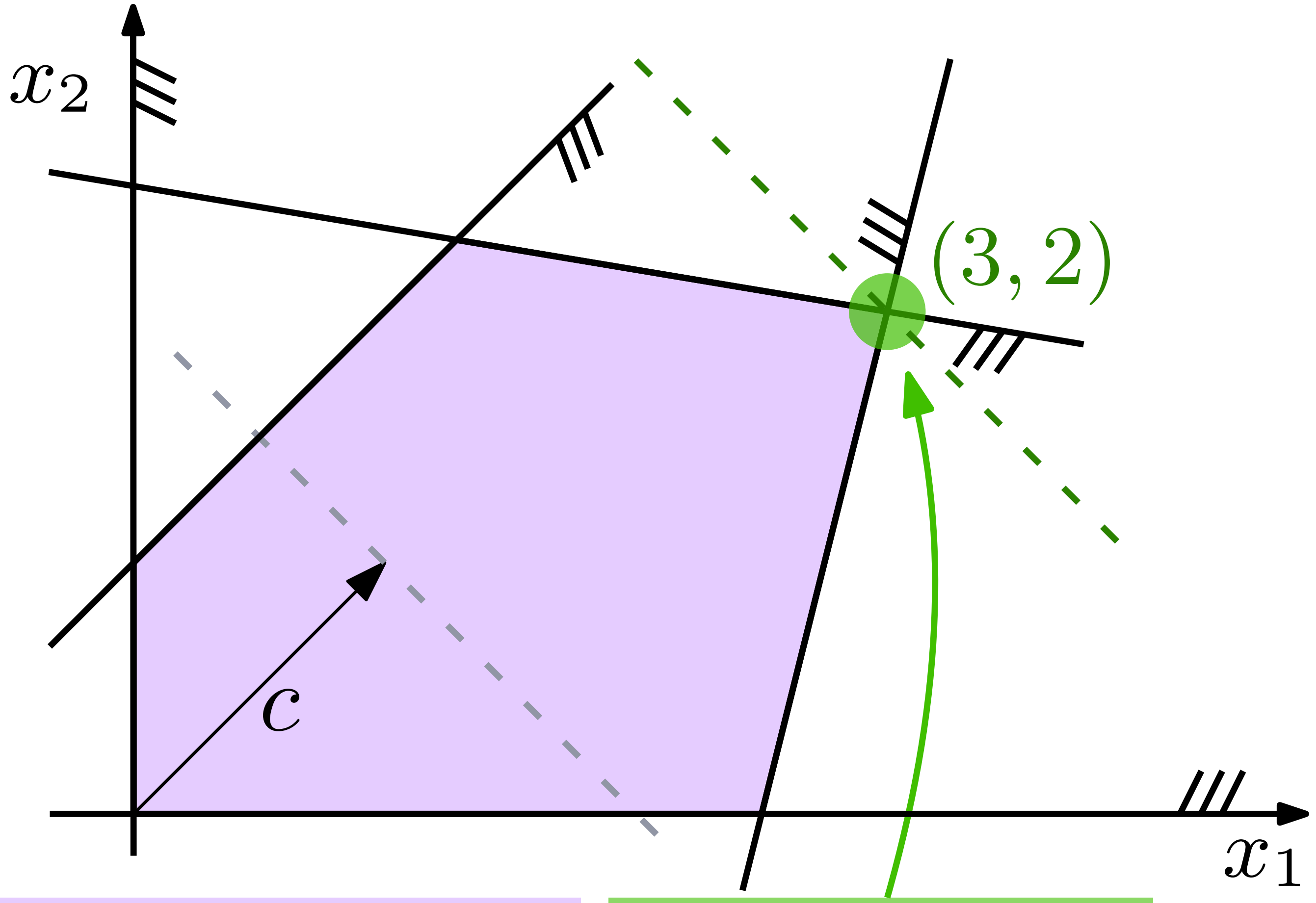


Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$



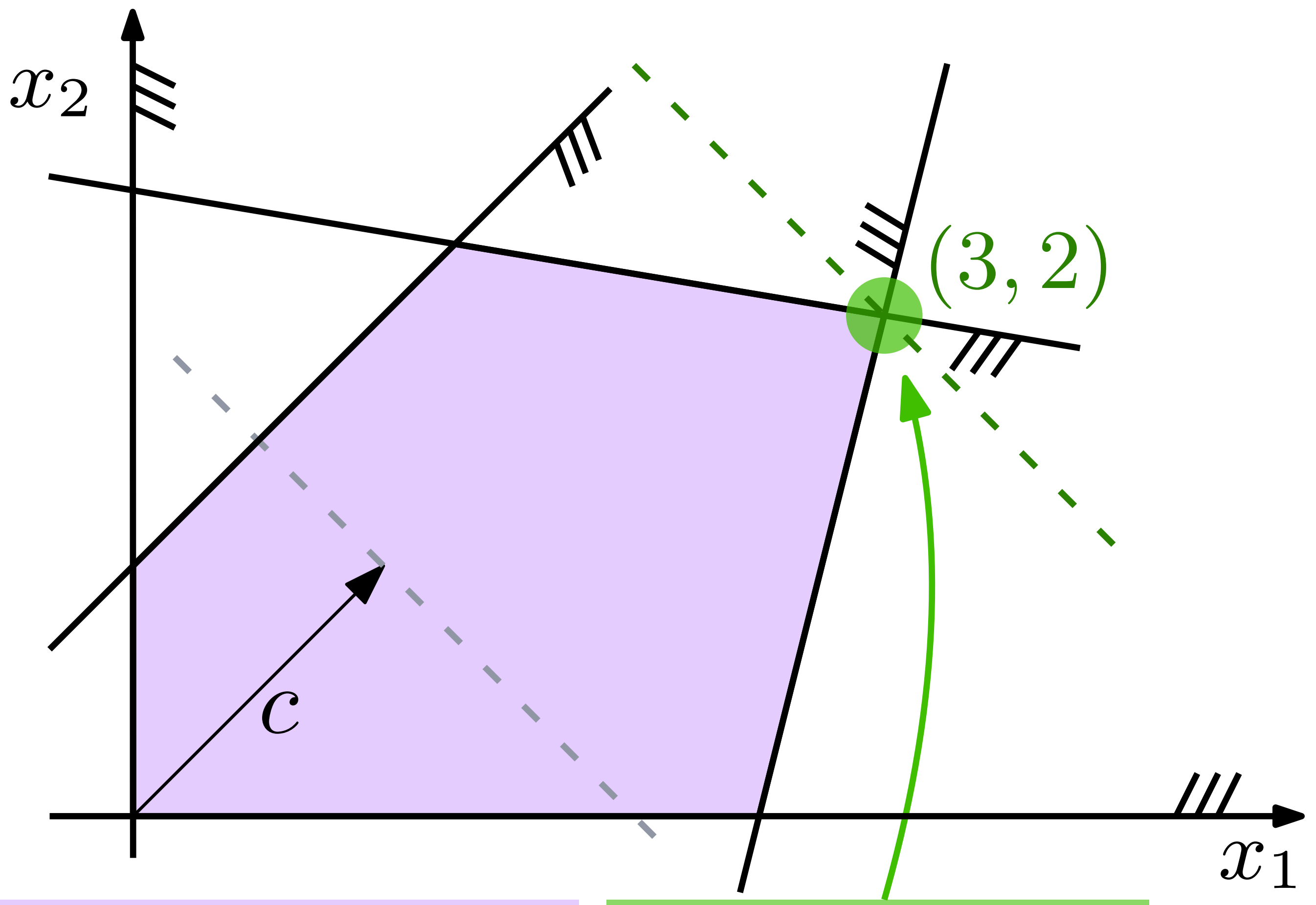
Vocabulary: objective function, constraints, feasible solutions, optimal solution

Introduction to Linear Programming by Example

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$



Vocabulary: objective function, constraints, feasible solutions, optimal solution

More generally:

$$\begin{aligned} &\text{maximize} && c^T x \\ &\text{subject to} && Ax \leq b \end{aligned}$$

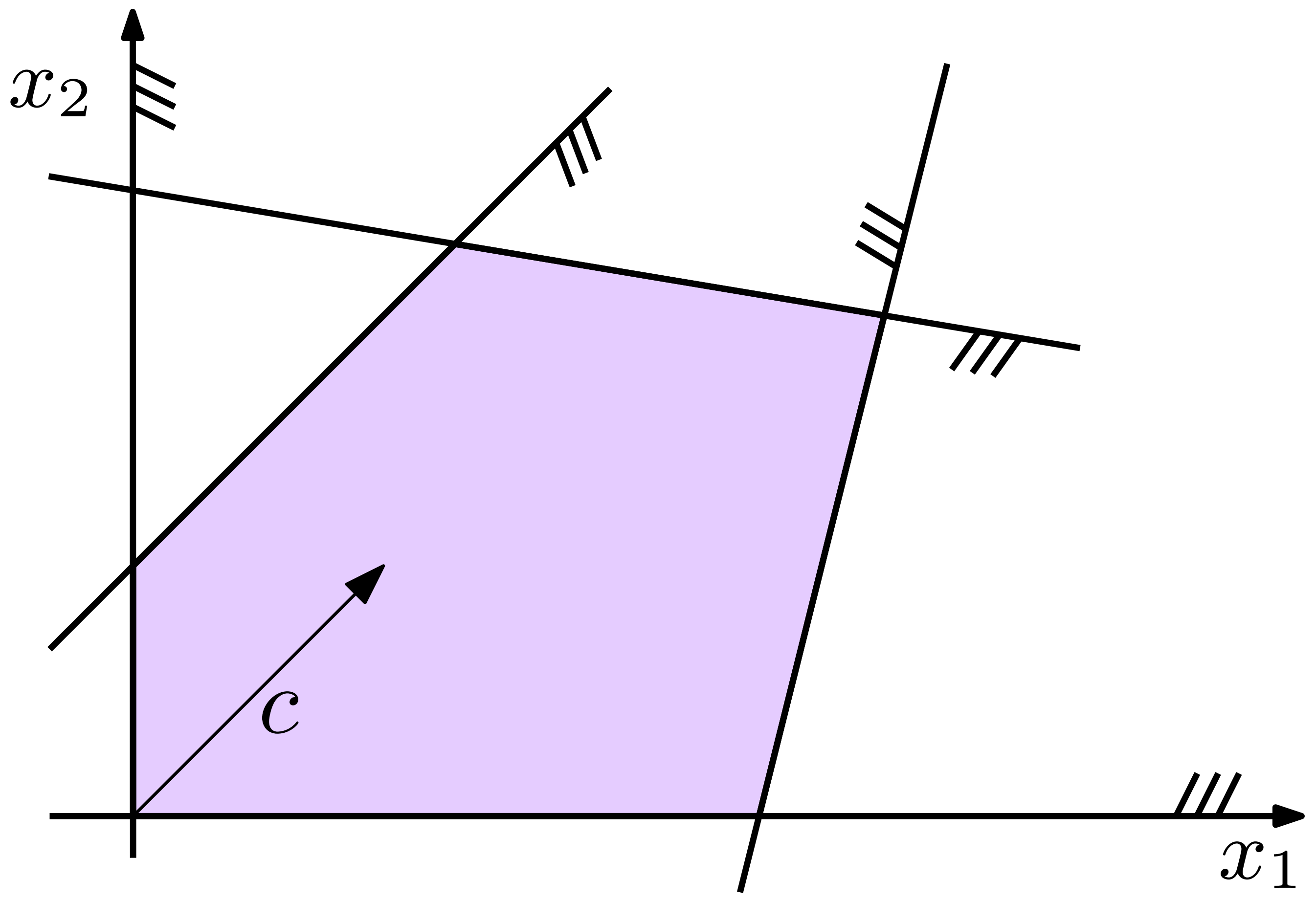
Here $x \in \mathbb{R}^n$ encodes the variables
and $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$ are given.

Introduction to Linear Programming

maximize $\cancel{x_1 + x_2}$ $x_1/6 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$

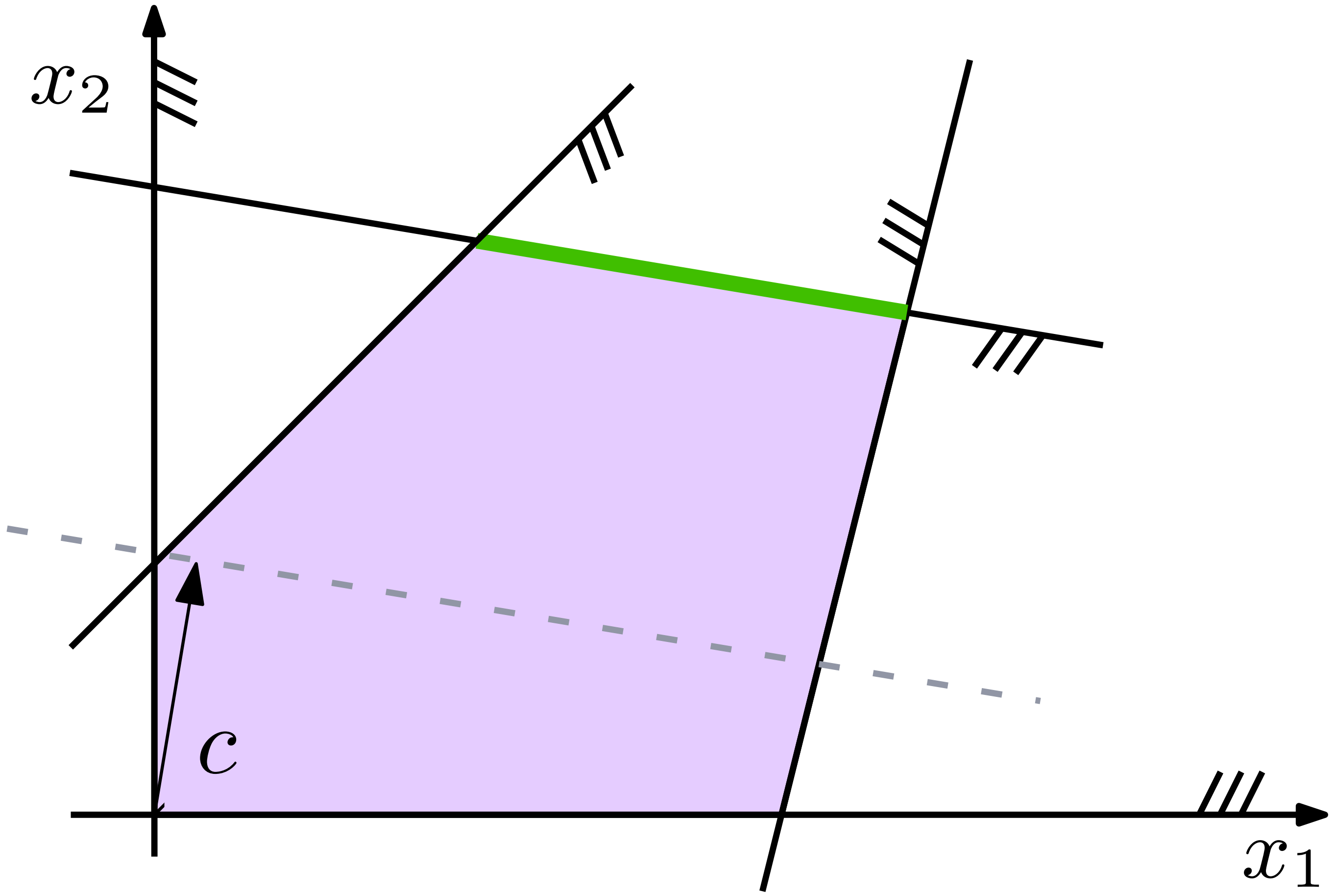


What if we change c to $(\frac{1}{6}, 1)$?

Introduction to Linear Programming

maximize $\cancel{x_1 + x_2}$ $x_1/6 + x_2$
for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$



What if we change c to $(\frac{1}{6}, 1)$?

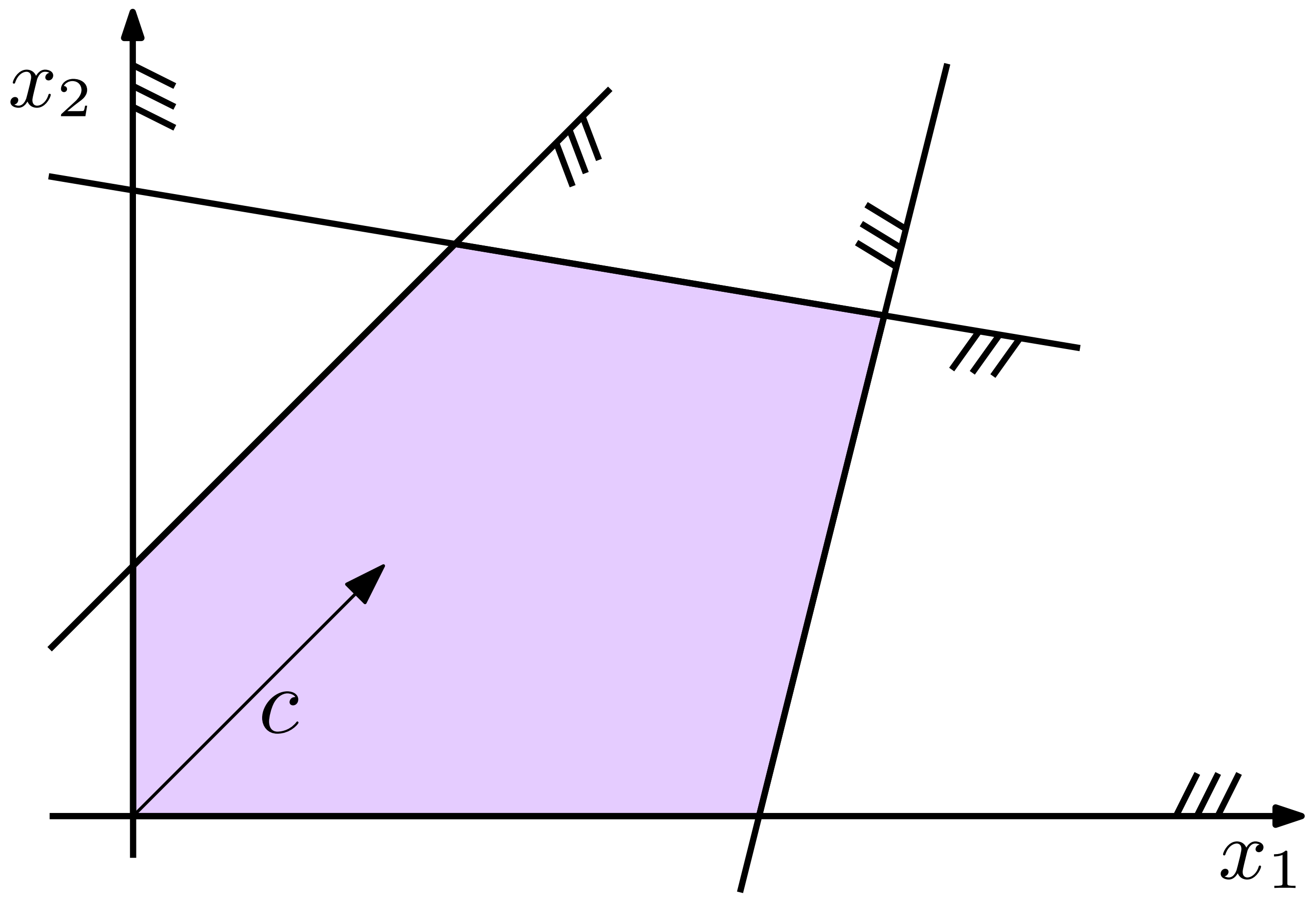
Line segment as optimal solution.

Introduction to Linear Programming

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ x_1 + 6x_2 &\cancel{\leq} 15 \\ 4x_1 - x_2 &\cancel{\leq} 10 \\ &\geq \end{aligned}$$



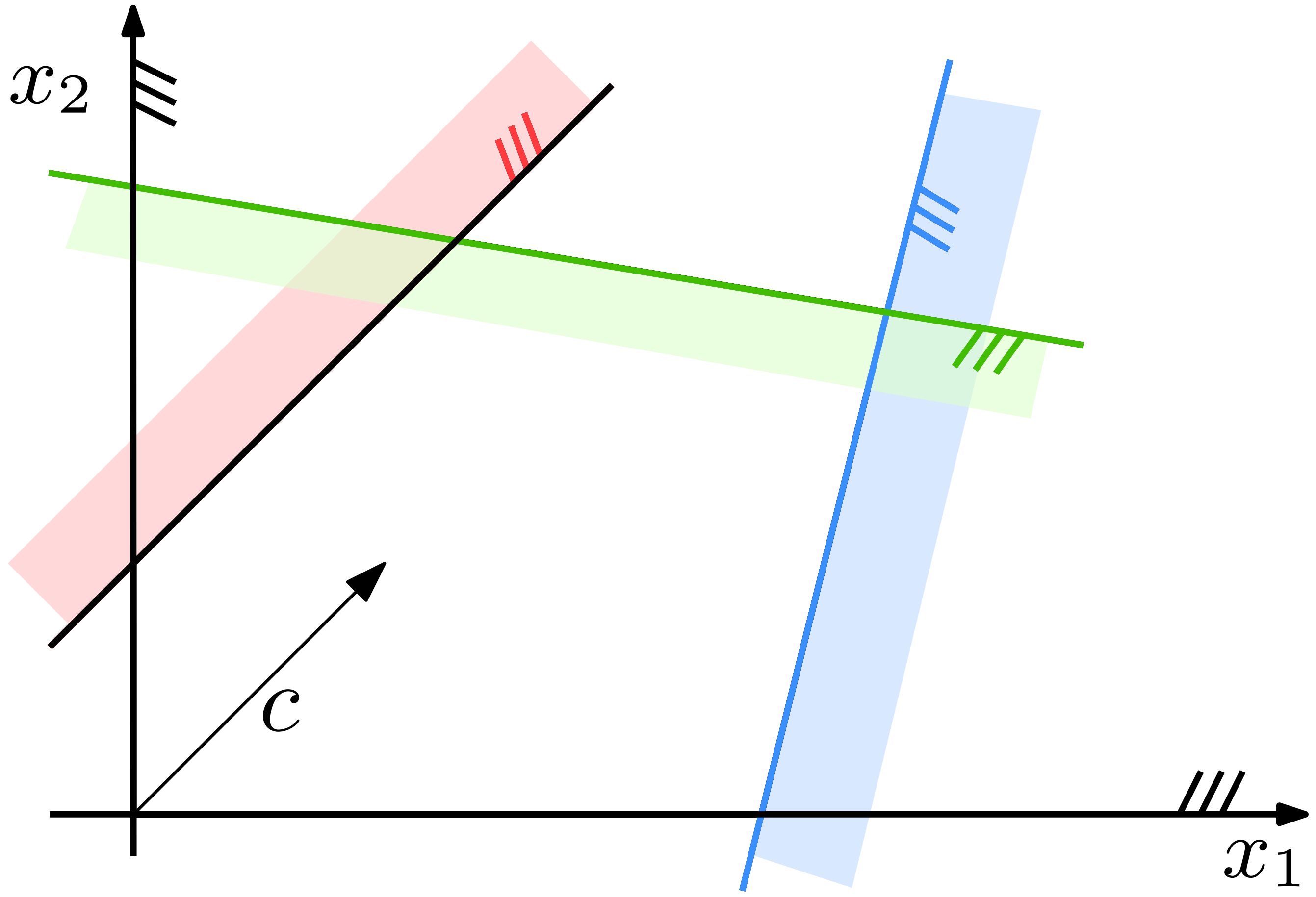
What if we change to $-x_1 + x_2 \geq 1$ and $4x_1 - x_2 \geq 10$?

Introduction to Linear Programming

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$
\begin{aligned}
x_1 &\geq 0 \\
x_2 &\geq 0 \\
-x_1 + x_2 &\leq 1 \\
x_1 + 6x_2 &\cancel{\leq} 15 \\
4x_1 - x_2 &\cancel{\leq} 10 \\
&\geq
\end{aligned}
$$



What if we change to $-x_1 + x_2 \geq 1$ and $4x_1 - x_2 \geq 10$?

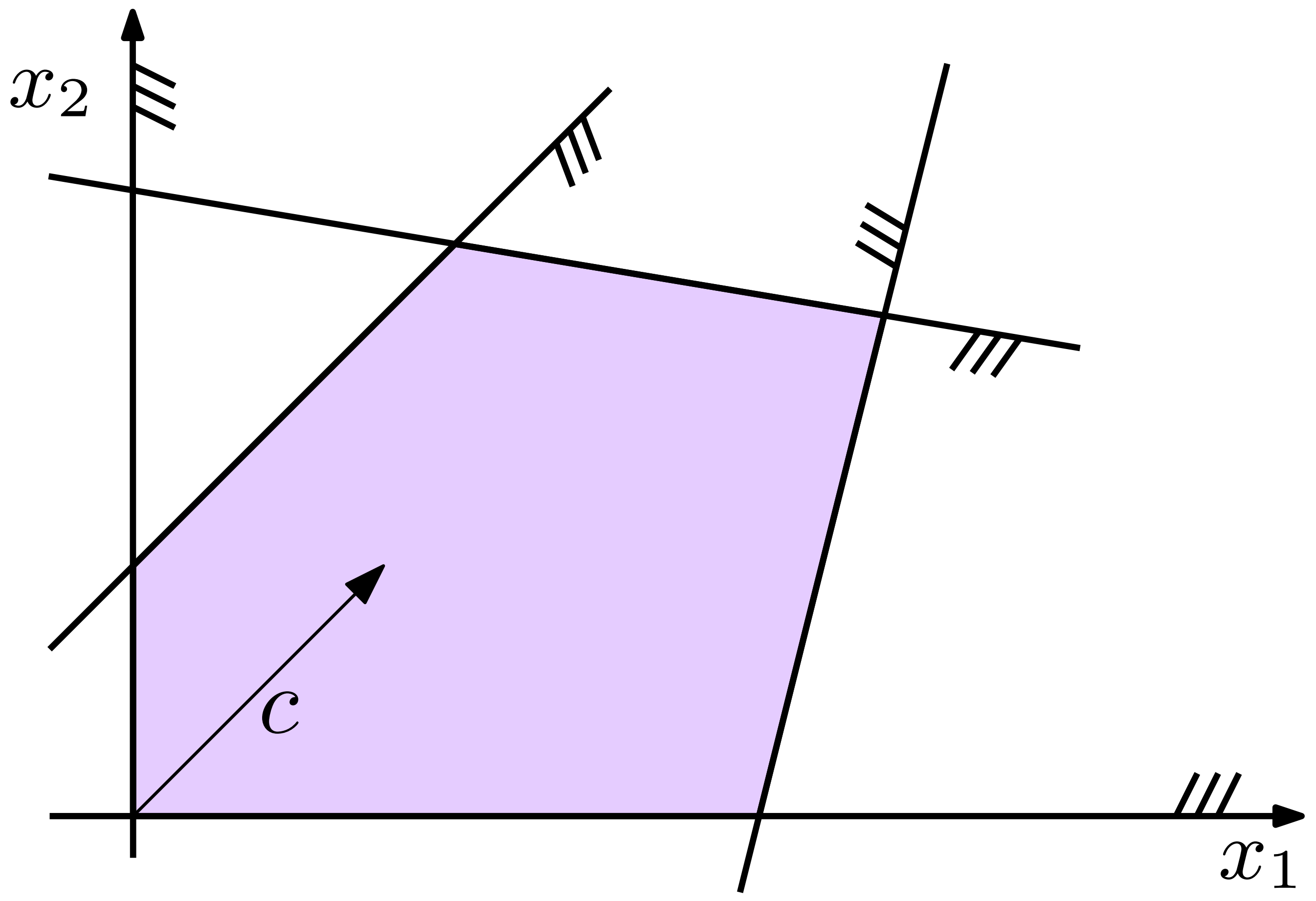
Infeasible.

Introduction to Linear Programming

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$
\begin{aligned}
x_1 &\geq 0 \\
x_2 &\geq 0 \\
-x_1 + x_2 &\leq 1 \\
\cancel{x_1 + 6x_2} &\cancel{\leq 15} \\
\cancel{4x_1 - x_2} &\cancel{\leq 10}
\end{aligned}
$$



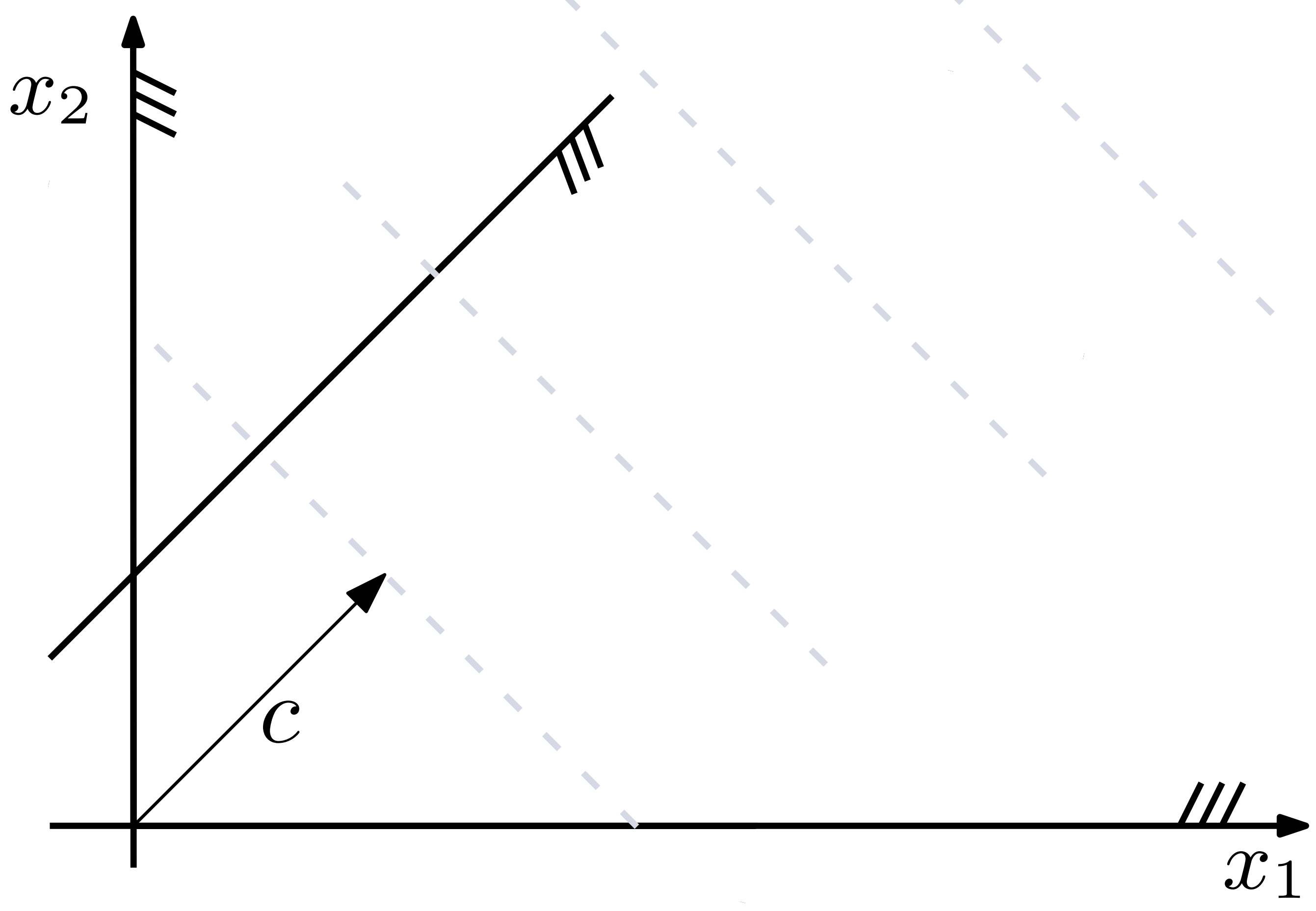
What if we remove the last two constraints?

Introduction to Linear Programming

maximize $x_1 + x_2$

for $x_1, x_2 \in \mathbb{R}$ satisfying

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ -x_1 + x_2 &\leq 1 \\ \cancel{x_1 + 6x_2} &\cancel{\leq 15} \\ \cancel{4x_1 - x_2} &\cancel{\leq 10} \end{aligned}$$



What if we remove the last two constraints?

Unbounded.

Efficiency of Linear Programming

Linear programs are efficiently solvable both

- in practice
  (good software, thousands of variables and constraints)
  simplex method: worst-case exponential time, but typically fast
- in theory
  (algorithms bounded in time by polynomial functions of inputs)
  ellipsoid method: polynomial time

Efficiency of Linear Programming

Linear programs are efficiently solvable both

- in practice
  (good software, thousands of variables and constraints)
  simplex method: worst-case exponential time, but typically fast
- in theory
  (algorithms bounded in time by polynomial functions of inputs)
  ellipsoid method: polynomial time

Note: Algorithms good for (i) are not the same as those for (ii)!

History of Algorithms for Linear Programming

simplex algorithm (Dantzig 1947):

- most commonly used algorithm
- in practice very fast
- worst-case exponential time

History of Algorithms for Linear Programming

simplex algorithm (Dantzig 1947):

- most commonly used algorithm
- in practice very fast
- worst-case exponential time

first polynomial-time algorithms:

- Leonid Khachiyan (1979): ellipsoid method
- Narendra Karmarkar (1984): interior point method
- numerically instable, difficult to implement

History of Algorithms for Linear Programming

simplex algorithm (Dantzig 1947):

- most commonly used algorithm
- in practice very fast
- worst-case exponential time

first polynomial-time algorithms:

- Leonid Khachiyan (1979): ellipsoid method
- Narendra Karmarkar (1984): interior point method
- numerically instable, difficult to implement

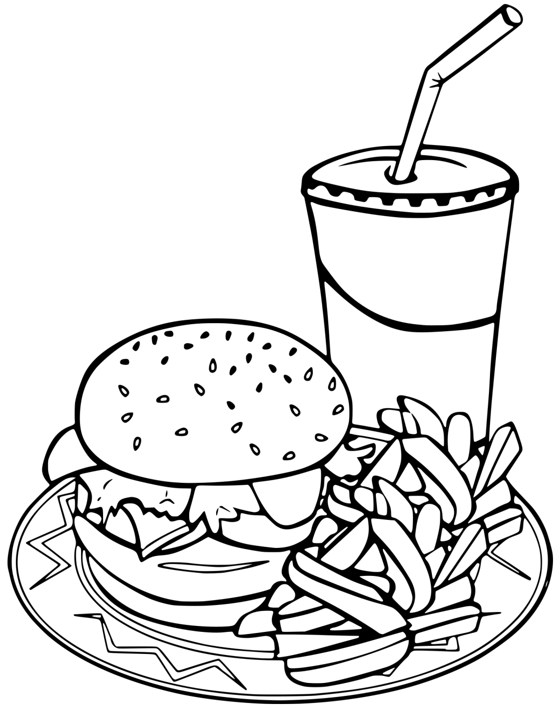
practically efficient software:

- starting 1990s: Robert E. Bixby: fast, stable simplex code (CPLEX), can solve very large LPs. Today: CPLEX, Gurobi and other solvers

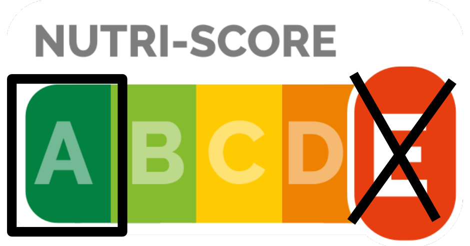
Examples

How to model problems as LPs

Example: Optimized Diet

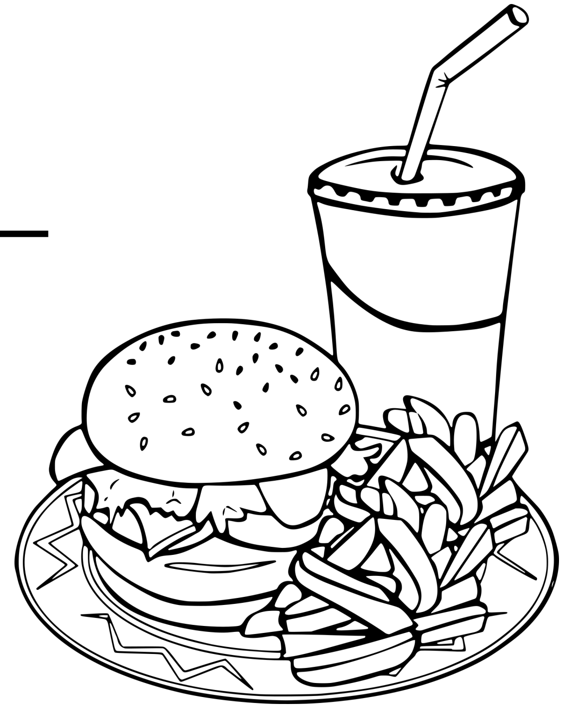


A restaurant needs side dishes to fulfill minimimum nutritional value of a meal.
They want to do so using carrots, cabbage, and pickles.
They aim to be as cheap as possible!

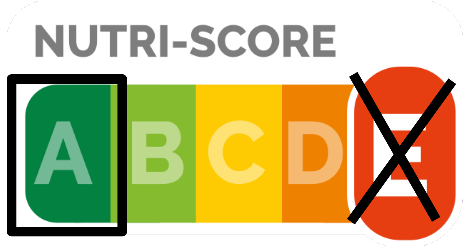


Example: Optimized Diet

Nutrition \ Food	Carrot, Raw	White Cabbage, Raw	Cucumber, Pickled	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–



A restaurant needs side dishes to fulfill minimimum nutritional value of a meal.
They want to do so using carrots, cabbage, and pickles.
They aim to be as cheap as possible!



Example: Optimized Diet

Nutrition \ Food	Carrot, Raw x_1	White Cabbage, Raw x_2	Cucumber, Pickled x_3	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–

minimize $\quad 0.75x_1 + 0.5x_2 + 0.15x_3$

Example: Optimized Diet

Food / Nutrition	Carrot, Raw x_1	White Cabbage, Raw x_2	Cucumber, Pickled x_3	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–

$$
\begin{aligned}
\text{minimize} \quad & 0.75x_1 + 0.5x_2 + 0.15x_3 \\
\text{subject to} \quad & 35x_1 + 0.5x_2 + 0.5x_3 \geq 0. \\
& 60x_1 + 300x_2 + 10x_3 \geq 1 \\
& 30x_1 + 20x_2 + 10x_3 \geq 4 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}
$$

Example: Optimized Diet

Food Nutrition	x_1 Carrot, Raw	x_2 White Cabbage, Raw	x_3 Cucumber, Pickled	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–

$$
\begin{array}{ll}
\text{minimize} & 0.75x_1 + 0.5x_2 + 0.15x_3 \\
\text{subject to} & 35x_1 + 0.5x_2 + 0.5x_3 \geq 0. \\
& 60x_1 + 300x_2 + 10x_3 \geq 1. \\
& 30x_1 + 20x_2 + 10x_3 \geq 4 \\
& x_1, x_2, x_3 \geq 0
\end{array}
$$

Optimal solution:

$x_1 = 9.5g, x_2 = 38g, x_3 = 295g$

Cost: 0.07€

Example: Optimized Diet

Food / Nutrition	Carrot, Raw (x_1)	White Cabbage, Raw (x_2)	Cucumber, Pickled (x_3)	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–

$$
\begin{array}{ll}
\text{minimize} & 0.75x_1 + 0.5x_2 + 0.15x_3 \\
\text{subject to} & 35x_1 + 0.5x_2 + 0.5x_3 \geq 0. \\
& 60x_1 + 300x_2 + 10x_3 \geq 1. \\
& 30x_1 + 20x_2 + 10x_3 \geq 4 \\
& x_1, x_2, x_3 \geq 0
\end{array}
$$

Optimal solution:

$x_1 = 9.5g, x_2 = 38g, x_3 = 295g$

Cost: 0.07€

Is this a practical solution?

Moral: Modeling is hard!

Example: Optimized Diet

Nutrition \ Food	Carrot, Raw	White Cabbage, Raw	Cucumber, Pickled	Required per dish
Vitamin A [mg/kg]	35	0.5	0.5	0.5 mg
Vitamin C [mg/kg]	60	300	10	15 mg
Dietary Fiber [g/kg]	30	20	10	4 g
price [€/kg]	0.75	0.5	0.15	–

Around 5:00 PM, Anne called, "Nu, it's five and you haven't called. What should I be cooking?" I replied that she didn't really want to know. I then read off the amounts of foods in the optimal diet. Her reaction: "The diet is a bit weird but conceivable. Is that it?"

"Not exactly," I replied, "AND 500 gallons of vinegar." She thought it funny and laughed.

Optimal solution:

$x_1 = 9.5g, x_2 = 38g, x_3 = 295g$

Cost: 0.07€

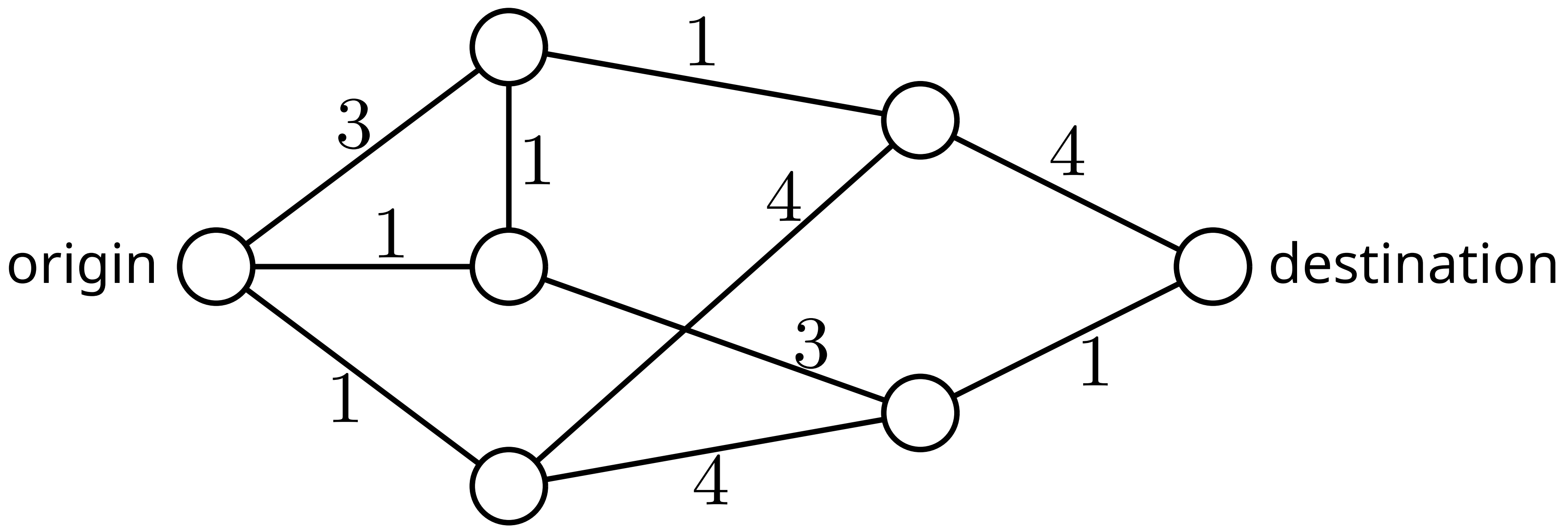
Is this a practical solution?

Moral: Modeling is hard!

from: Dantzig, George B. "The diet problem." Interfaces 20.4 (1990): 43-47.

Example: Flow in a Network

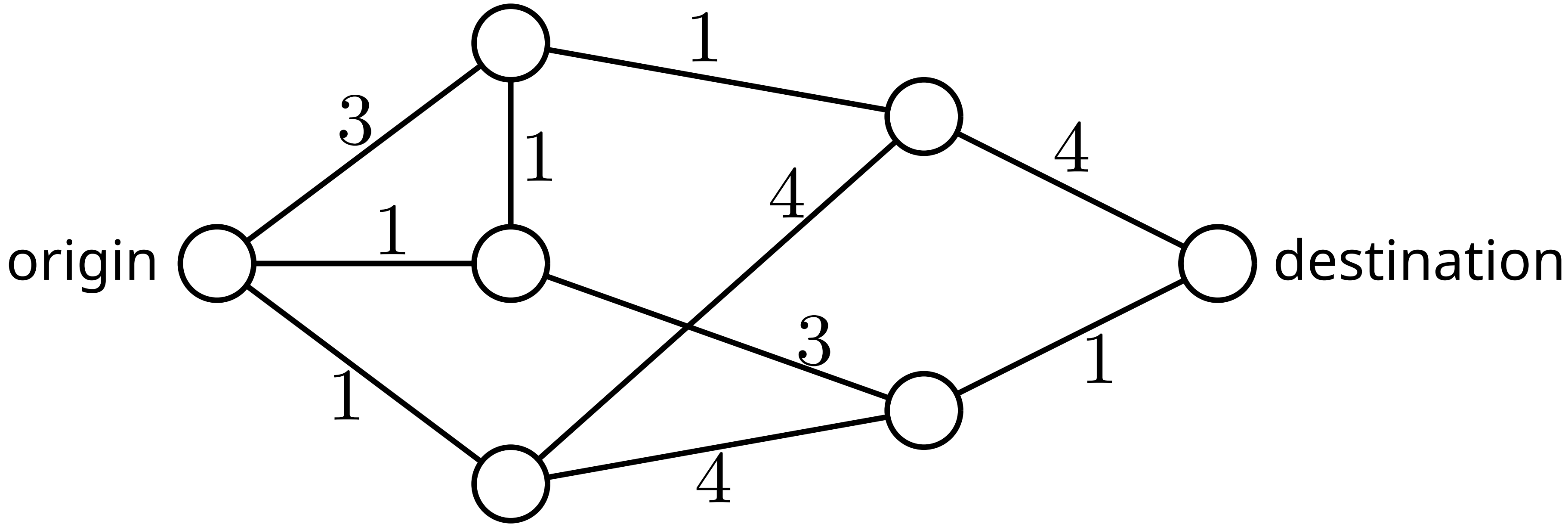
How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

Example: Flow in a Network

How to send as much data as possible over a local network?

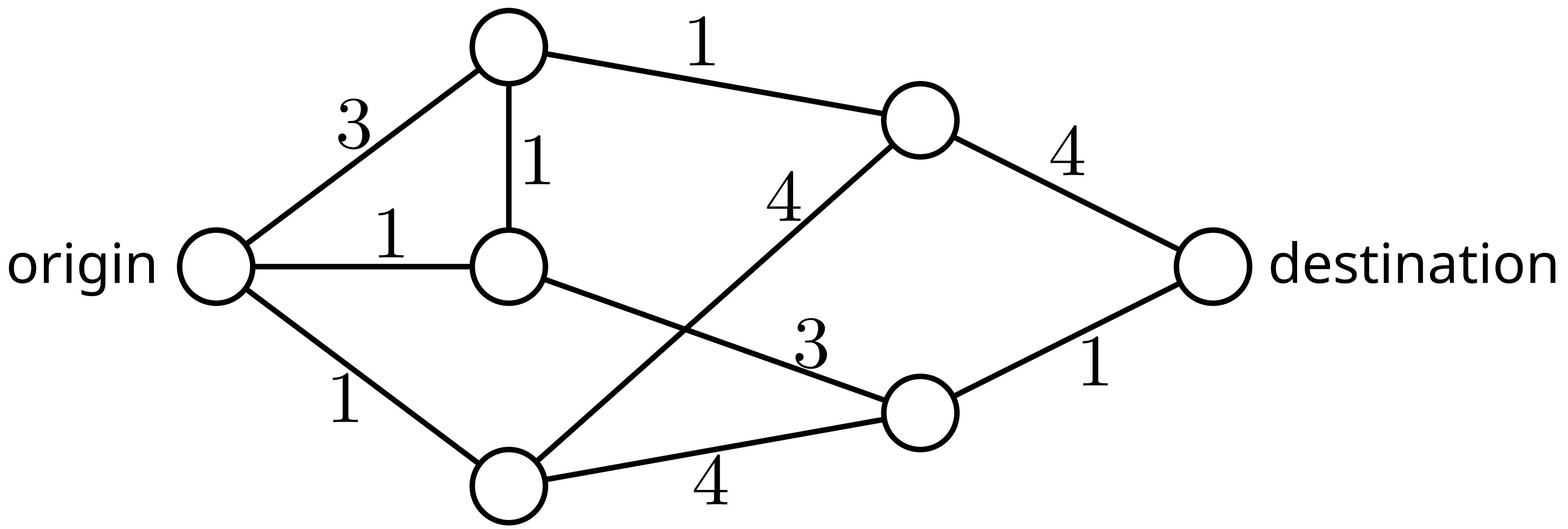


nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

Example: Flow in a Network

How to send as much data as possible over a local network?



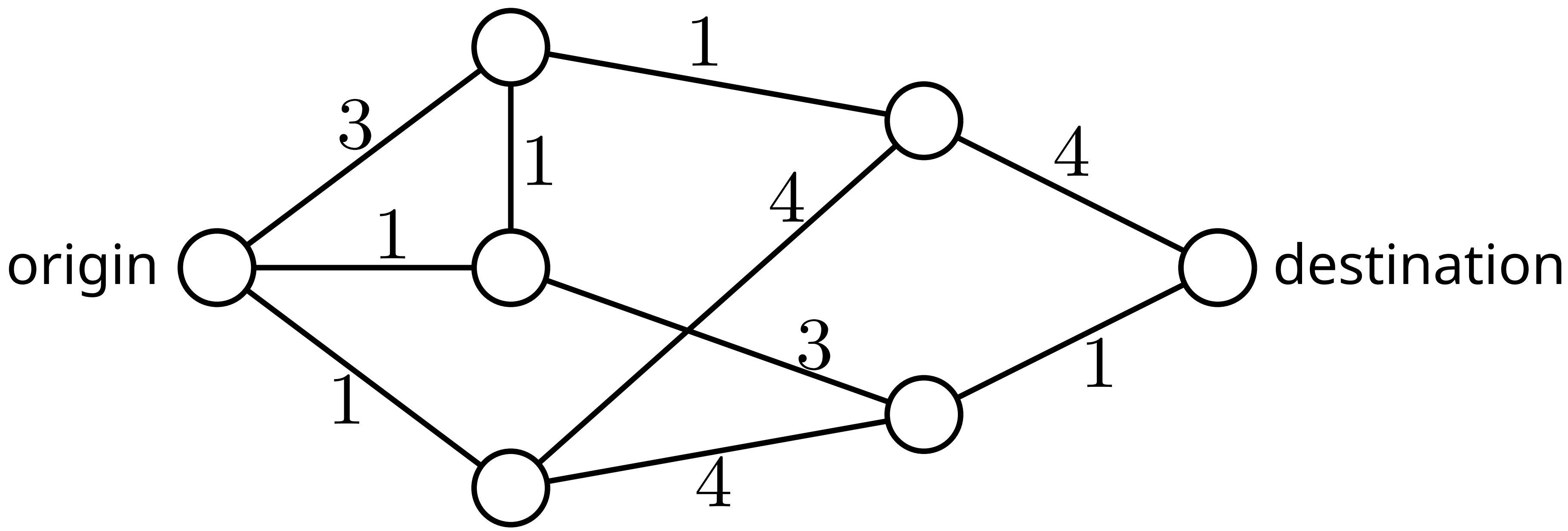
nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

Can we formulate this as LP?

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

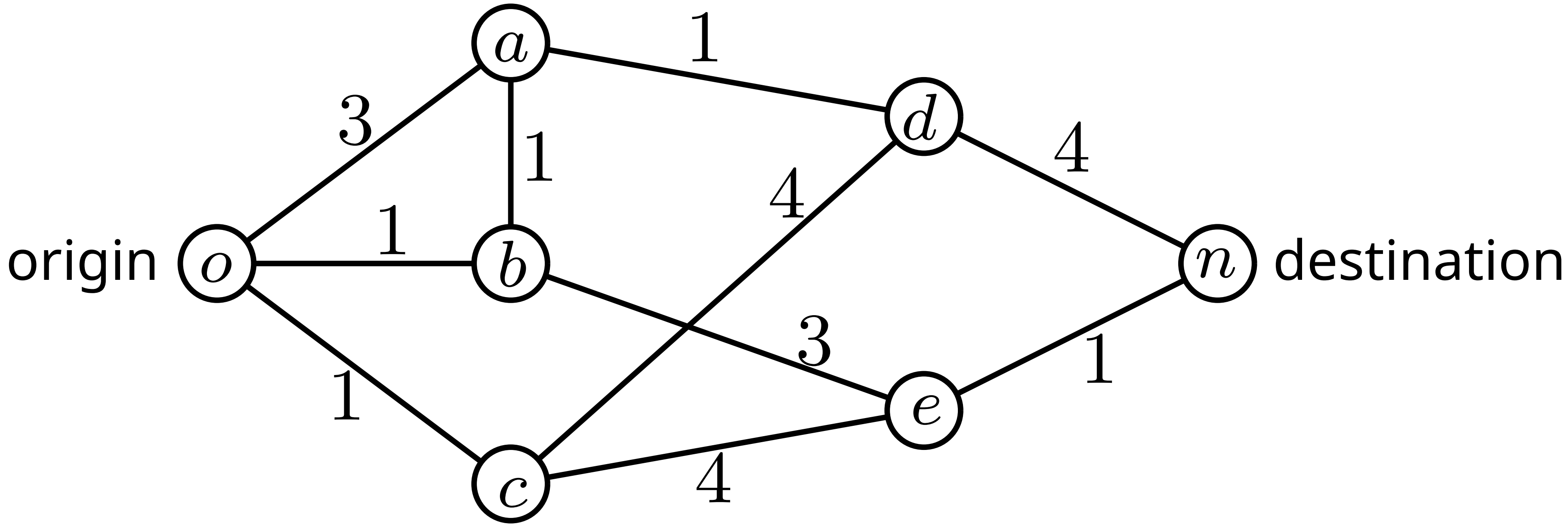
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

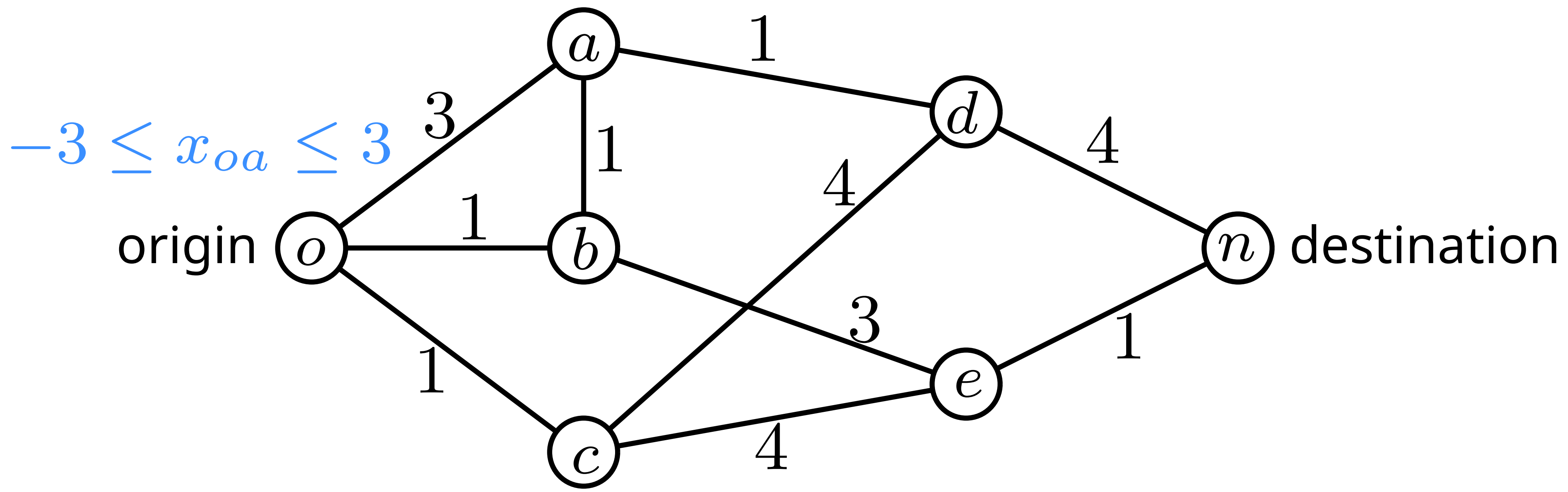
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

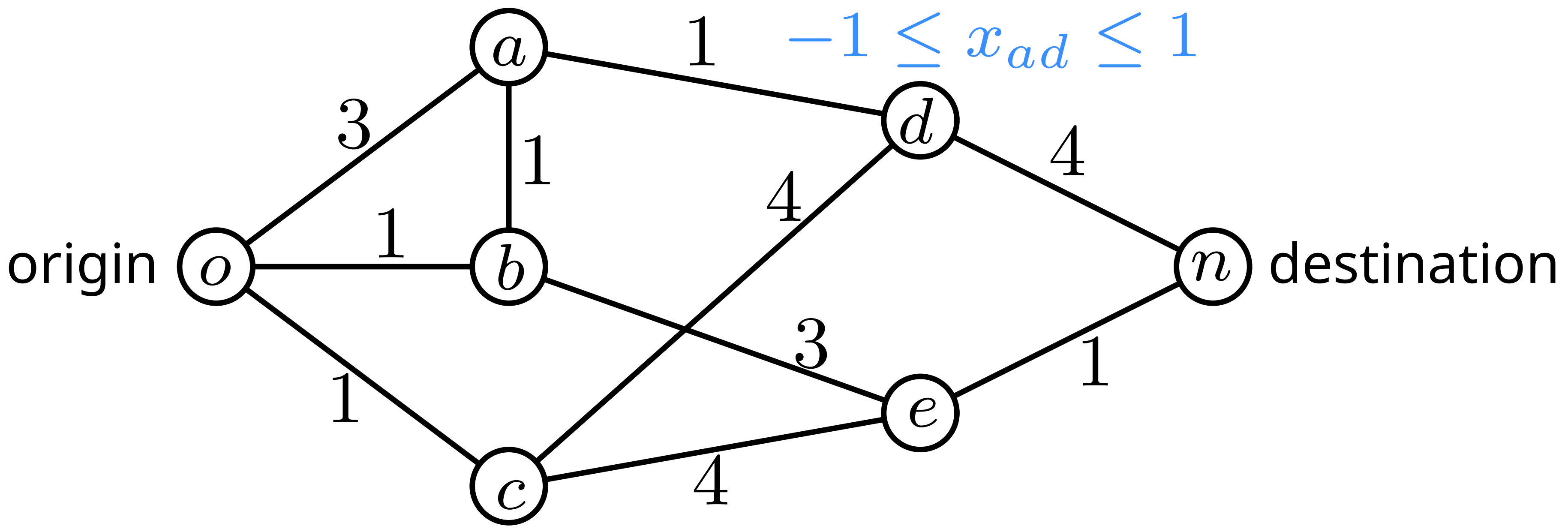
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

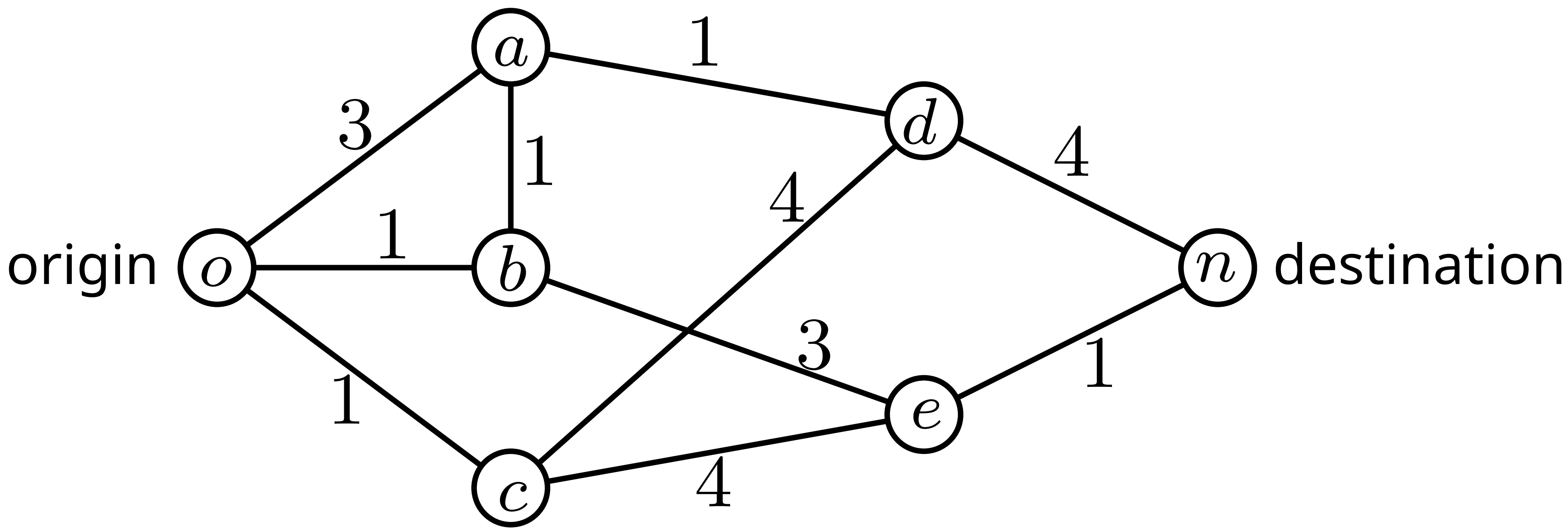
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

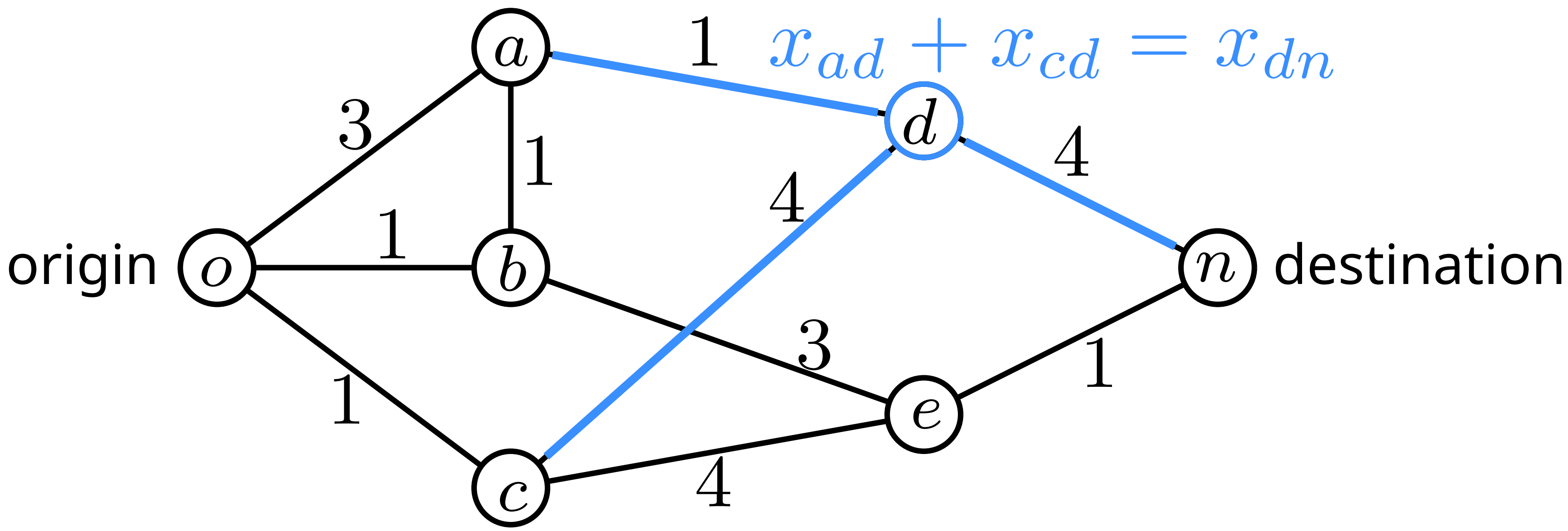
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges how?
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

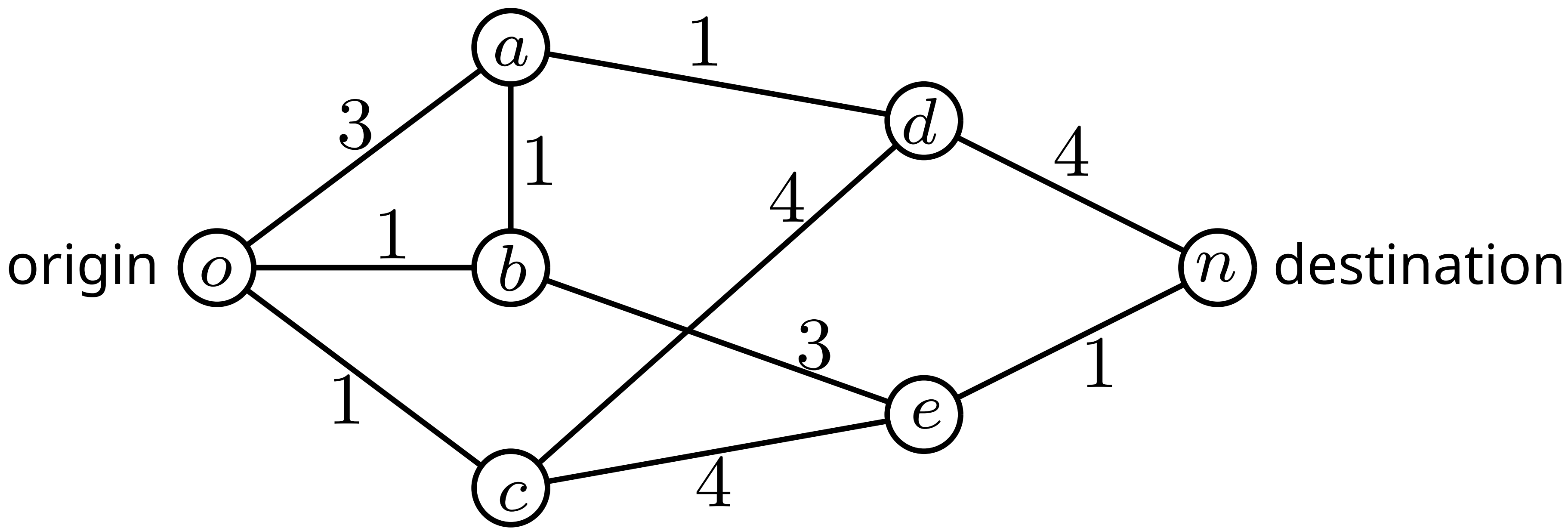
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges how?
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



Maximize . . .?

nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

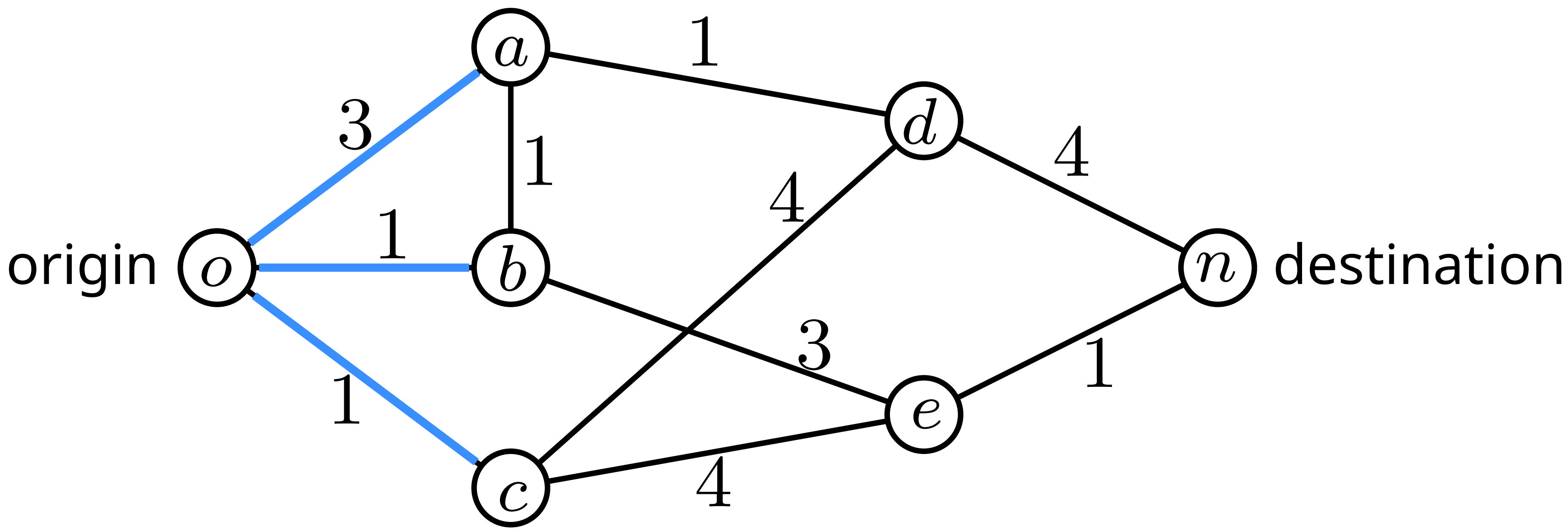
Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

1. flow $\leq$ capacities on edges how?
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

How to send as much data as possible over a local network?



Maximize . . .?

$x_{oa} + x_{ob} + x_{oc}$

nodes cannot store data and links can transport in only one direction

$\rightarrow$ need to determine orientation and amount per edge (with direction)

Can we formulate this as LP?

$\rightarrow$ introduce variable x_{uv} for each edge (u, v) and require

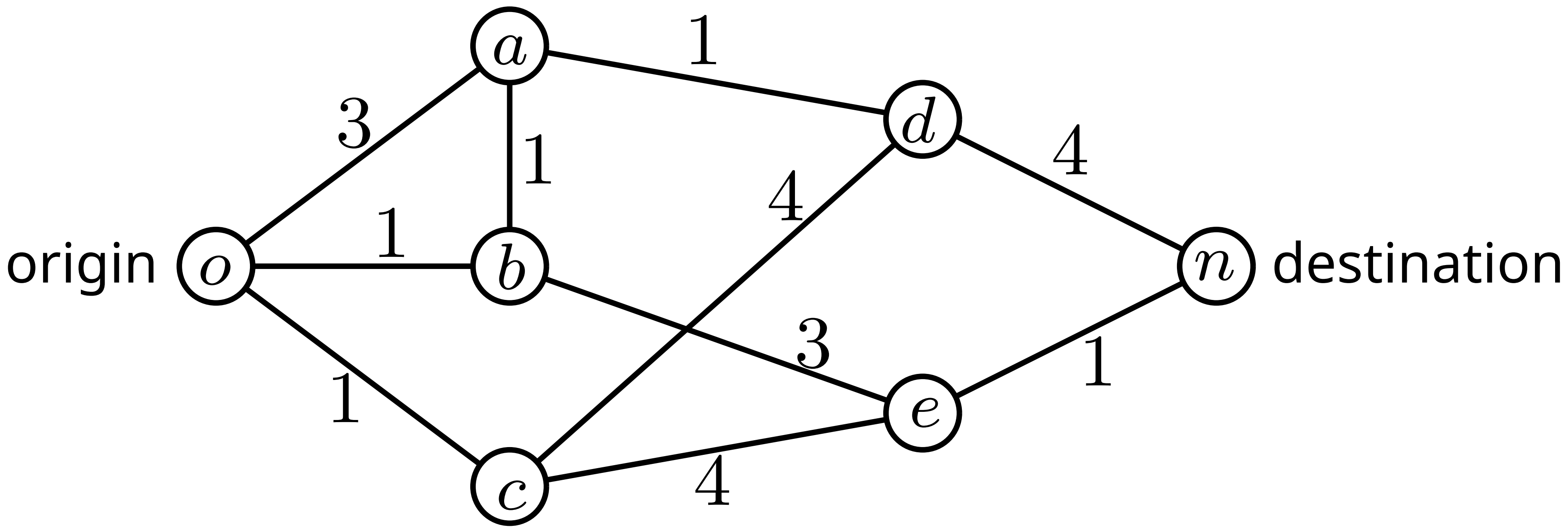
1. flow $\leq$ capacities on edges   how?
2. inflow = outflow on all nodes (except origin, destination)

Example: Flow in a Network

Linear Program Formulation

maximize $x_{oa} + x_{ob} + x_{oc}$

subject to
$$-3 \leq x_{oa} \leq 3,\ -1 \leq x_{ob} \leq 1,\ -1 \leq x_{oc} \leq 1$$
$$-1 \leq x_{ab} \leq 1,\ -1 \leq x_{ad} \leq 1,\ -3 \leq x_{be} \leq 3$$
$$-4 \leq x_{cd} \leq 4,\ -4 \leq x_{ce} \leq 4,\ -4 \leq x_{dn} \leq 4$$
$$-1 \leq x_{en} \leq 1$$
$$x_{oa} = x_{ab} + x_{ad}$$
$$x_{ob} + x_{ab} = x_{be}$$
$$x_{oc} = x_{cd} + x_{ce}$$
$$x_{ad} + x_{cd} = x_{dn}$$
$$x_{be} + x_{ce} = x_{en}$$



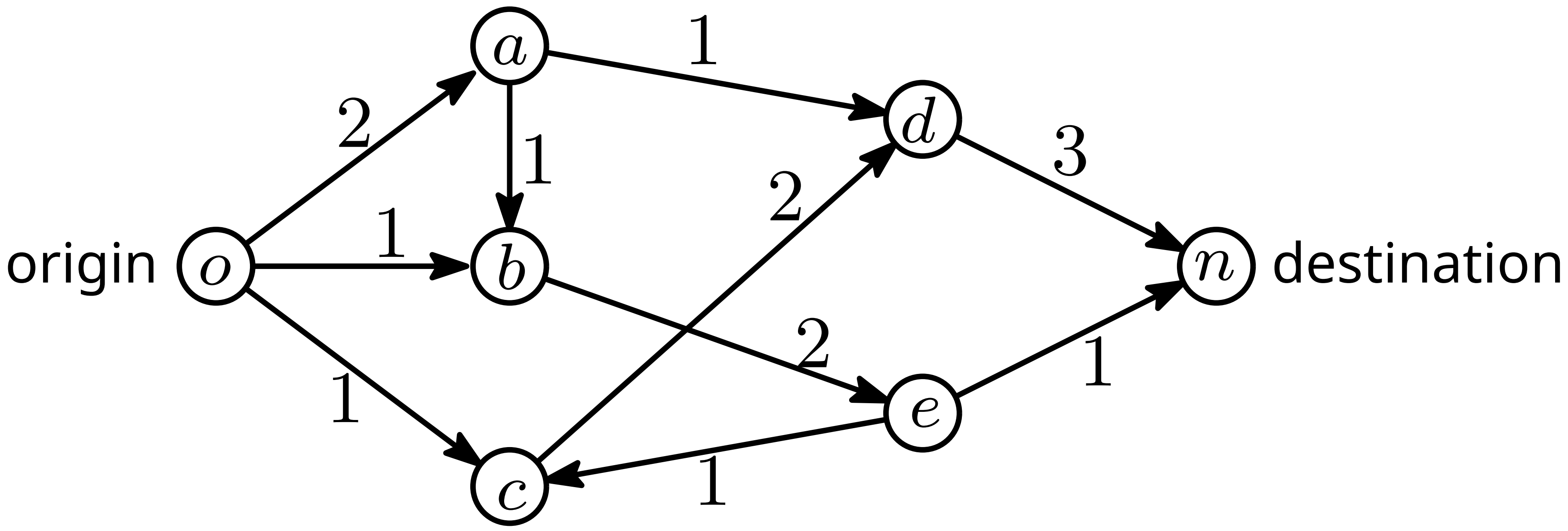
Example: Flow in a Network

Linear Program Formulation

maximize $x_{oa} + x_{ob} + x_{oc}$

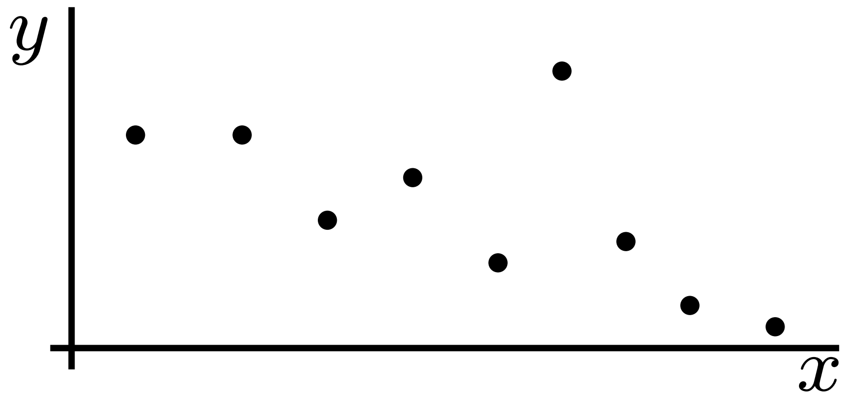
subject to
$$
\begin{aligned}
&-3 \leq x_{oa} \leq 3,\ -1 \leq x_{ob} \leq 1,\ -1 \leq x_{oc} \leq 1\\
&-1 \leq x_{ab} \leq 1,\ -1 \leq x_{ad} \leq 1,\ -3 \leq x_{be} \leq 3\\
&-4 \leq x_{cd} \leq 4,\ -4 \leq x_{ce} \leq 4,\ -4 \leq x_{dn} \leq 4\\
&-1 \leq x_{en} \leq 1\\
&x_{oa} = x_{ab} + x_{ad}\\
&x_{ob} + x_{ab} = x_{be}\\
&x_{oc} = x_{cd} + x_{ce}\\
&x_{ad} + x_{cd} = x_{dn}\\
&x_{be} + x_{ce} = x_{en}
\end{aligned}
$$

Optimal solution: 4



Example: Fitting a Line

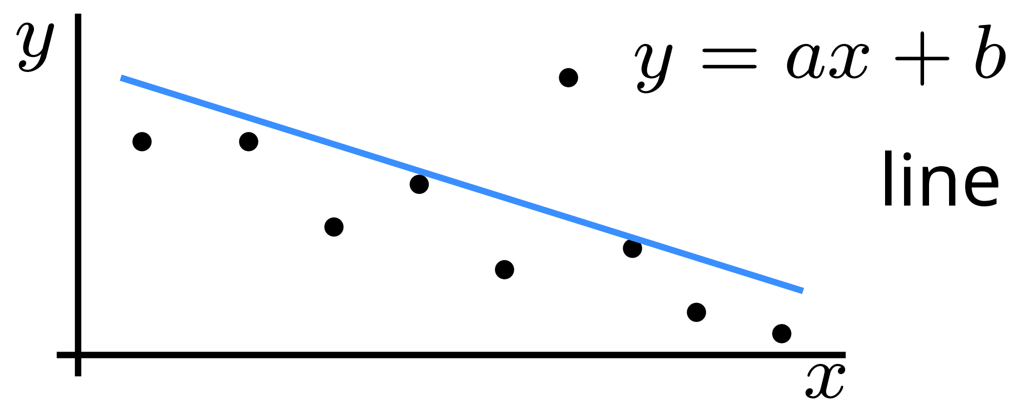
Given 2d data points, whose coordinates have a linear dependency (e.g., physical measurements),



Example: Fitting a Line

Given 2d data points, whose coordinates have a linear dependency (e.g., physical measurements),

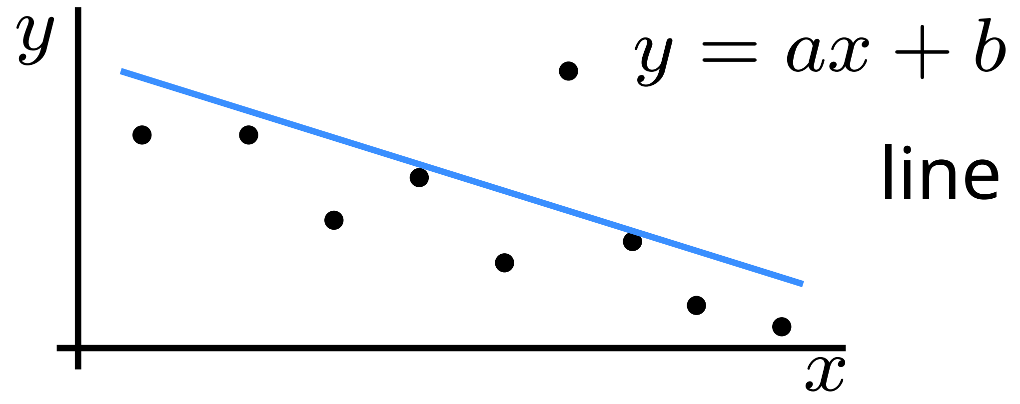
we want to quantify the linear dependency by fitting a line through the points.



Example: Fitting a Line

Given 2d data points, whose coordinates have a linear dependency (e.g., physical measurements),

we want to quantify the linear dependency by fitting a line through the points.



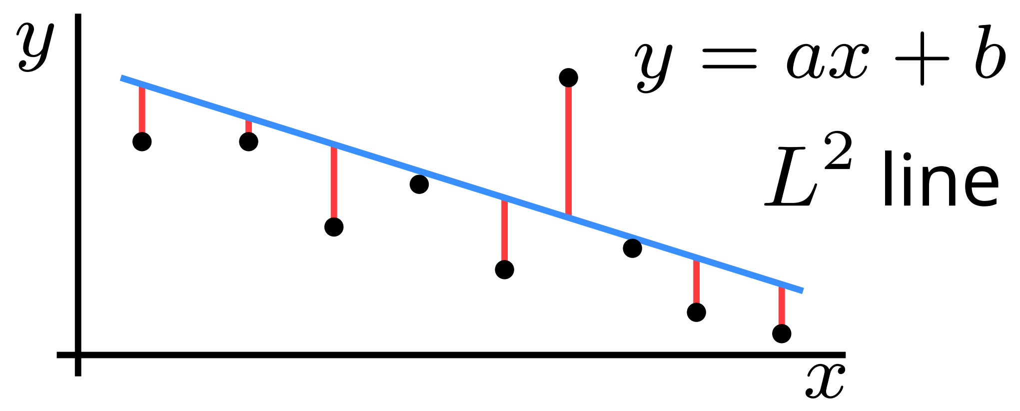
How to best fit a line, i.e., measure the error to a line?

Example: Fitting a Line

L^2 error:

minimize $\sum_{i=1}^{n}(ax_i + b - y_i)^2$

(sensitive to extreme outliers)

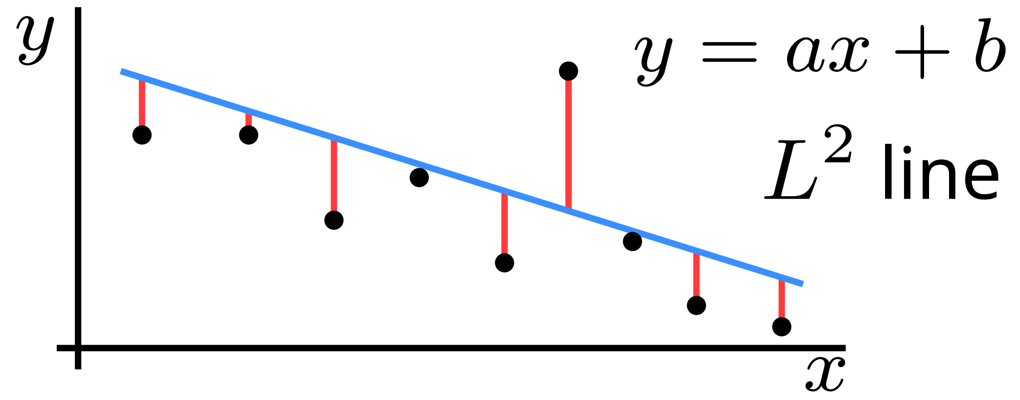


Example: Fitting a Line

L^2 error:

minimize $\sum_{i=1}^{n}(ax_i + b - y_i)^2$

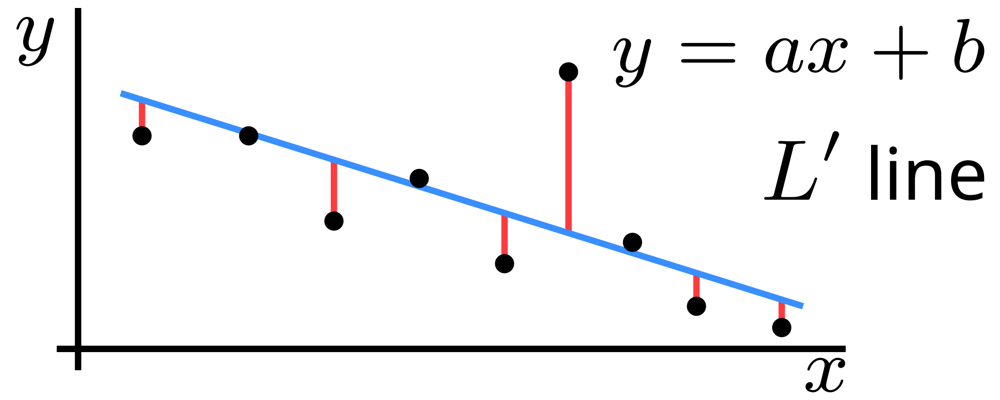
(sensitive to extreme outliers)



L' error:

minimize $\sum_{i=1}^{n}|ax_i + b - y_i|$

Can we formulate this as LP?

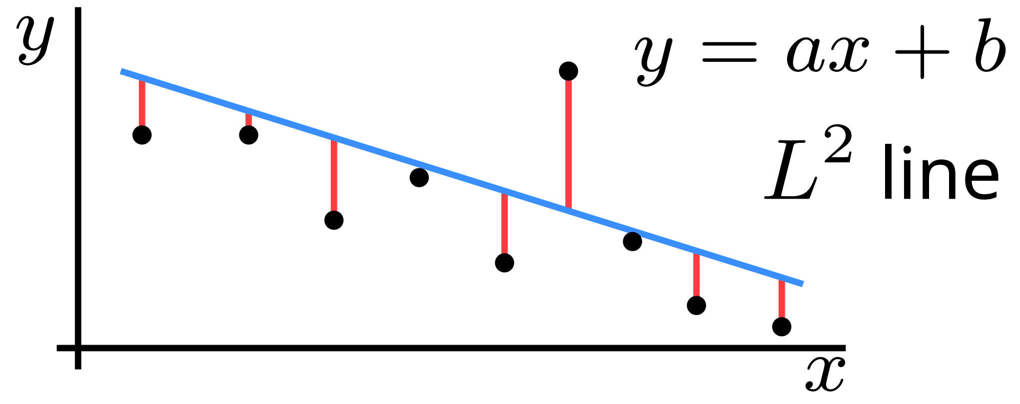


Example: Fitting a Line

L^2 error:

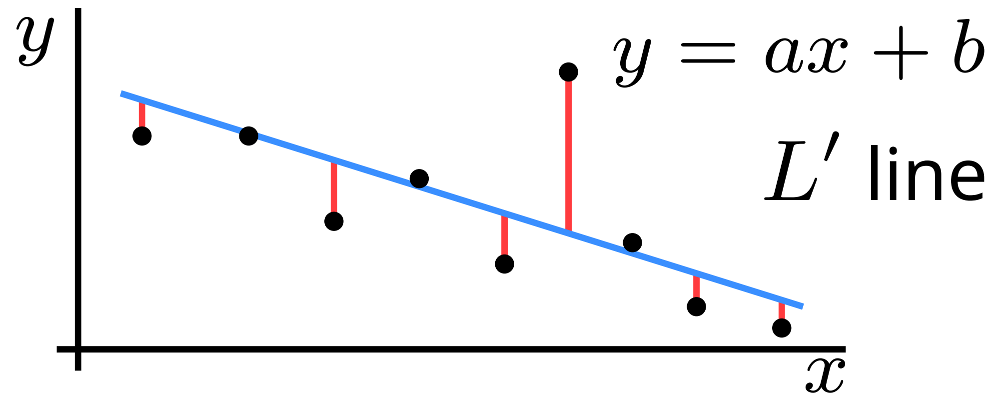
minimize $\sum_{i=1}^{n}(ax_i + b - y_i)^2$

(sensitive to extreme outliers)



L' error:

minimize $\sum_{i=1}^{n}|ax_i + b - y_i|$



Can we formulate this as LP?

minimize $e_1 + e_2 + \ldots + e_n$

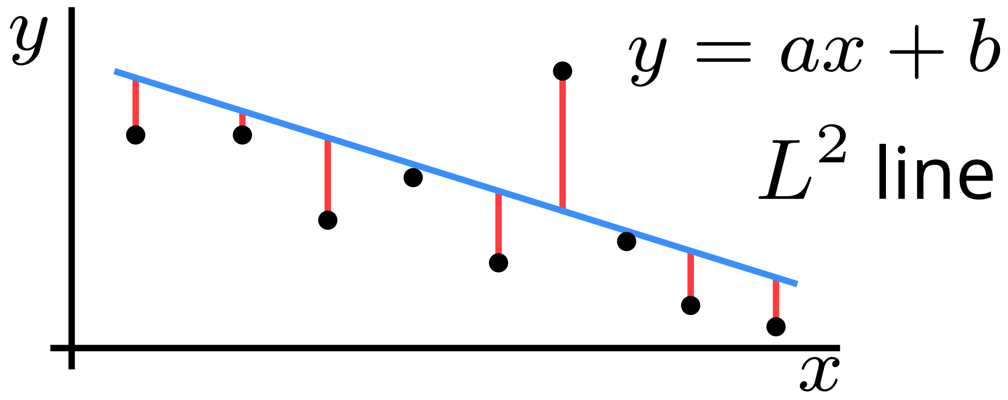
subject to $e_i = |ax_i + b - y_i|$ for all i

Example: Fitting a Line

L^2 error:

minimize $\sum_{i=1}^{n} (ax_i + b - y_i)^2$

(sensitive to extreme outliers)



L' error:

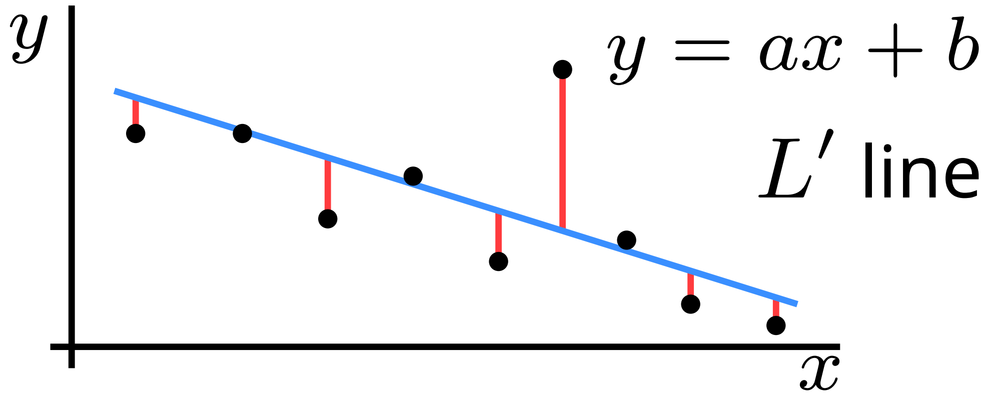
minimize $\sum_{i=1}^{n} |ax_i + b - y_i|$

Can we formulate this as LP?

minimize $e_1 + e_2 + \ldots + e_n$

subject to $e_i = |ax_i + b - y_i|$ for all i

not (yet) an LP!



Example: Fitting a Line

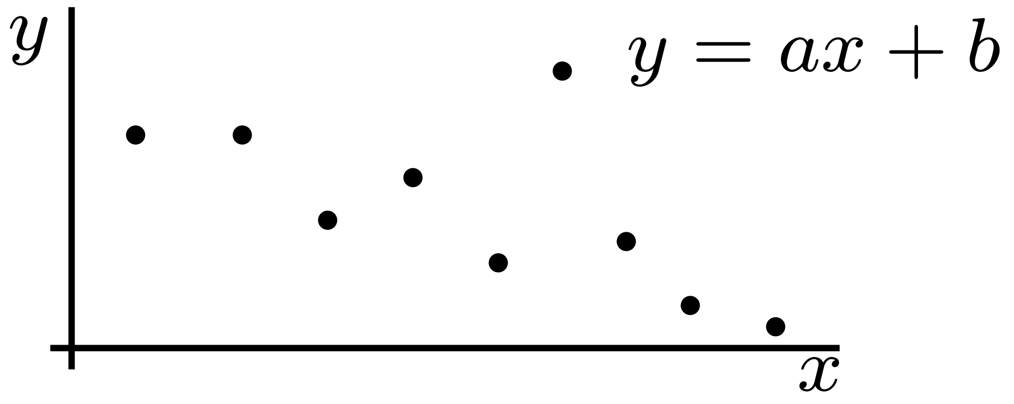
Linear program:

minimize $e_1 + e_2 + ... + e_n$

subject to $e_i \geq ax_i + b - y_i$

$e_i \geq -(ax_i + b - y_i)$

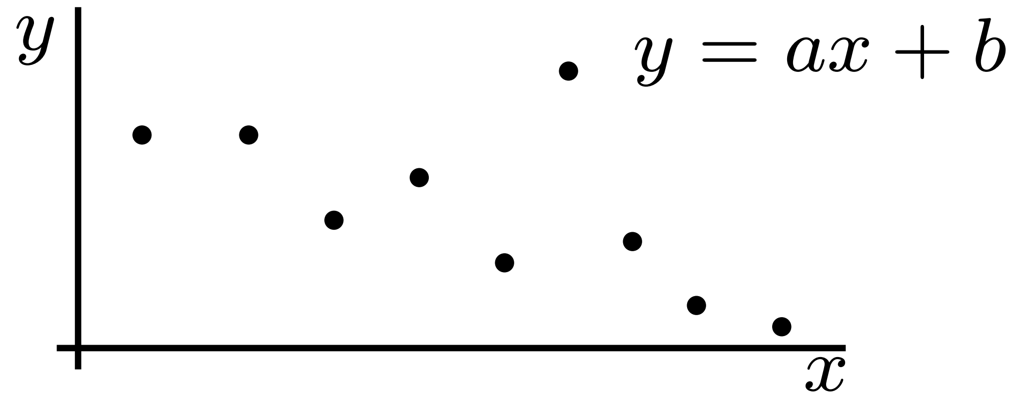
for all i



Example: Fitting a Line

Linear program:

$$\begin{aligned} \text{minimize } & e_1 + e_2 + \ldots + e_n \\ \text{subject to } & e_i \geq ax_i + b - y_i \\ & e_i \geq -(ax_i + b - y_i) \\ & \text{for all } i \end{aligned}$$



The constraints guarantee
$e_i \geq \max\{ax_i+b-y_i, -(ax_i+b-y_i)\} = |ax_i+b-y_i|$.

In an optimal solution, is satisfied with equality.

Example: Fitting a Line

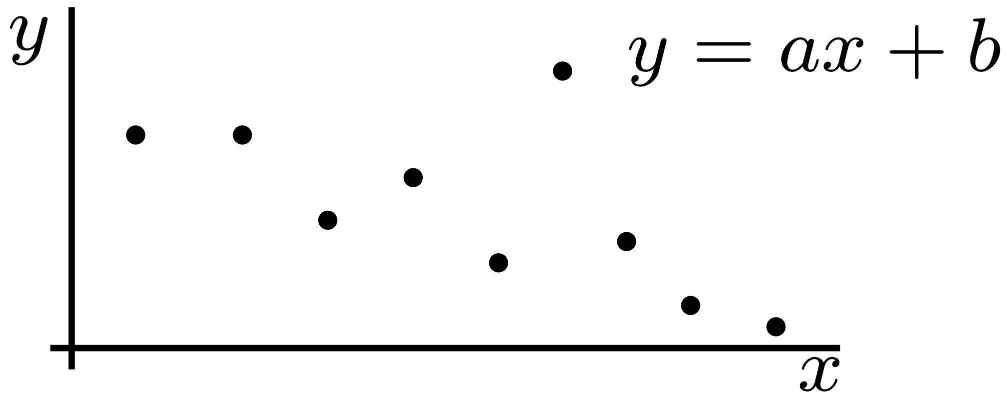
Linear program:

minimize $e_1 + e_2 + \ldots + e_n$

subject to $e_i \geq ax_i + b - y_i$

$e_i \geq -(ax_i + b - y_i)$

for all i



The constraints guarantee

$$e_i \geq \max\{ax_i + b - y_i, -(ax_i + b - y_i)\} = |ax_i + b - y_i|.$$

In an optimal solution, $\geq$ is satisfied with equality.

How many variables do we have for n data points?

Example: Fitting a Line

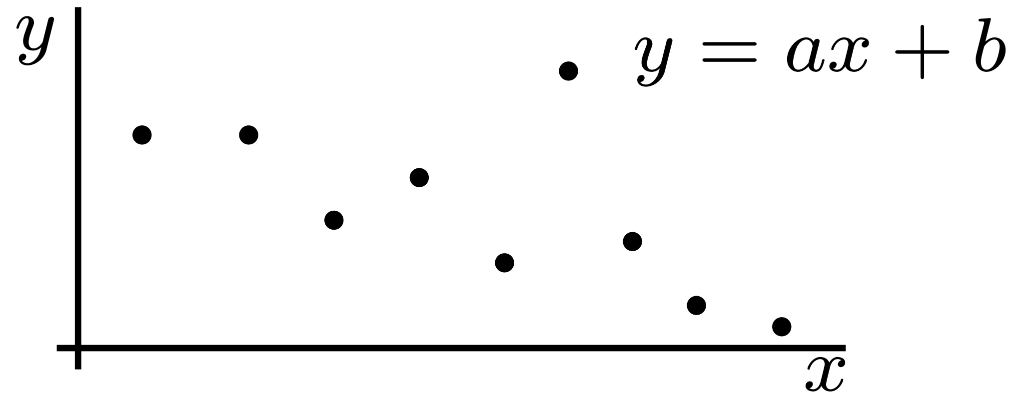
Linear program:

minimize $e_1 + e_2 + \ldots + e_n$

subject to $e_i \geq ax_i + b - y_i$

$e_i \geq -(ax_i + b - y_i)$

for all i



The constraints guarantee

$$e_i \geq \max\{ax_i + b - y_i, -(ax_i + b - y_i)\} = |ax_i + b - y_i|.$$

In an optimal solution, is satisfied with equality.

How many variables do we have for n data points?

n+2; per point

e_i, a, b

Example: Fitting a Line

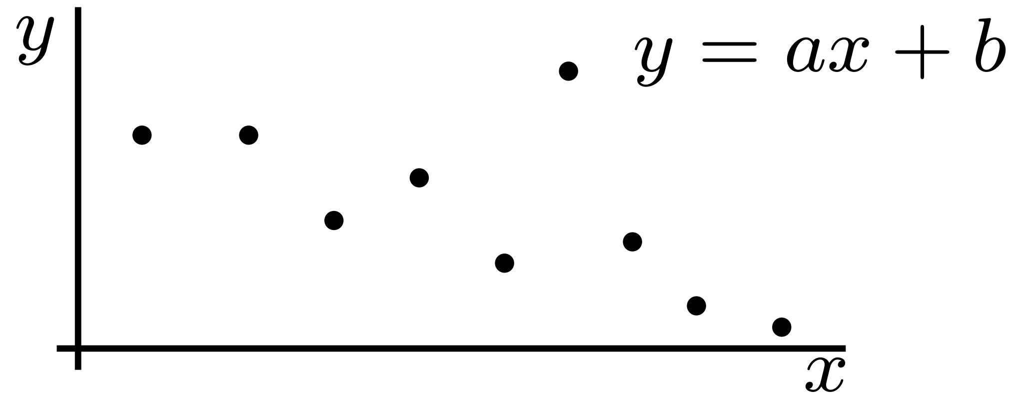
Linear program:

minimize $e_1 + e_2 + \ldots + e_n$

subject to $e_i \geq ax_i + b - y_i$

$e_i \geq -(ax_i + b - y_i)$

for all i



The constraints guarantee

$$e_i \geq \max\{ax_i + b - y_i, -(ax_i + b - y_i)\} = |ax_i + b - y_i|.$$

In an optimal solution, is satisfied with equality.

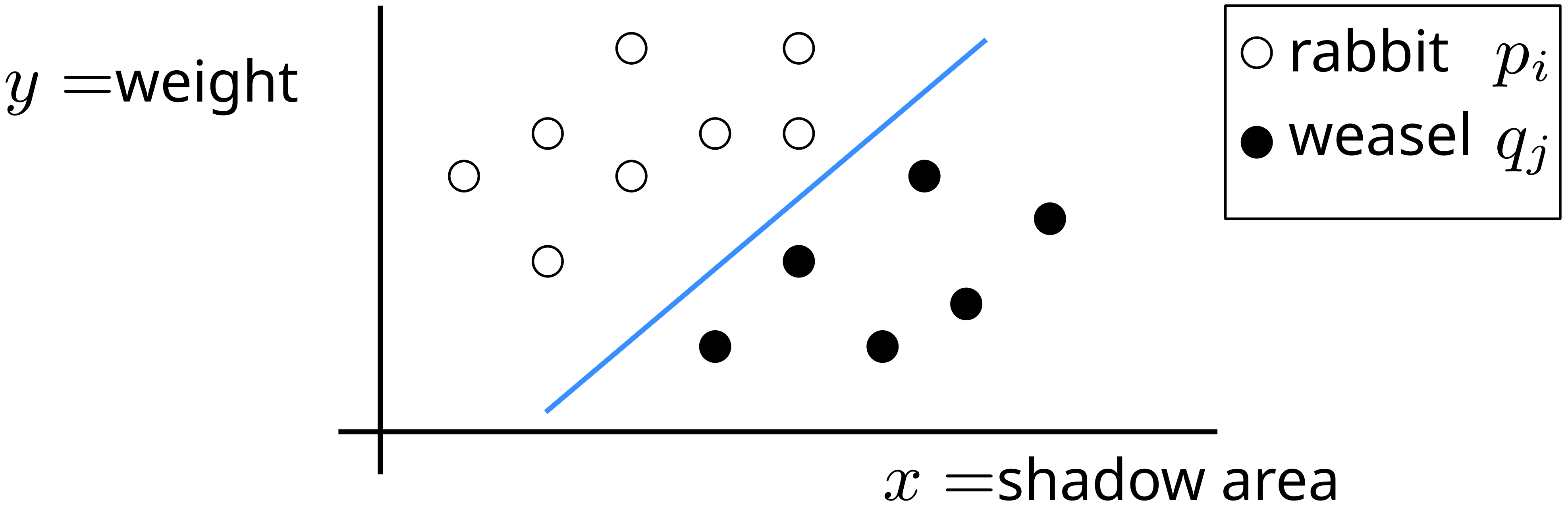
How many variables do we have for n data points?

n+2; per point

e_i, a, b

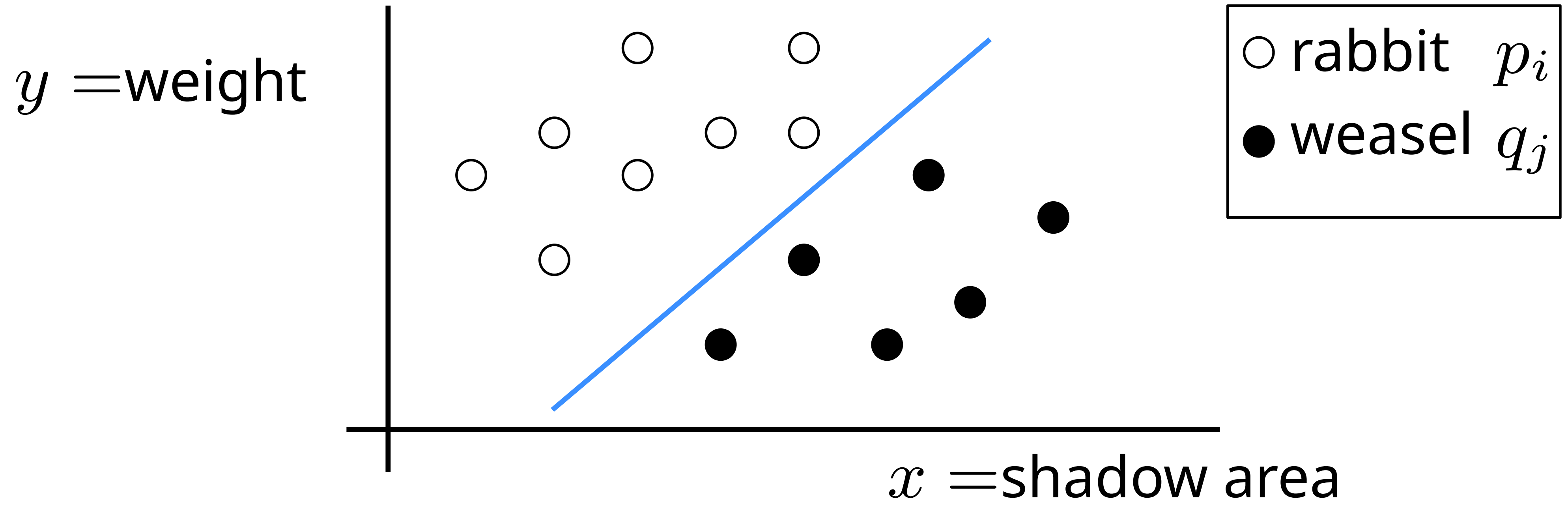
Moral: Objective functions or constraints with absolute values can often be handled by introducing extra variables or constraints.

Example: Separation of points



Does there exist a separating line $y = ax + b$?

Example: Separation of points

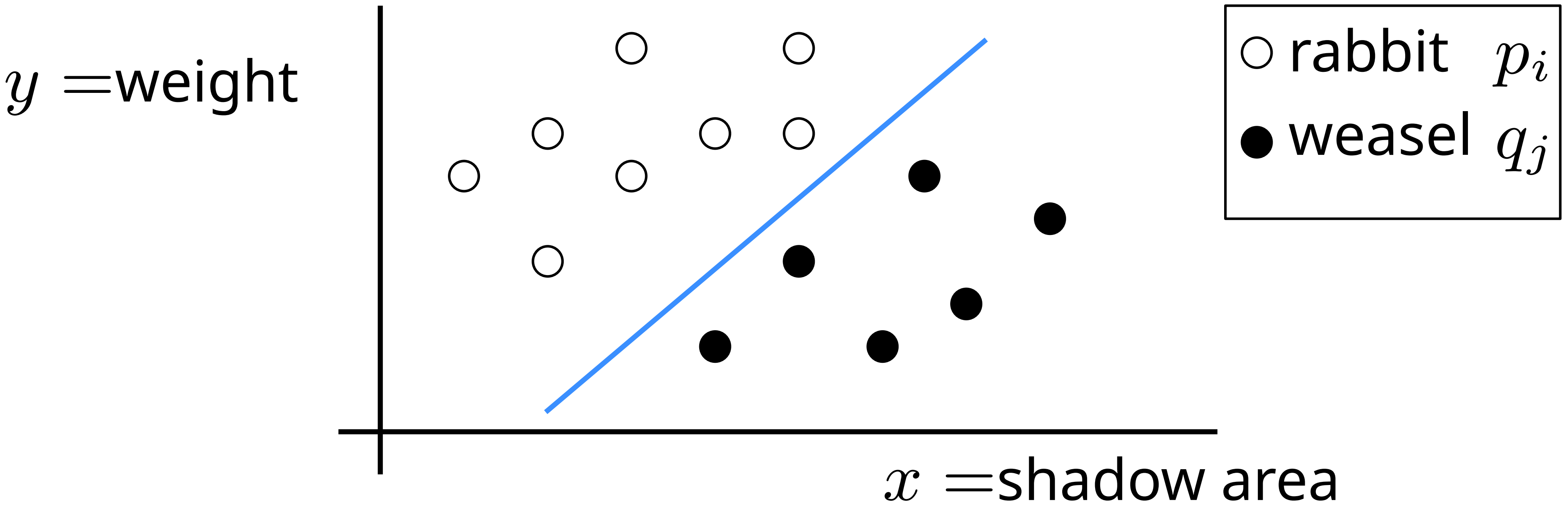


Does there exist a separating line $y = ax + b$?

Case p_i points "above" q_j points:

$$y(p_i) > ax(p_i) + b \qquad \text{for all } i$$
$$y(q_i) < ax(q_j) + b \qquad \text{for all } j$$

Example: Separation of points



Does there exist a separating line $y = ax + b$?

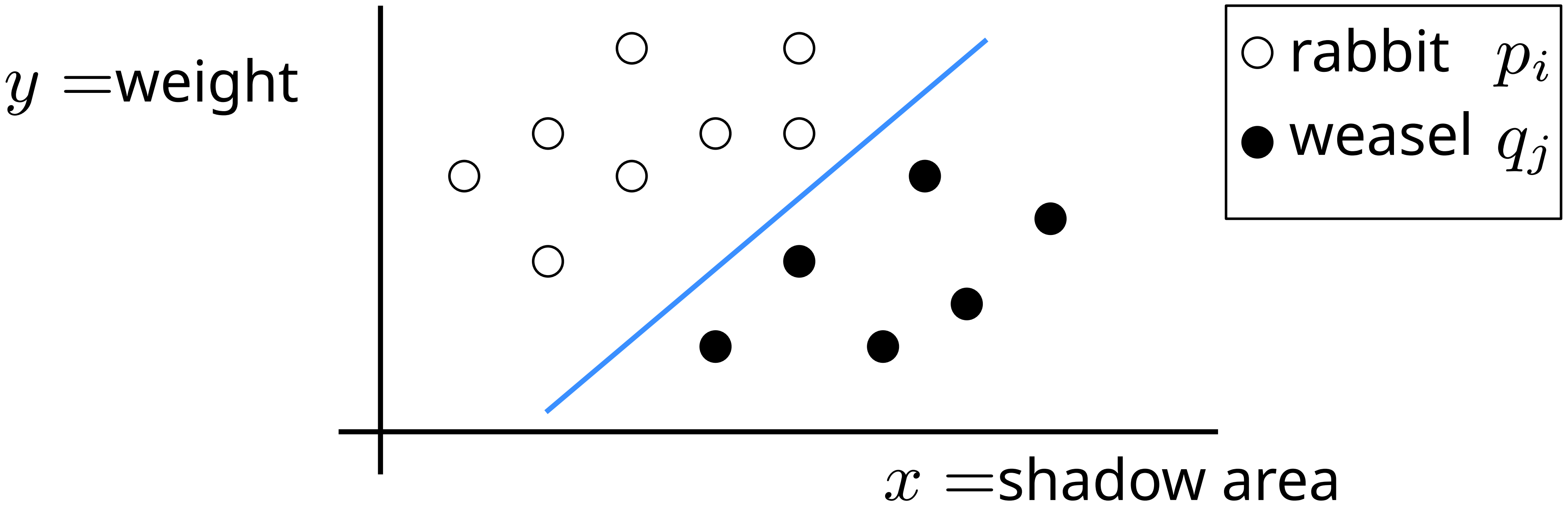
Case p_i points "above" q_j points:

$$y(p_i) > ax(p_i) + b \qquad \text{for all } i$$

$$y(q_i) < ax(q_j) + b \qquad \text{for all } j$$

strict inequalities
not allowed

Example: Separation of points



Does there exist a separating line $y = ax + b$?

Case p_i points "above" q_j points:

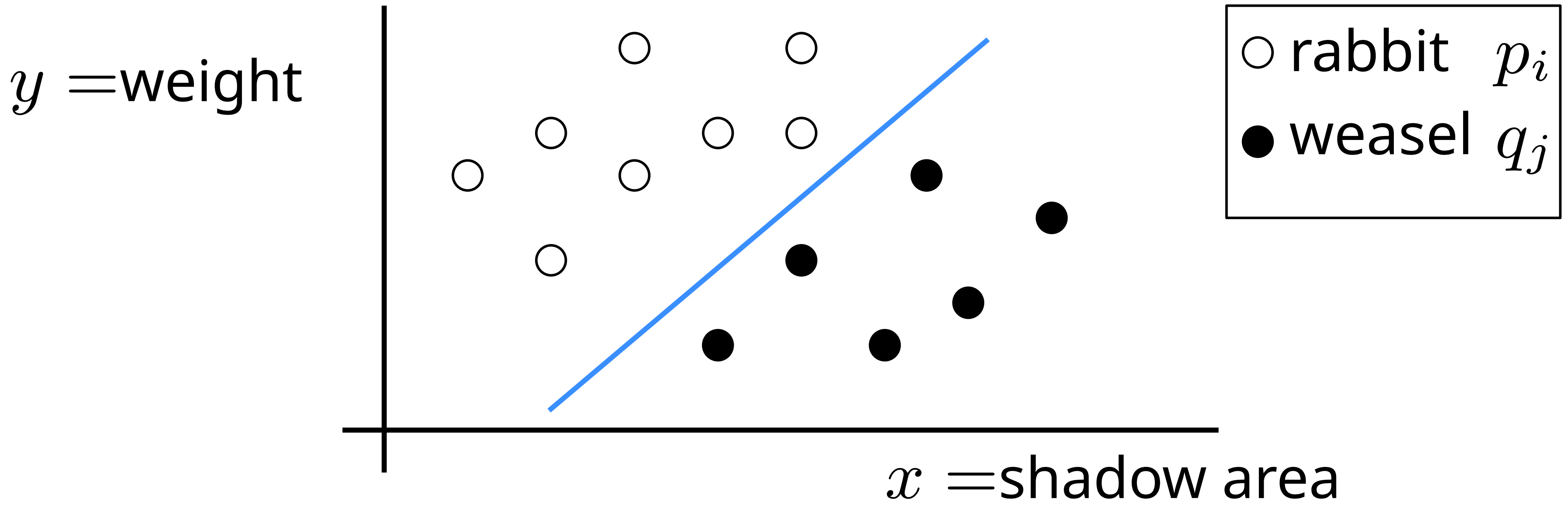
introduce variable δ for separation

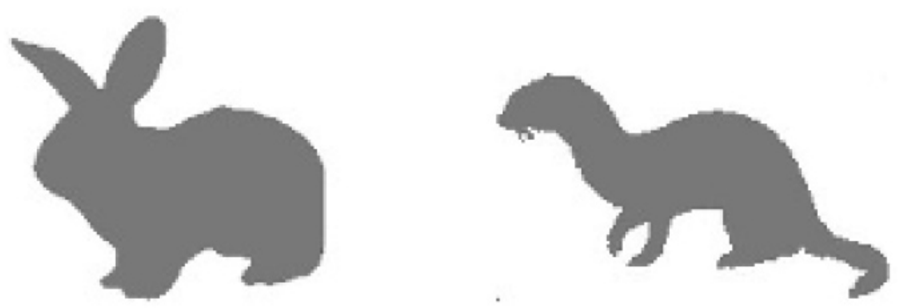
maximize δ

subject to $y(p_i) \geq ax(p_i) + b + \delta$ for all i

$y(q_i) \leq ax(q_j) + b - \delta$ for all j

Example: Separation of points





Does there exist a separating line $y = ax + b$?

Case p_i points "above" q_j points:

introduce variable δ for separation

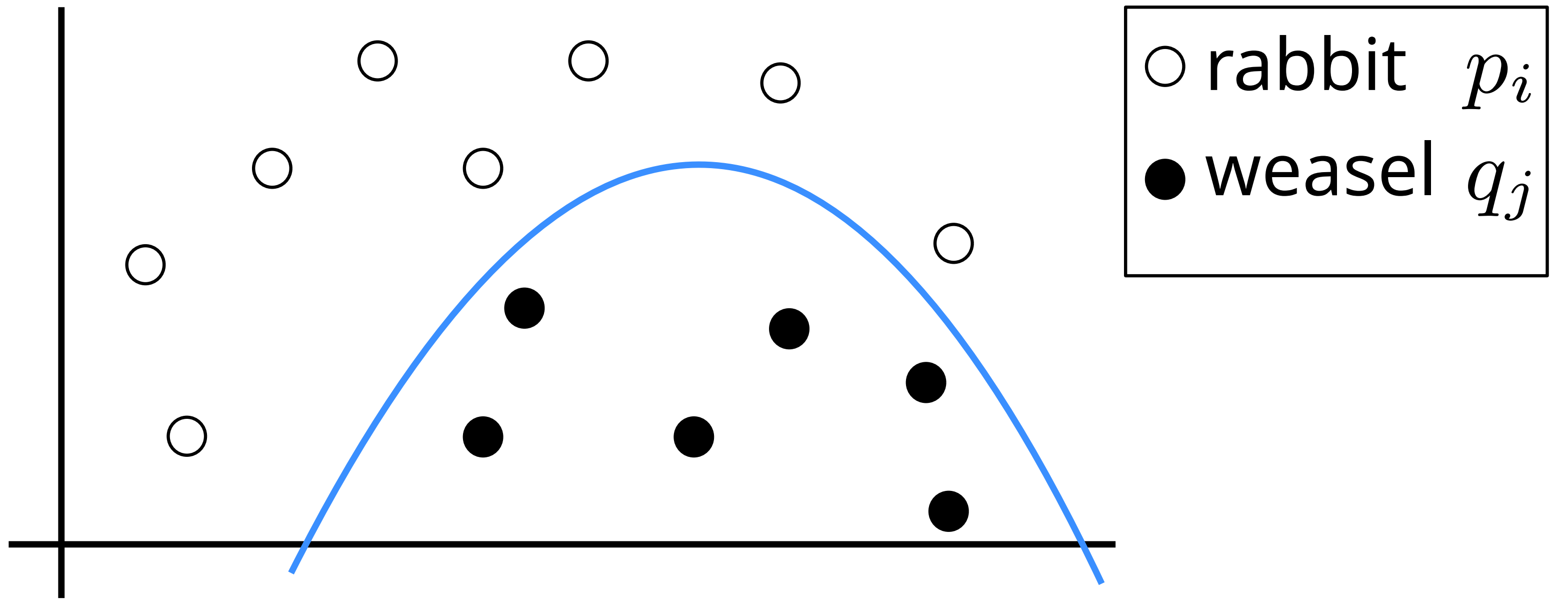
maximize δ

subject to $y(p_i) \geq ax(p_i) + b + \delta$ for all i

$y(q_i) \leq ax(q_j) + b - \delta$ for all j

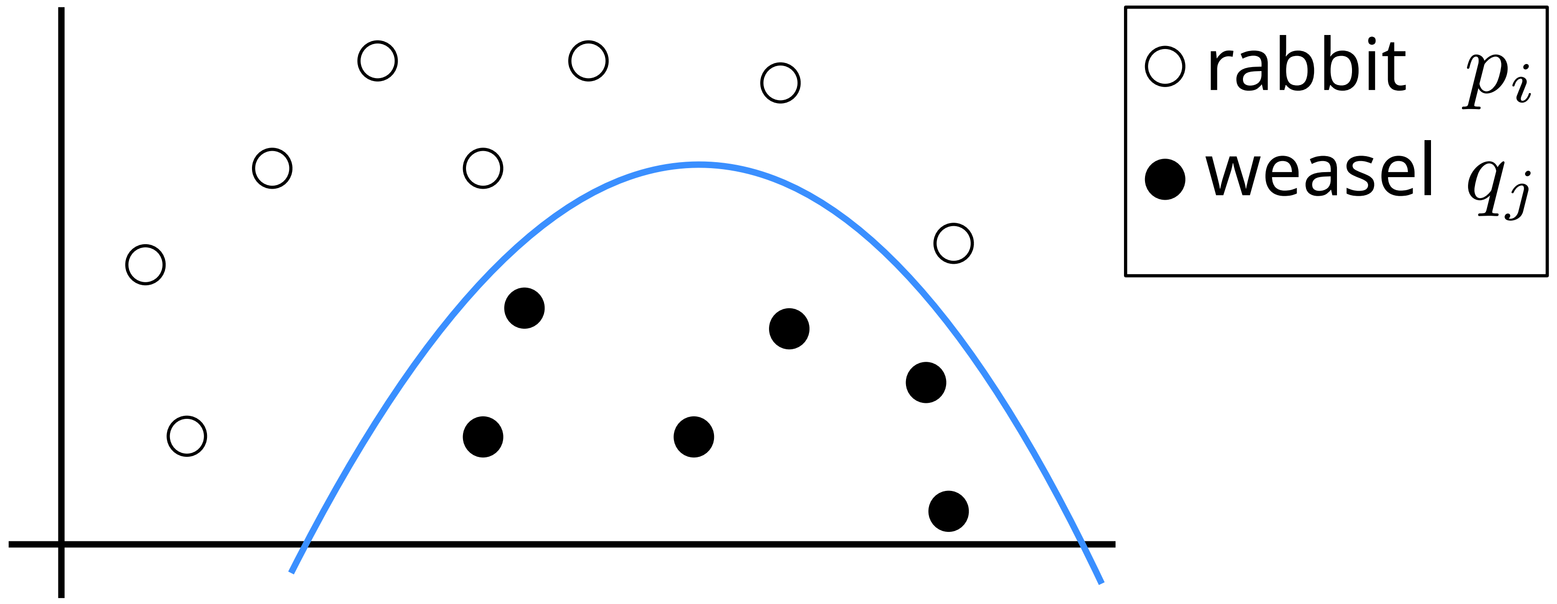
If solution gives $\delta = 0$, then no solution for strict inequalities

Example: Separation of points



Does there exist a separating parabola $y = ax^2 + bx + c$?

Example: Separation of points



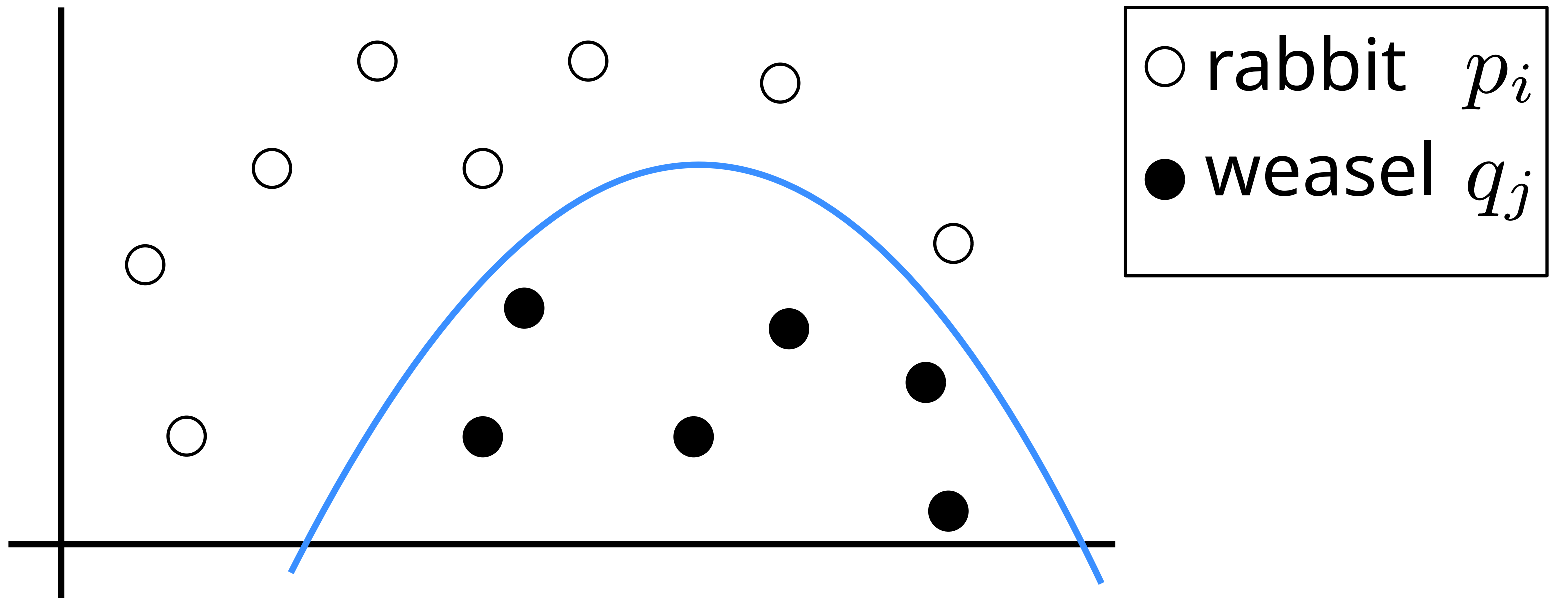
Does there exist a separating parabola $y = ax^2 + bx + c$?

maximize δ

subject to $y(p_i) \geq ax(p_i)^2 + bx(p_i) + c + \delta$ for all i

$y(q_i) \geq ax(q_j)^2 + bx(q_j) + c - \delta$ for all j

Example: Separation of points



Does there exist a separating parabola $y = ax^2 + bx + c$?

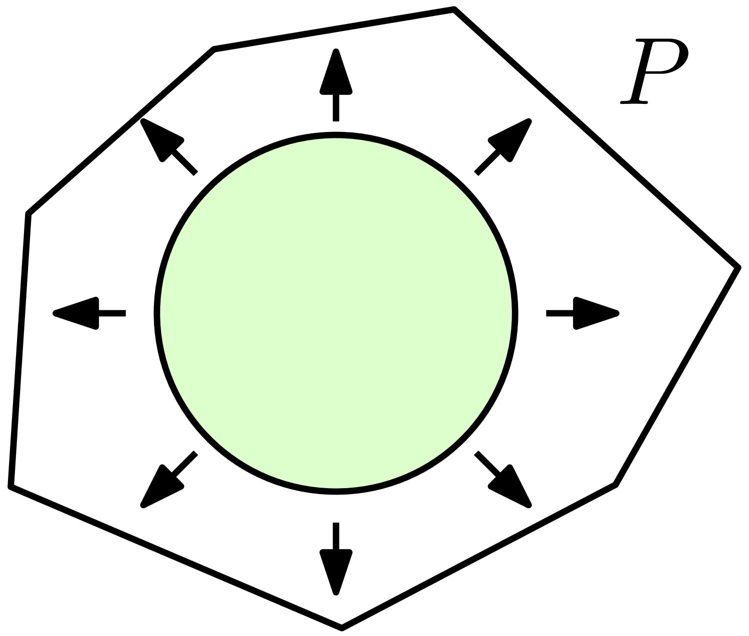
maximize δ

subject to $y(p_i) \geq ax(p_i)^2 + bx(p_i) + c + \delta$ for all i

$y(q_i) \geq ax(q_j)^2 + bx(q_j) + c - \delta$ for all j

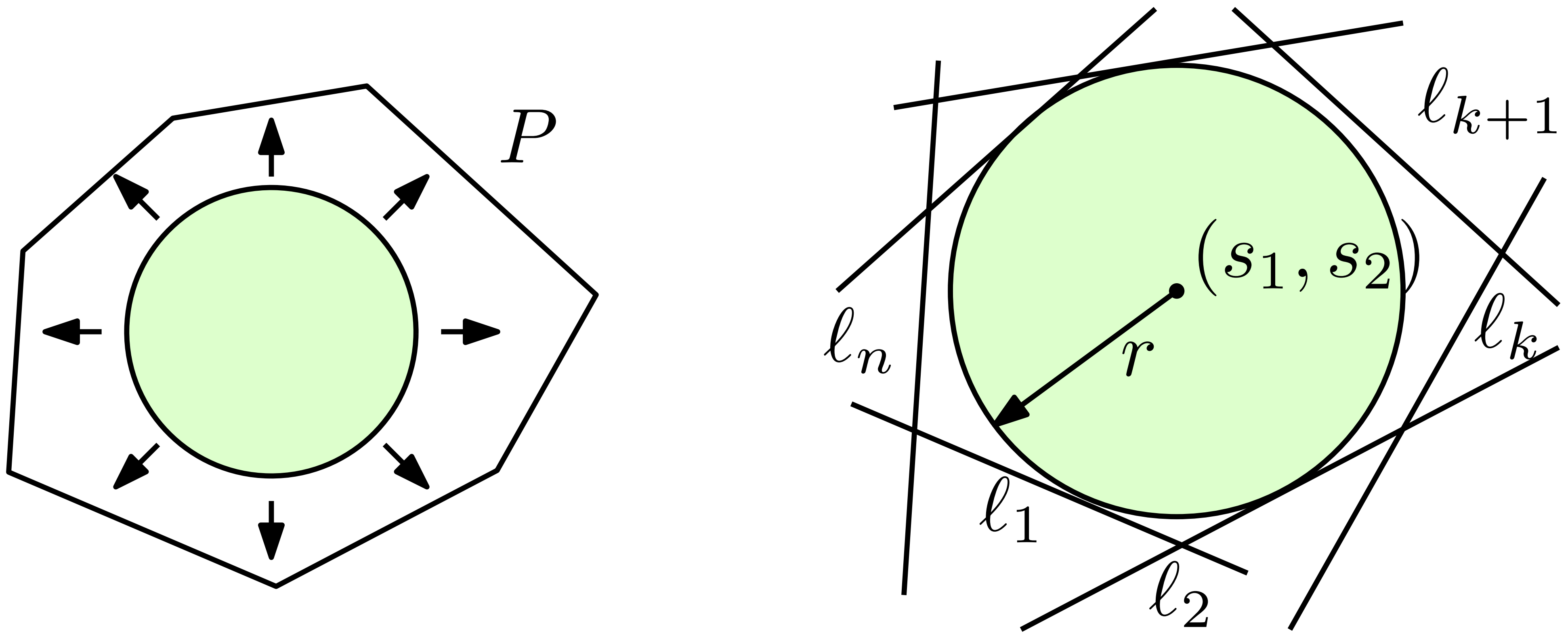
Morals: Strict inequalities can be modeled by extra variable.
Also: Non-linear problems can sometimes be incorporated into coefficients of LP.

Example: Largest Disk in a Polygon



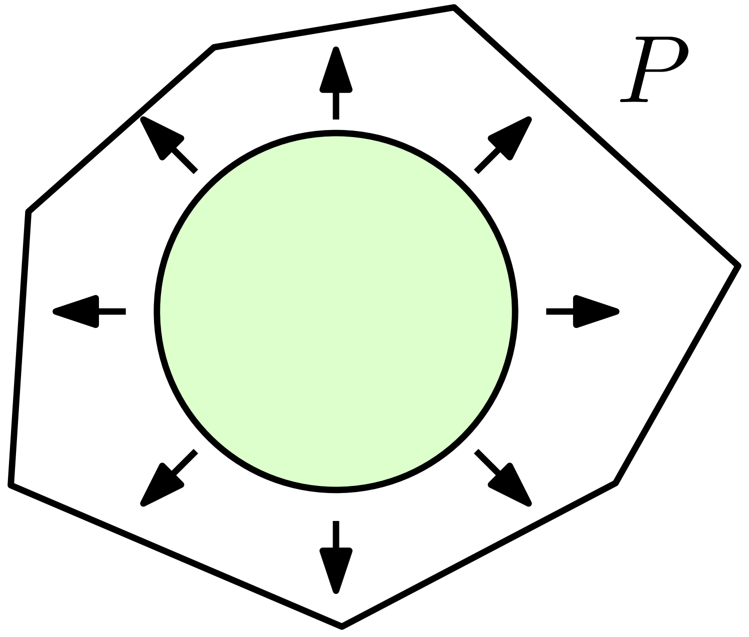
Given a polygon P, find a disk of maximum radius inside P.

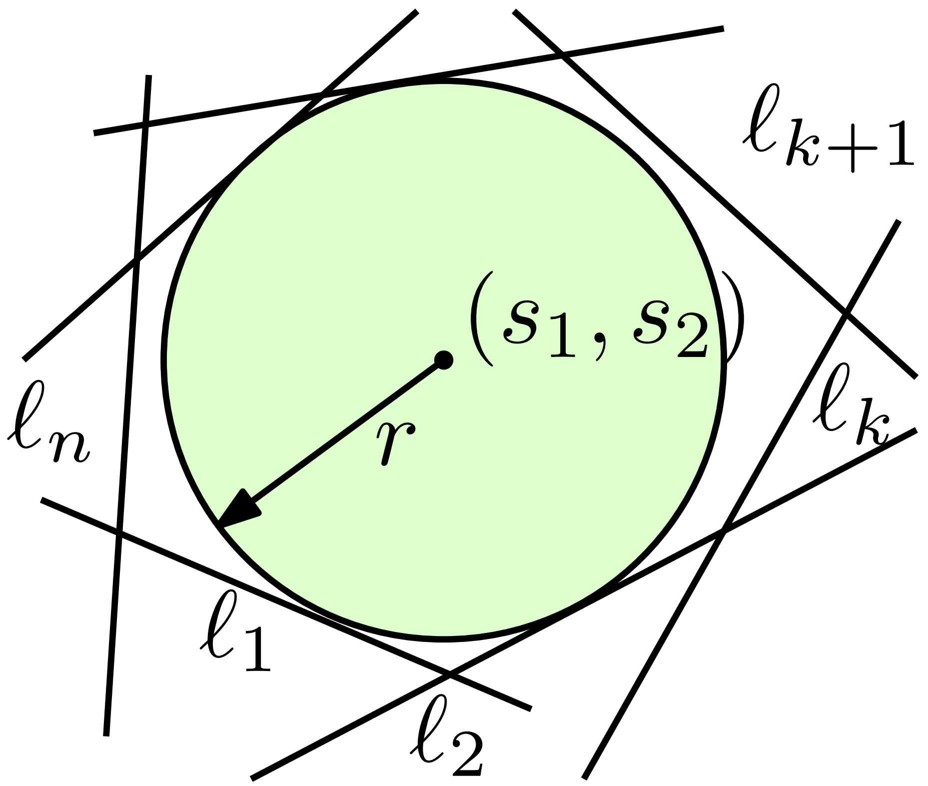
Example: Largest Disk in a Polygon



Given a polygon P, find a disk of maximum radius inside P.

Example: Largest Disk in a Polygon

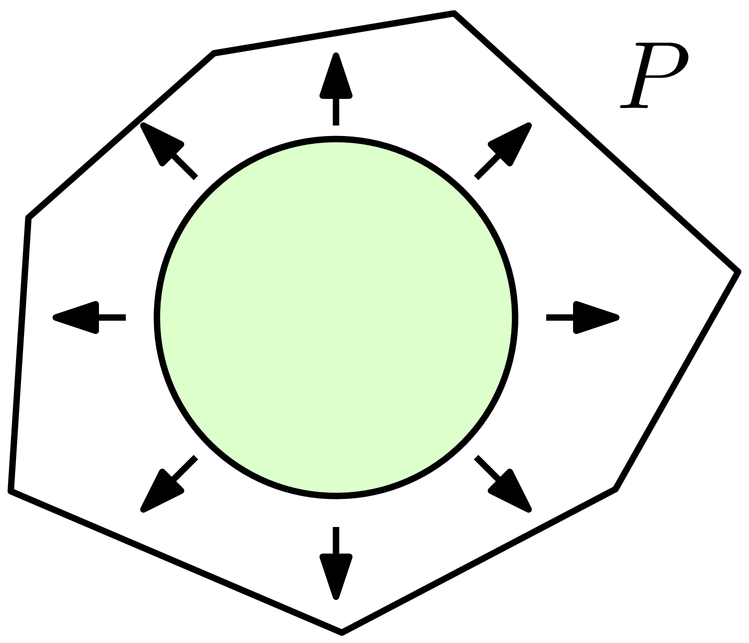


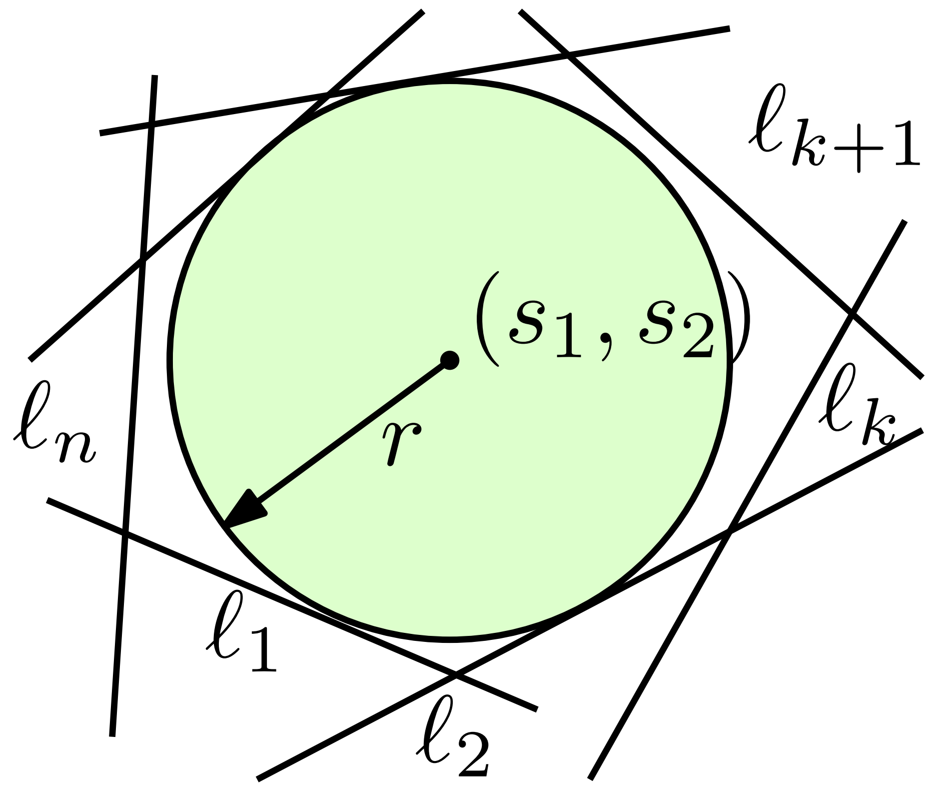


Given a polygon P, find a disk of maximum radius inside P.

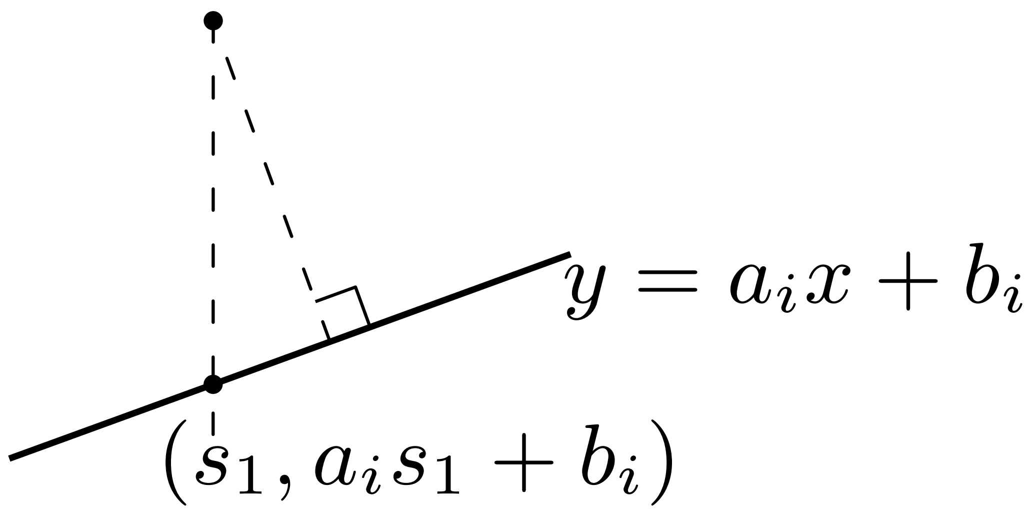
What is the distance of the center point (s_1, s_2) to a line ℓ_i?

Example: Largest Disk in a Polygon

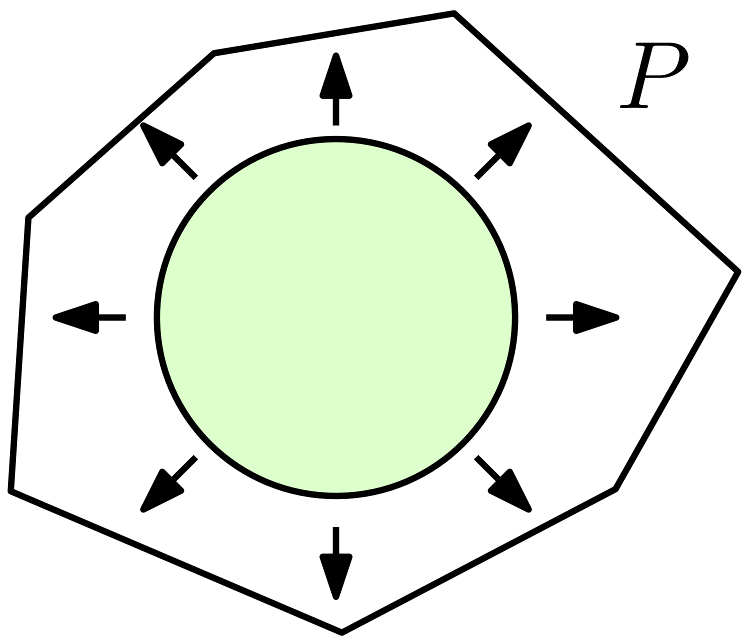


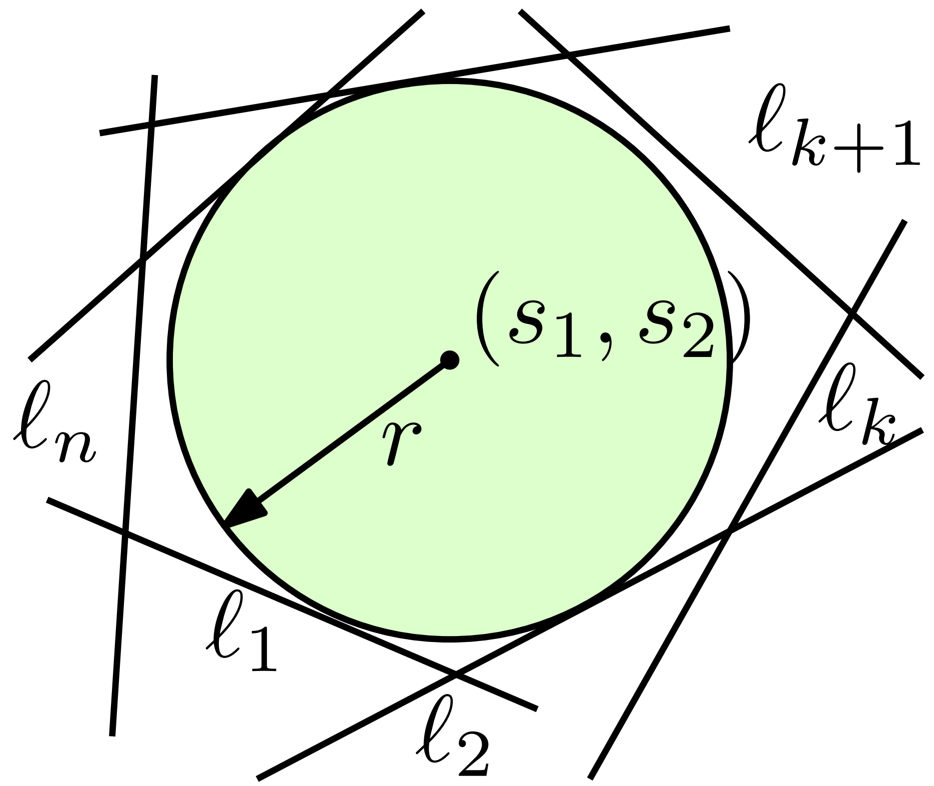


The distance from the center point (s_1, s_2) to the line ℓ_i ($y = a_i x + b_i$) is the absolute value of $\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}$.

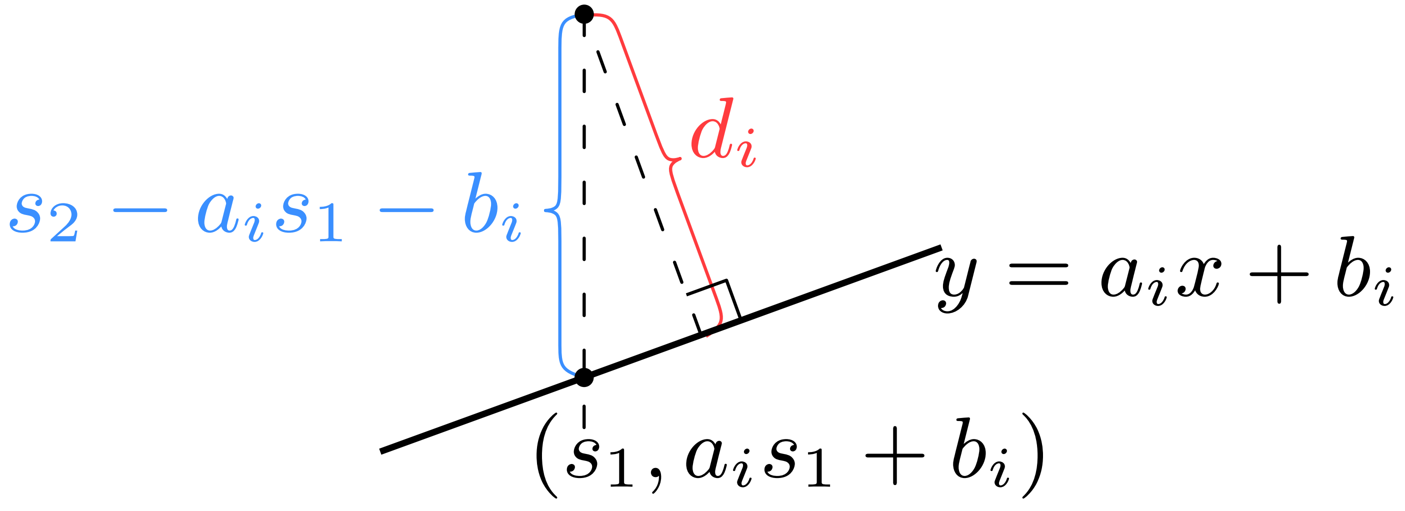


Example: Largest Disk in a Polygon

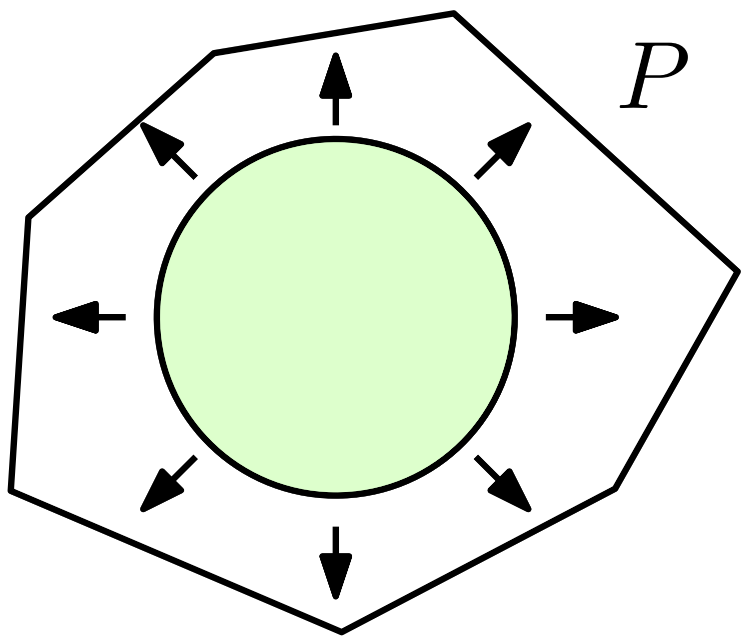


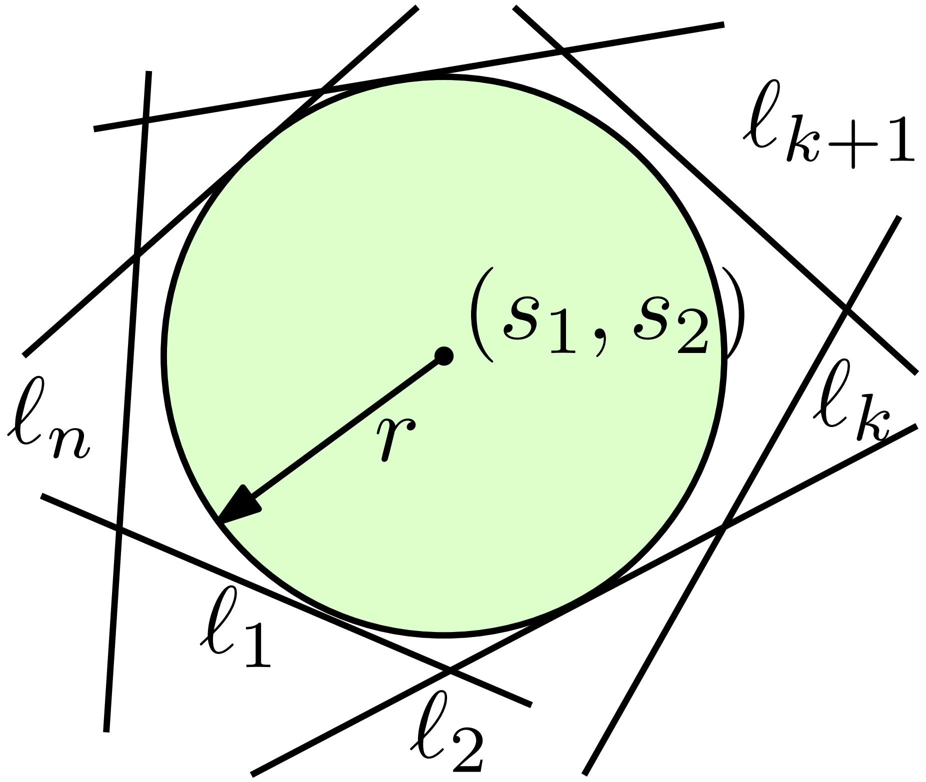


The distance from the center point (s_1, s_2) to the line ℓ_i ($y = a_i x + b_i$) is the absolute value of $\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}$.

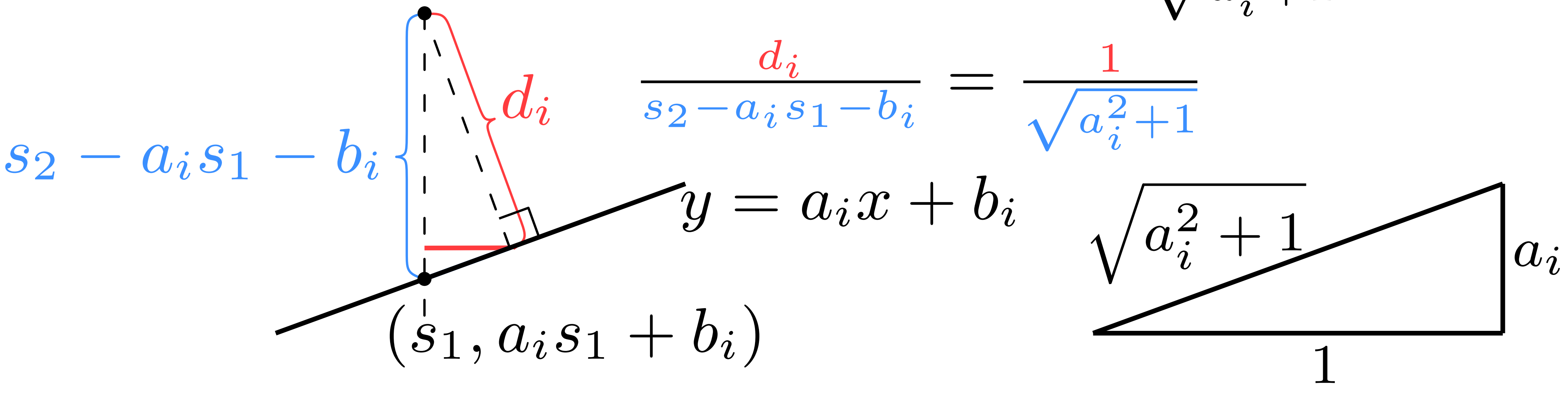


Example: Largest Disk in a Polygon

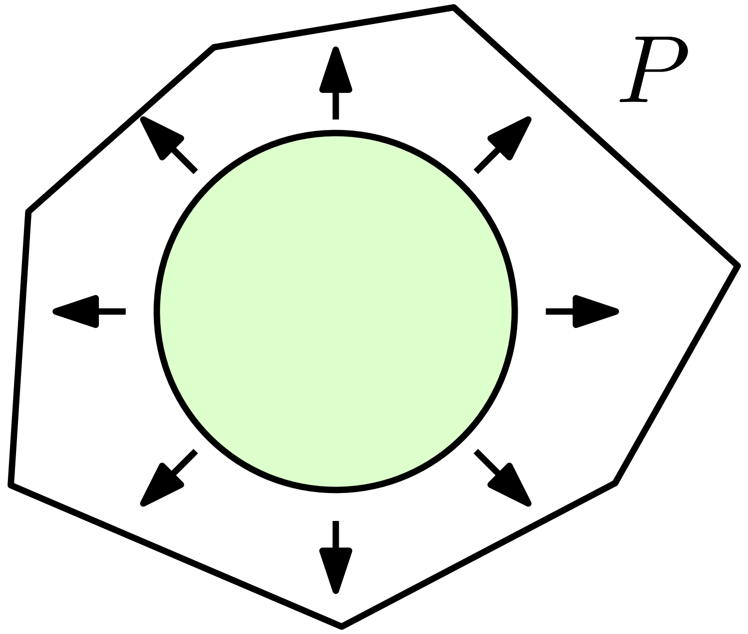


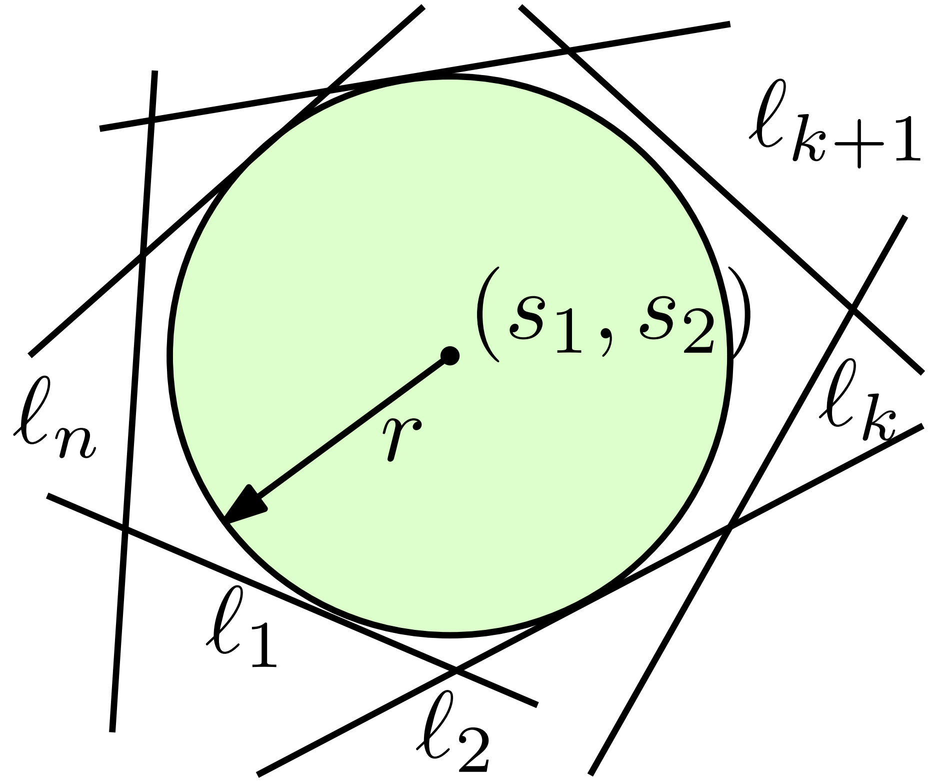


The distance from the center point (s_1, s_2) to the line ℓ_i $(y = a_i x + b_i)$ is the absolute value of $\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}}$.

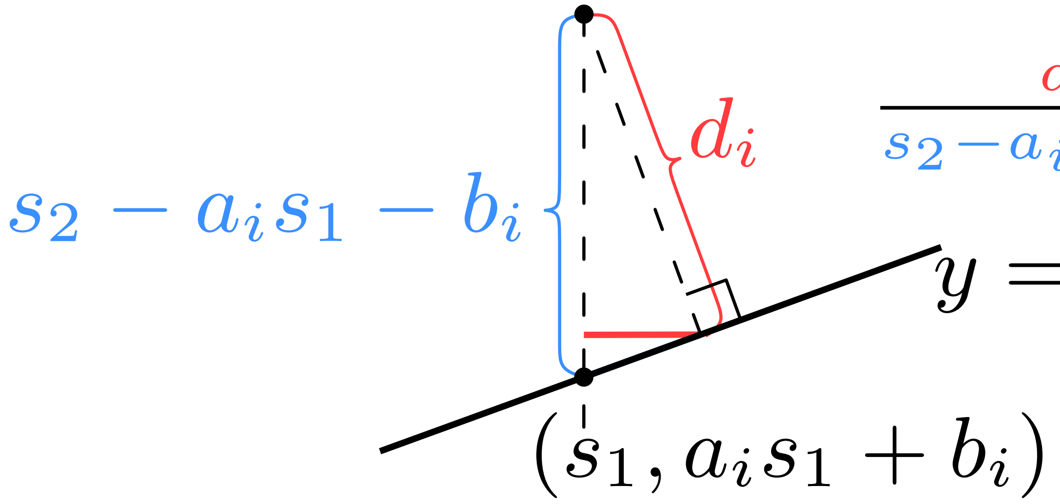


Example: Largest Disk in a Polygon

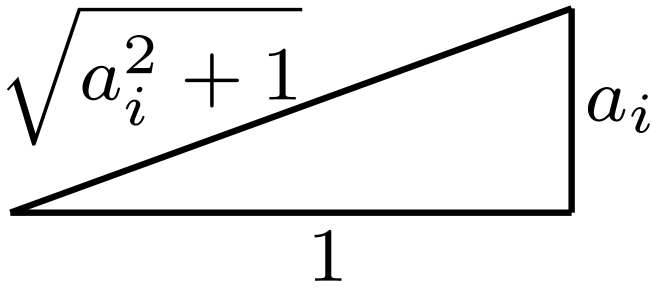




The distance from the center point (s_1, s_2) to the line ℓ_i ($y = a_i x + b_i$) is the absolute value of $\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}}$.



$\frac{d_i}{s_2 - a_i s_1 - b_i} = \frac{1}{\sqrt{a_i^2+1}}$ so $d_i = \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}}$



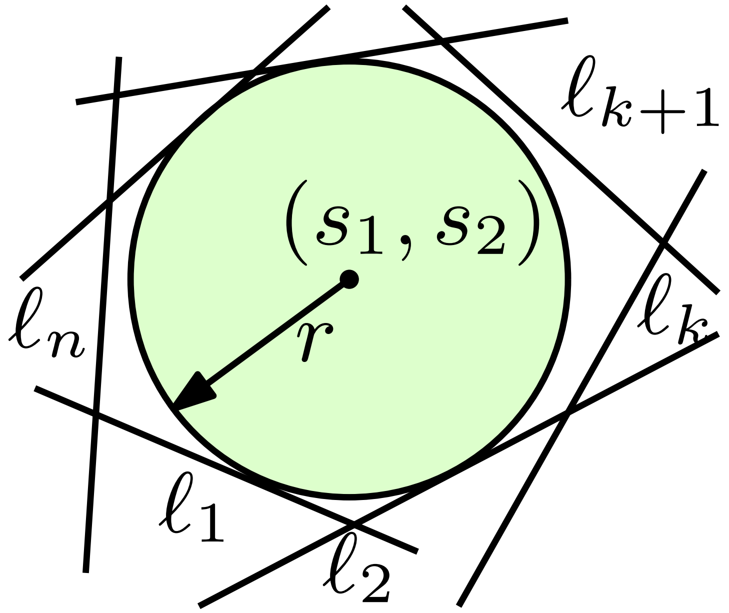
Example: Largest Disk in a Polygon

We get the following linear programm:

maximize r

subject to $\dfrac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}} \geq r$ for $i = 1, 2, ..., k$

$\dfrac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}} \leq -r$ for $i = k+1, k+2, ..., n$



There are three variables: s_1, s_2 and r.

An optimal solution yields the desired largest disk contained in P.

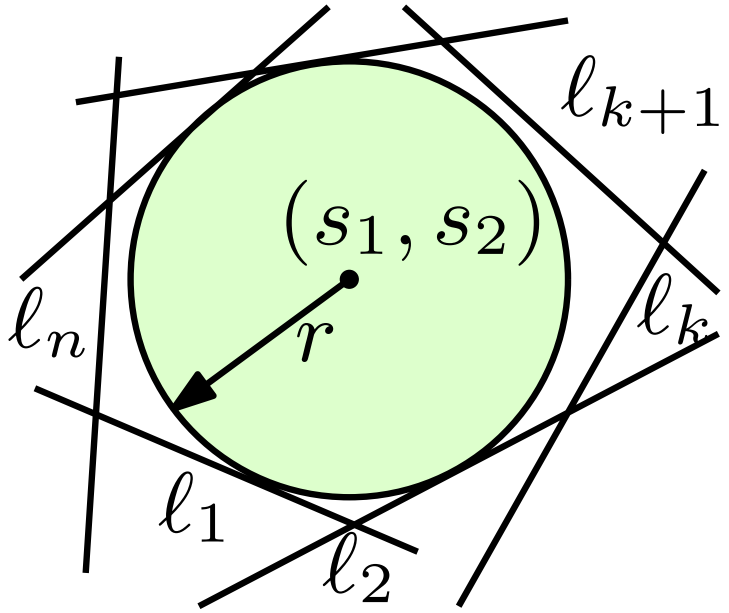
Example: Largest Disk in a Polygon

We get the following linear programm:

maximize r

subject to $\dfrac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \geq r$ for $i = 1, 2, ..., k$

$\dfrac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \leq -r$ for $i = k+1, k+2, ..., n$



There are three variables: s_1, s_2 and r.

An optimal solution yields the desired largest disk contained in P.

Also possible for $\mathbb{R}^n$ instead of $\mathbb{R}^2$.

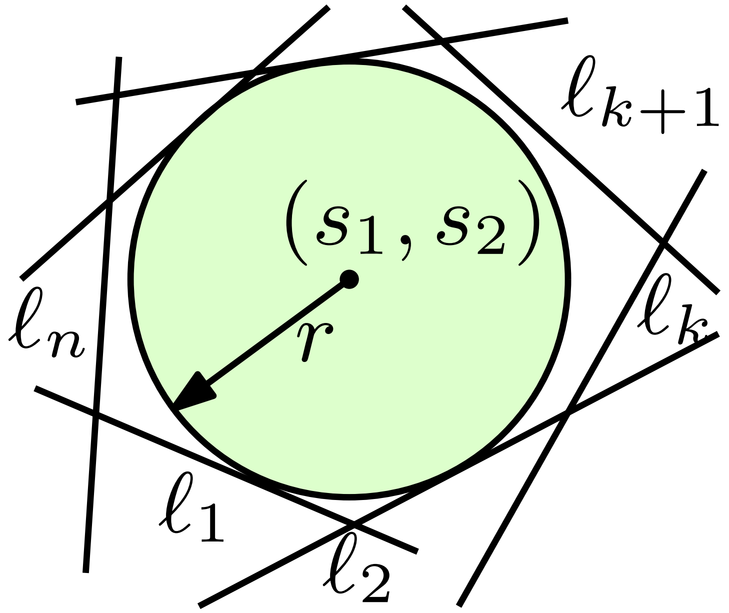
Example: Largest Disk in a Polygon

We get the following linear programm:

maximize r

$$\text{subject to } \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \geq r \text{ for } i = 1, 2, \ldots, k$$

$$\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2+1}} \leq -r \text{ for } i = k+1, k+2, \ldots, n$$



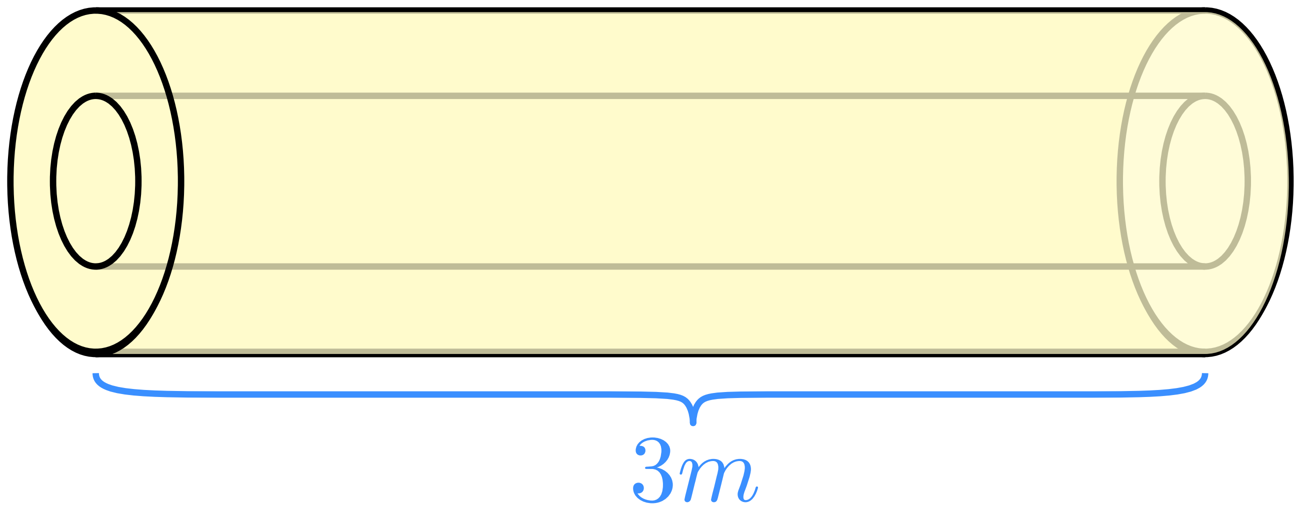
There are three variables: s_1, s_2 and r.
An optimal solution yields the desired largest disk contained in P.

Also possible for $\mathbb{R}^n$ instead of $\mathbb{R}^2$.

Note: Finding the smallest disk containing a polygon is not linear but is convex optimization problem.

Example: Cutting Paper Rolls

Paper mill makes 3 meter paper rolls.

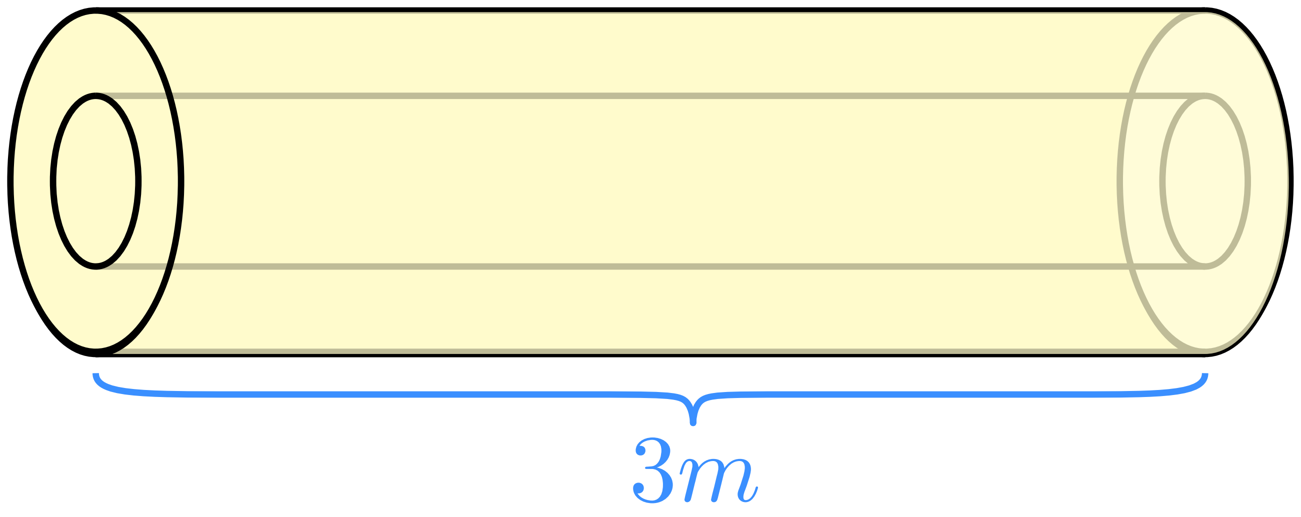


What's the fewest number of rolls need to satisfy an order of:

- 97 rolls width 135cm
- 610 rolls width 108cm
- 395 rolls width 93cm
- 211 rolls width 42cm

Example: Cutting Paper Rolls

Paper mill makes 3 meter paper rolls.



What's the fewest number of rolls need to satisfy an order of:

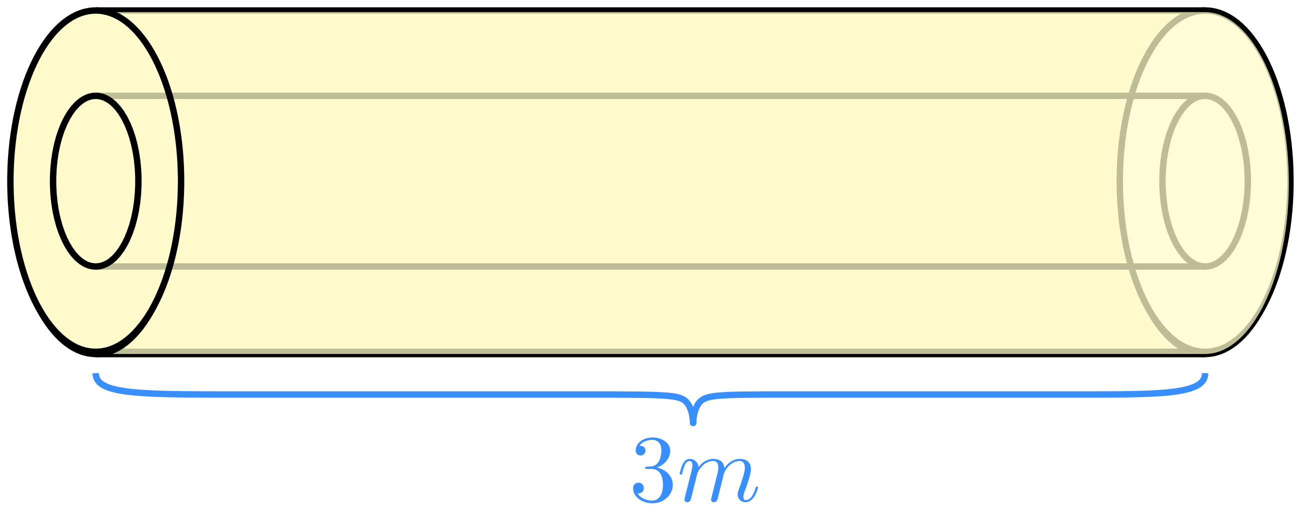
- 97 rolls width 135cm
- 610 rolls width 108cm
- 395 rolls width 93cm
- 211 rolls width 42cm

How to formulate this as an LP?

Example: Cutting Paper Rolls

What's the fewest number of rolls need to satisfy an order of:

- 97 rolls width 135cm
- 610 rolls width 108cm
- 395 rolls width 93cm
- 211 rolls width 42cm



Possible ways to cut roll with <42cm wasted:

P1: $2 \cdot 135$

P2: $135 + 108 + 42$

P3: $135 + 93 + 42$

P4: $135 + 3 \cdot 42$

P5: $2 \cdot 108 + 2 \cdot 42$

P6: $108 + 2 \cdot 93$

P7: $108 + 93 + 2 \cdot 42$

P8: $108 + 4 \cdot 42$

P9: $2 \cdot 93$

P10: $2 \cdot 93 + 2 \cdot 42$

P11: $93 + 4 \cdot 42$

P12: $7 \cdot 42$

can be generated by computer

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to $2x_1 + x_2 + x_3 + x_4 \geq 97$

$$x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610$$

$$x_3 + 2x_6 + x_7 + 3x_9 + 2x_10 + x_11 \geq 395$$

$$x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211$$

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to
$$2x_1 + x_2 + x_3 + x_4 \geq 97$$
$$x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610$$
$$x_3 + 2x_6 + x_7 + 3x_9 + 2x_10 + x_11 \geq 395$$
$$x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211$$

Optimal solution:

$x_1 = 48.5$, $x_5 = 206.25$, $x_6 = 197.5$, all others zero.

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to $2x_1 + x_2 + x_3 + x_4 \geq 97$

$x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610$

$x_3 + 2x_6 + x_7 + 3x_9 + 2x_10 + x_11 \geq 395$

$x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211$

Optimal solution:

$x_1 = 48.5$, $x_5 = 206.25$, $x_6 = 197.5$, all others zero.

What if we want an integer solution?

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

$$\text{minimize } \sum_{j=1}^{12} x_j \quad \text{(total \# of rolls cut)}$$

$$\begin{aligned}
\text{subject to } & 2x_1 + x_2 + x_3 + x_4 \geq 97 \\
& x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610 \\
& x_3 + 2x_6 + x_7 + 3x_9 + 2x_10 + x_11 \geq 395 \\
& x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211
\end{aligned}$$

Optimal solution:

$x_1 = 48.5$, $x_5 = 206.25$, $x_6 = 197.5$, all others zero.

What if we want an integer solution?

rounding gives good solution, but better solution exists

Example: Cutting Paper Rolls

For each possibility P_j, add a variable $x_j \geq 0$ representing # rolls cut that way.

minimize $\sum_{j=1}^{12} x_j$ (total # of rolls cut)

subject to $2x_1 + x_2 + x_3 + x_4 \geq 97$

$$x_2 + 2x_5 + x_6 + x_7 + x_8 \geq 610$$

$$x_3 + 2x_6 + x_7 + 3x_9 + 2x_10 + x_11 \geq 395$$

$$x_2 + x_3 + 3x_4 + 2x_5 + 2x_7 + 4x_8 + 2x_{10} + 4x_{11} + 7x_{12} \geq 211$$

Optimal solution:

$x_1 = 48.5$, $x_5 = 206.25$, $x_6 = 197.5$, all others zero.

What if we want an integer solution?

rounding gives good solution, but better solution exists

next lecture: integer linear programming

Summary

A linear program (LP) is the problem of maximizing a given linear function over the set of all vectors that satisfy a given system of linear equations and inequalities.

Any LP can easily be transformed to the form

$$\text{maximize } c^T x \text{ subject to } Ax \leq b$$

Summary

A linear program (LP) is the problem of maximizing a given linear function over the set of all vectors that satisfy a given system of linear equations and inequalities.

Any LP can easily be transformed to the form

$$\text{maximize } c^T x \text{ subject to } Ax \leq b$$

LPs are efficiently solvable, both in theory and in practice.

Summary

A linear program (LP) is the problem of maximizing a given linear function over the set of all vectors that satisfy a given system of linear equations and inequalities.

Any LP can easily be transformed to the form

$$\text{maximize } c^T x \text{ subject to } Ax \leq b$$

LPs are efficiently solvable, both in theory and in practice.

Modelling problems as LPs:

- many problems have natural LP formulation (e.g. diet problem)
- important algorithmic problems, e.g., network flow
- some non-linear concepts can be handled by additional variables, e.g., absolute value
- geometric problems: e.g., separation problems, fitting disk in polygon
- encoding many possibilities by variables (paper cutting);
  integer variables ?!

Attributions

- book cover: "Understanding and Using Linear Programming" by Matoušek and Gärtner
- image sources (history): de.wikipedia.org, news.standford.de
- image source (burger): https://www.publicdomainpictures.net/en/view-image.php?image=408431\&picture=fast-food-food-snack
- image source (shadows): "Understanding and Using Linear Programming"