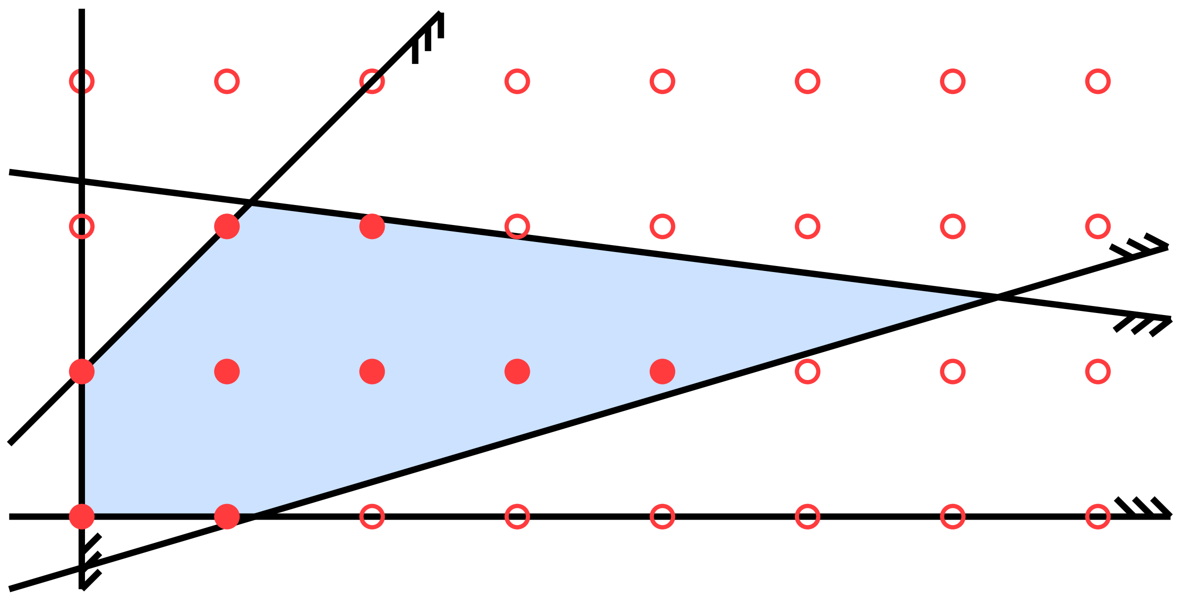
Linear Programming

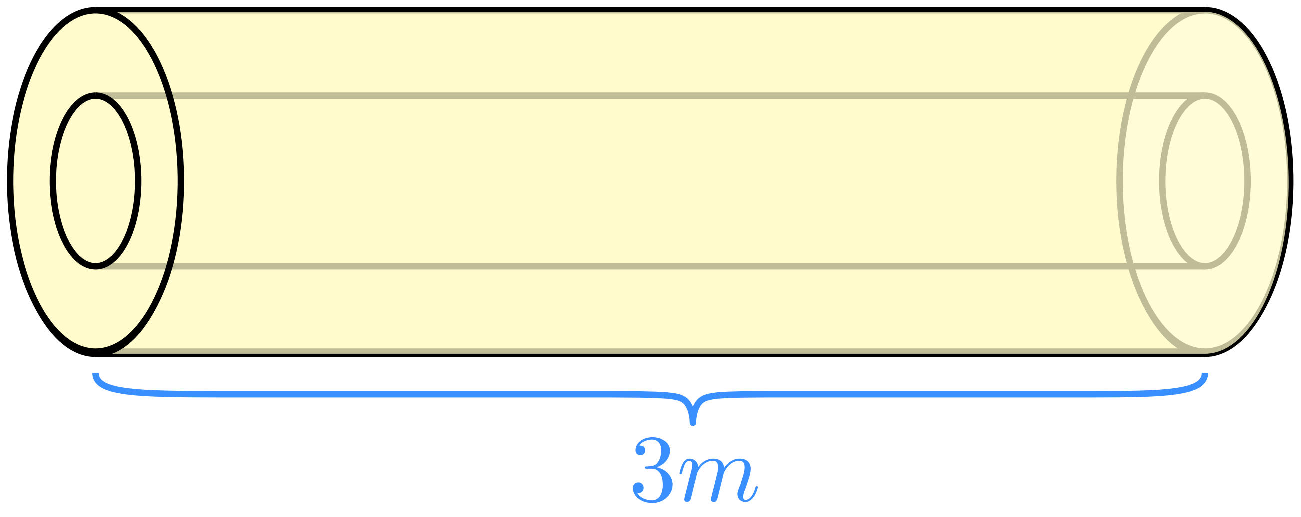
Integer Linear Programming



Recall 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: $3 \cdot 93$

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

P11: $93 + 4 \cdot 42$

P12: $7 \cdot 42$

Example: Cutting Paper Rolls

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

minimize $\displaystyle\sum_{j=1}^{12} x_j$ (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.

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

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

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

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?

We can round up to $49, 207, 198$, i.e. a total of 454.

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?

We can round up to $49, 207, 198$, i.e. a total of 454.

Not bad, since an optimal integral solution is at least 453.

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.

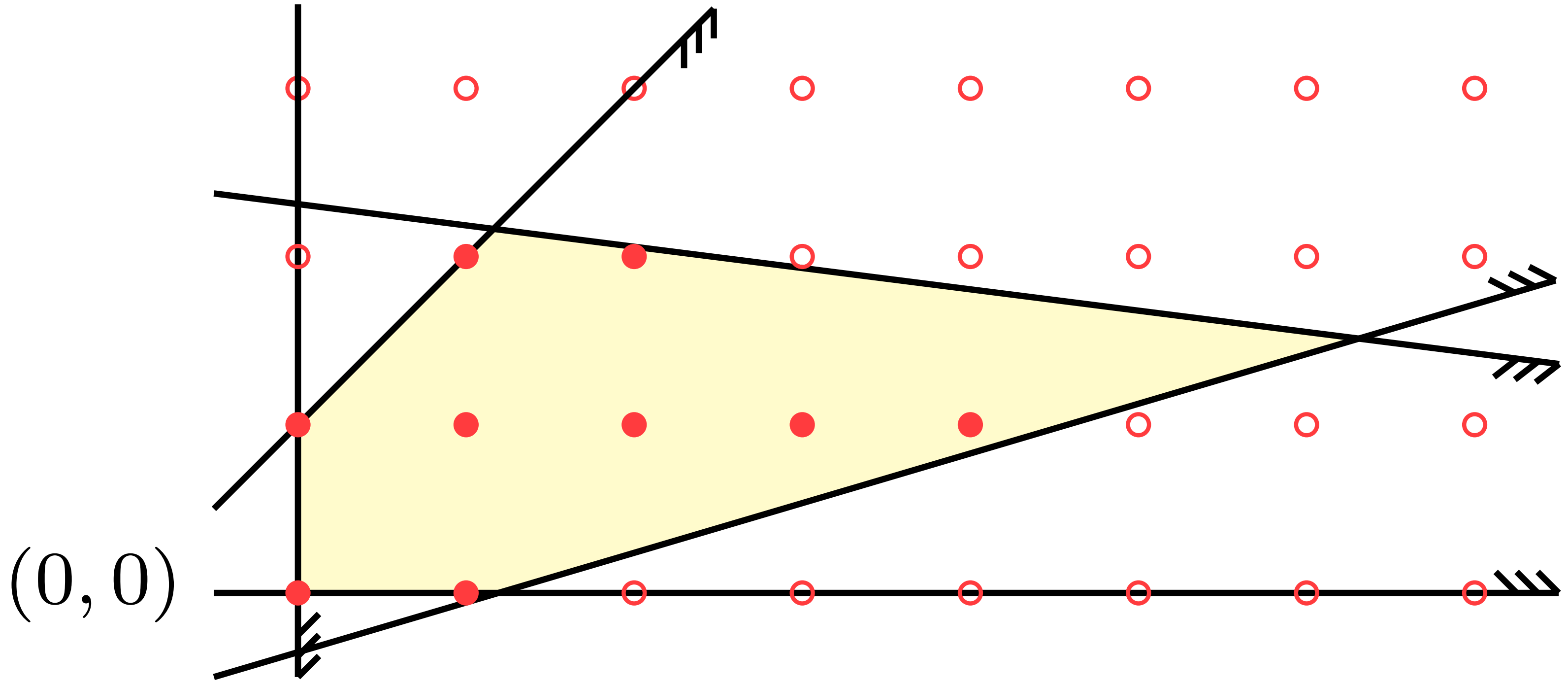
What if we want an integer solution?

We can round up to $49, 207, 198$, i.e. a total of 454.

Not bad, since an optimal integral solution is at least 453.

Here, $48, 206, 197, 1$ (for x_9) is an optimal integral solution.

Integer Linear Programming

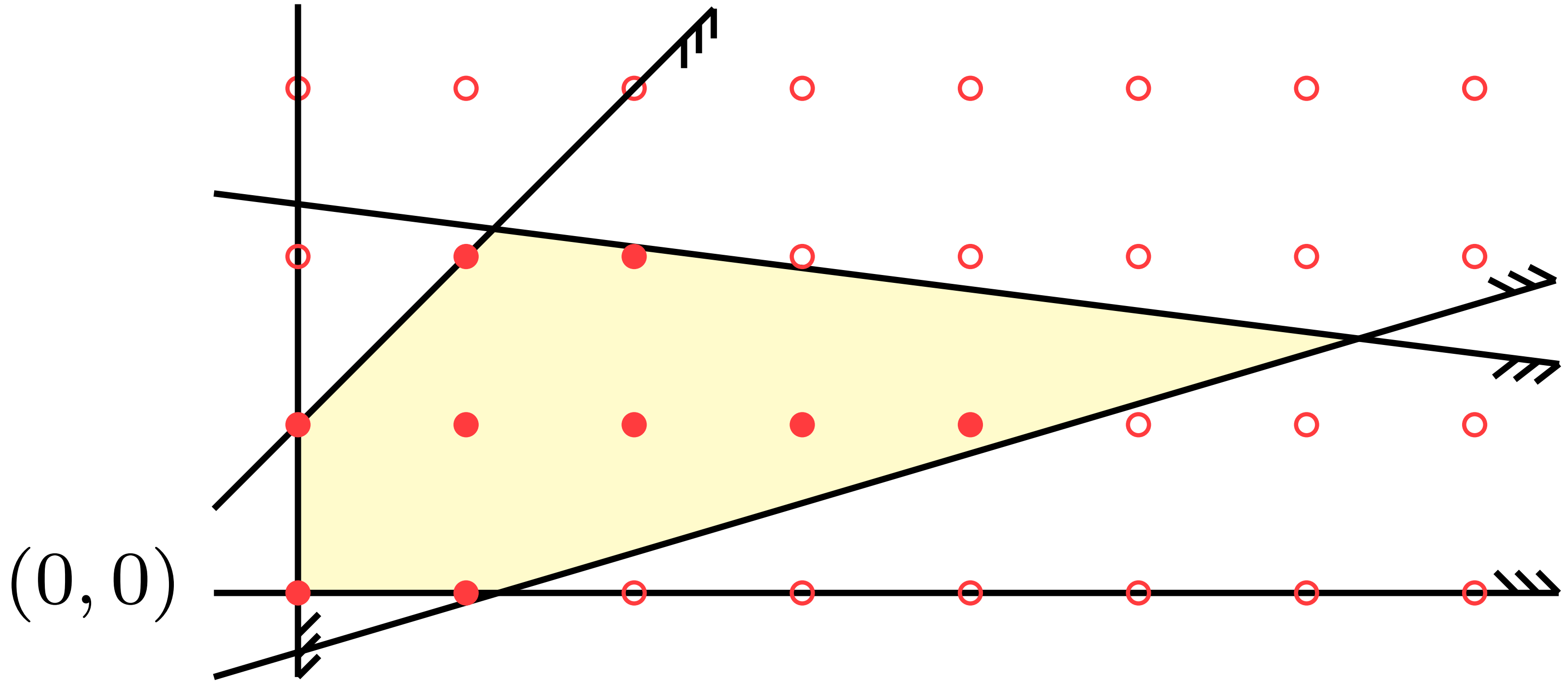


maximize $c^T x$

subject to $Ax \leq b$

$x \in \mathbb{Z}^n$

Integer Linear Programming



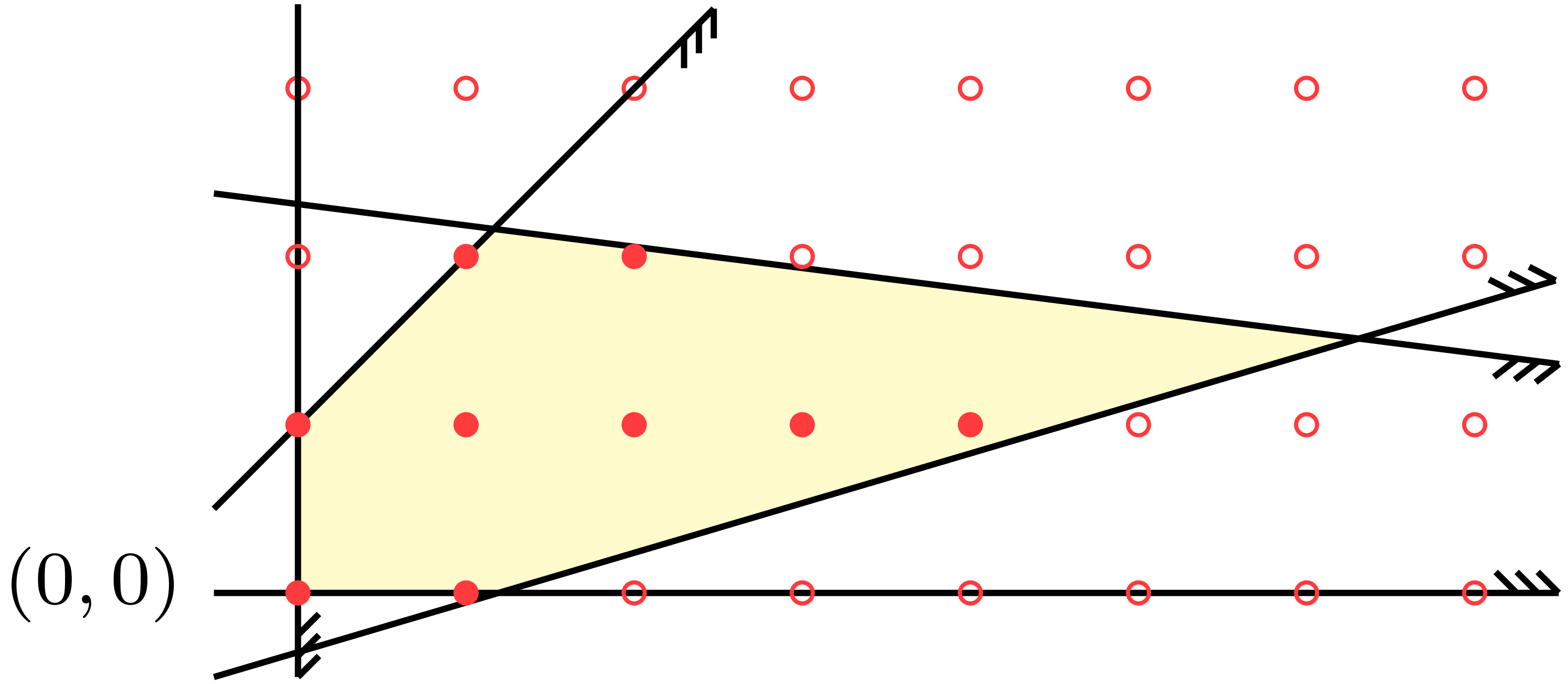
maximize $c^T x$

subject to $Ax \leq b$

$x \in \mathbb{Z}^n$

Integer Linear Programming (ILP) is NP-hard.

Integer Linear Programming



maximize $c^T x$

subject to $Ax \leq b$

$x \in \mathbb{Z}^n$

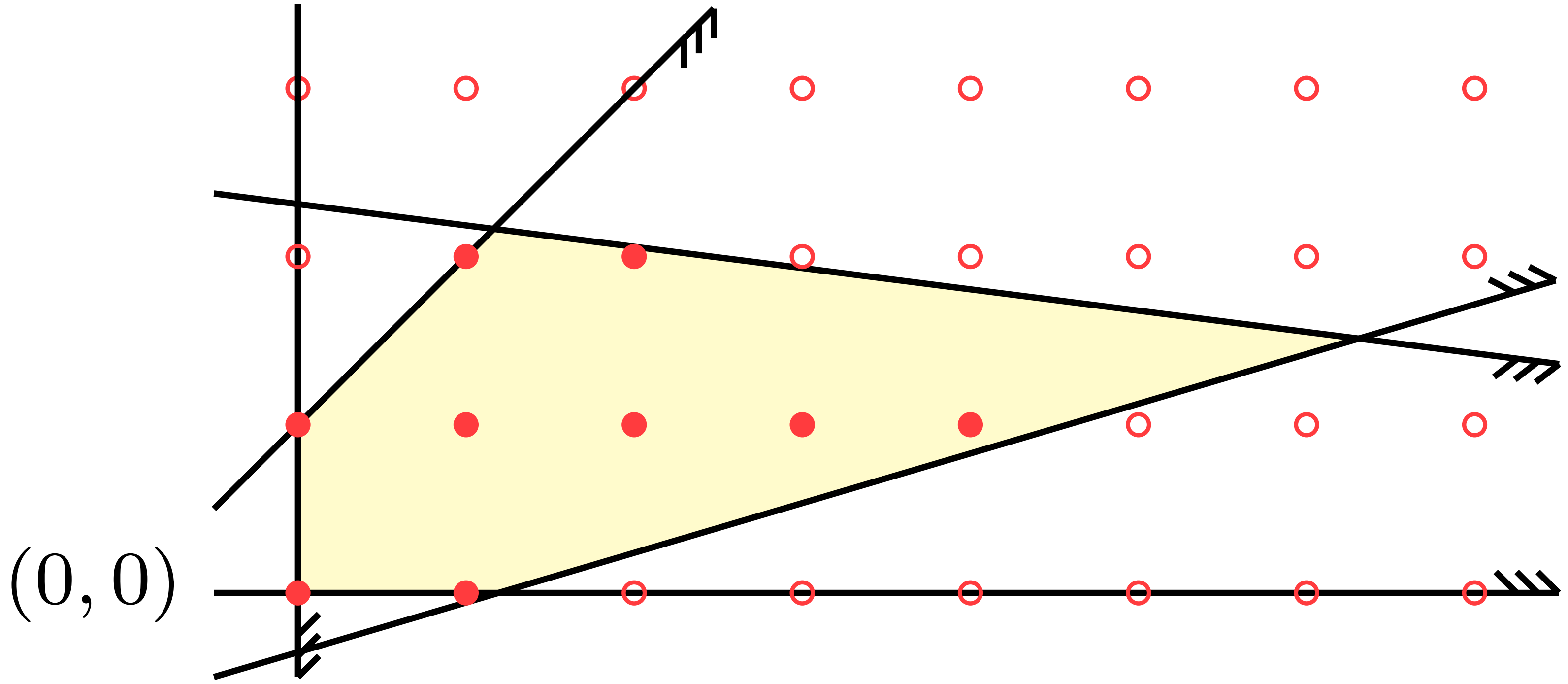
Integer Linear Programming (ILP) is NP-hard.

Variants:

0/1-Linear Programming: $x \in \{0, 1\}^n$,

Mixed Integer Linear Programming: some variables in $\mathbb{Z}$, some in $\mathbb{R}$.

Integer Linear Programming



maximize $c^T x$

subject to $Ax \leq b$

$x \in \mathbb{Z}^n$

Integer Linear Programming (ILP) is NP-hard.

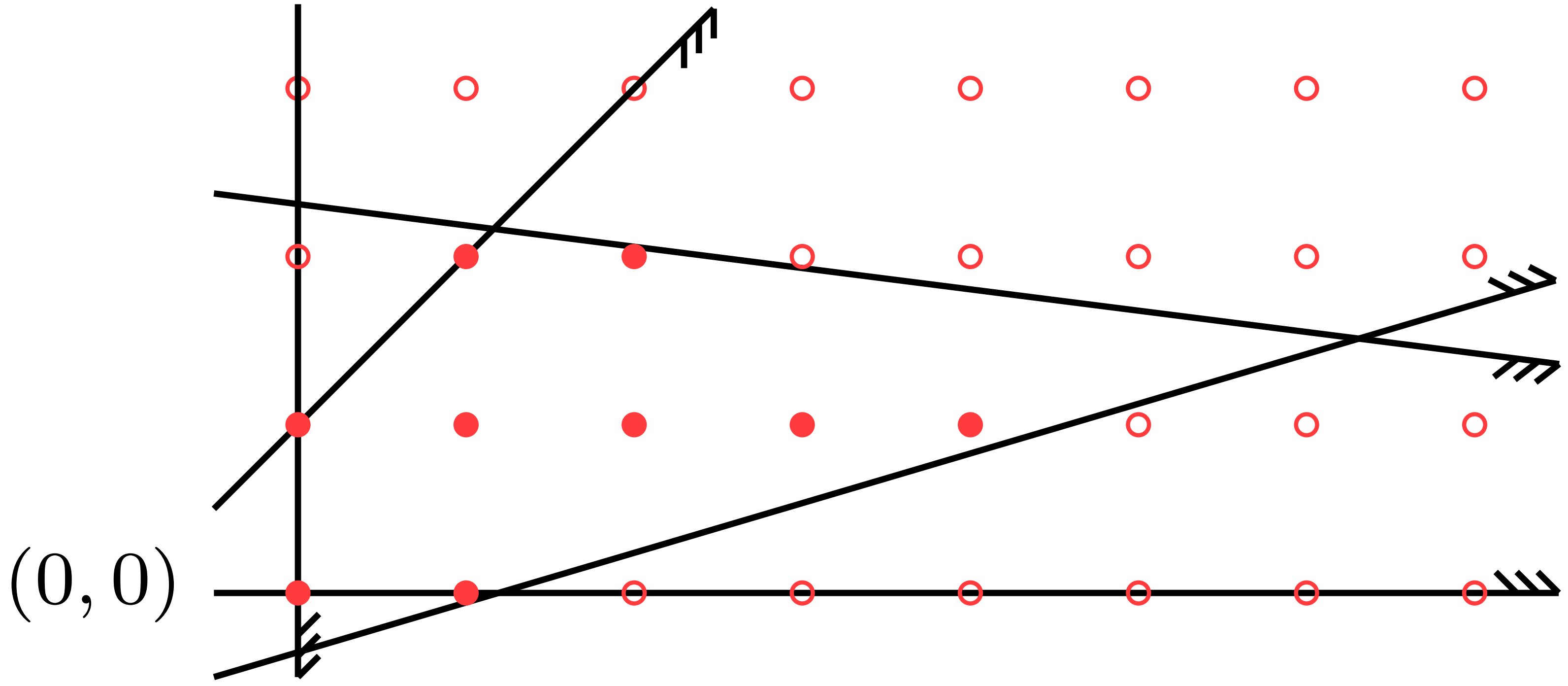
Variants:

0/1-Linear Programming: $x \in \{0, 1\}^n$,

Mixed Integer Linear Programming: some variables in $\mathbb{Z}$, some in $\mathbb{R}$.

ILP and variants most wide-spread use of LPs.

Integer Linear Programming



maximize $c^T x$

subject to $Ax \leq b$

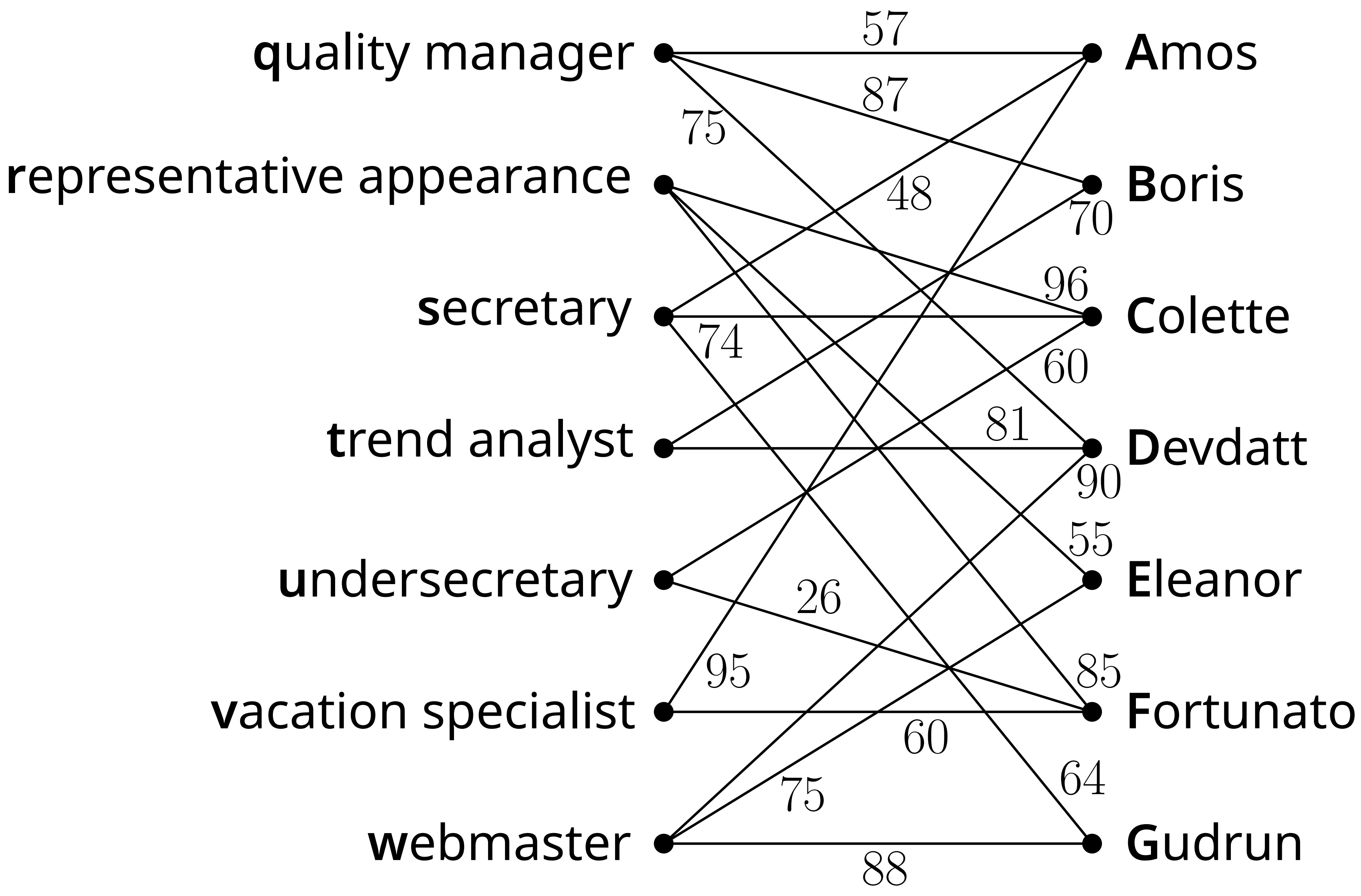
$x \in \mathbb{Z}^n$

Next: Example “easy, medium, hard” integer programs.

- Maximum weight matching
- Minimum vertex cover
- Maximum independent set

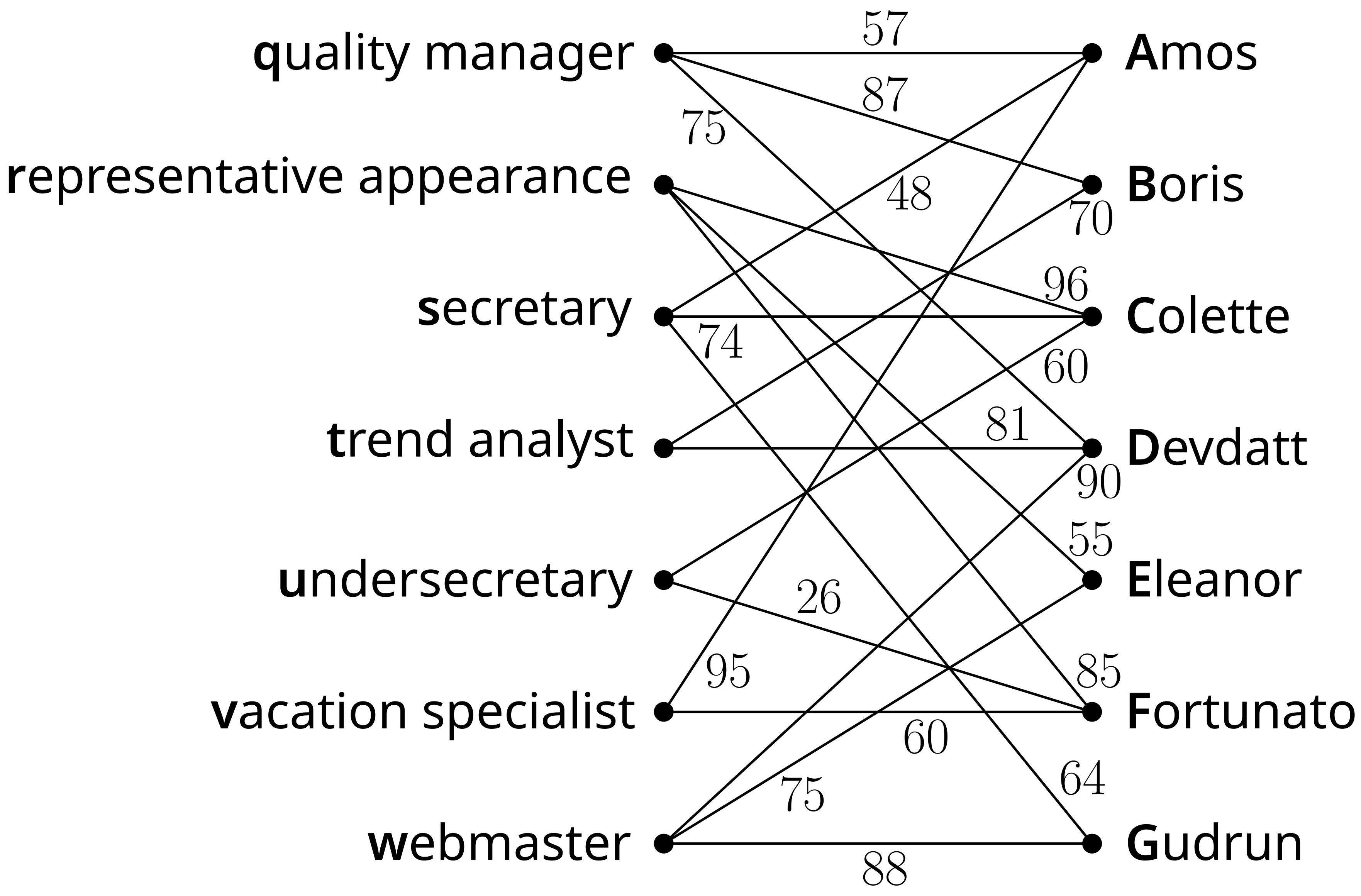
Maximum weight perfect matching

A company is assigning workers to jobs. In the bipartite graph an edge connects a worker to a job they are willing to take, weighted by their suitability for the job.



Maximum weight perfect matching

A company is assigning workers to jobs. In the bipartite graph an edge connects a worker to a job they are willing to take, weighted by their suitability for the job.

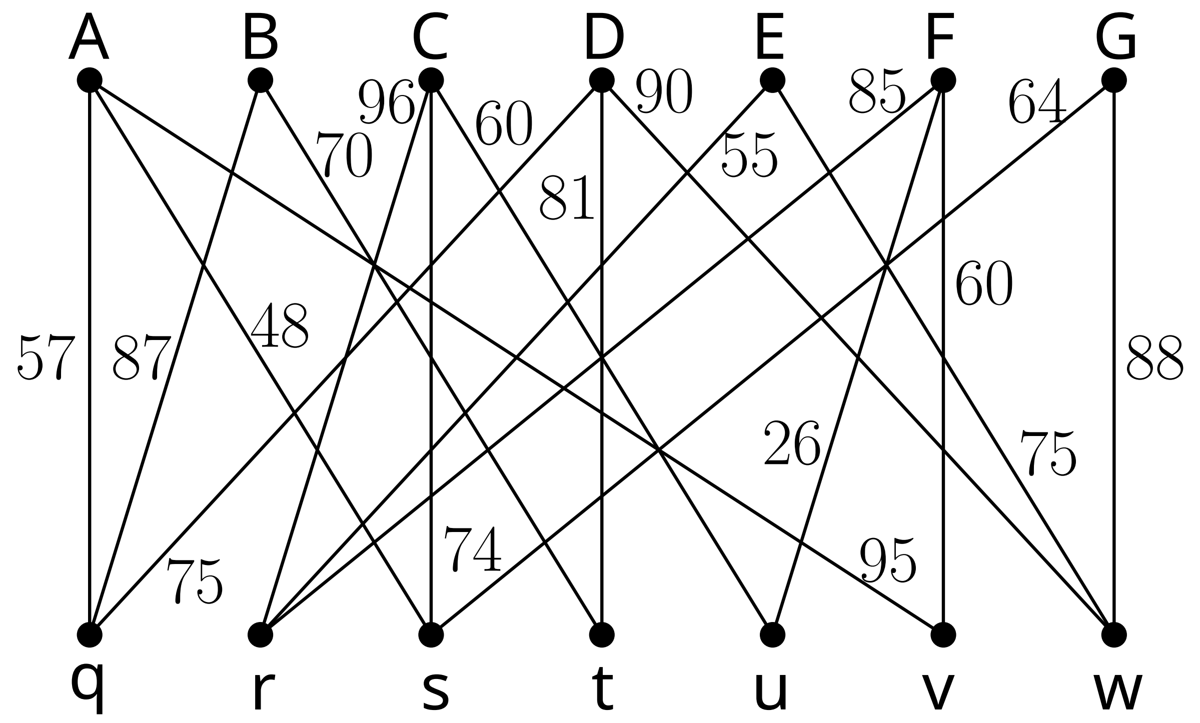


How to find an optimal assigment?

Maximum weight perfect matching

Greedy approach

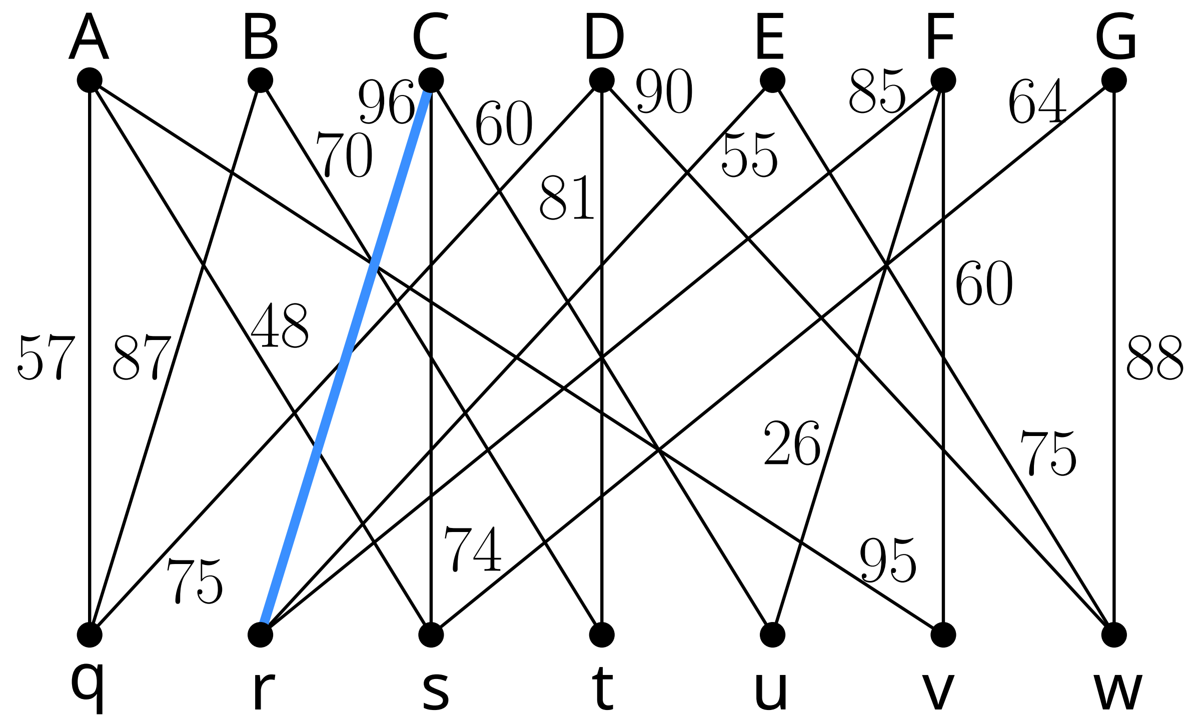
always assign largest possible score



Maximum weight perfect matching

Greedy approach

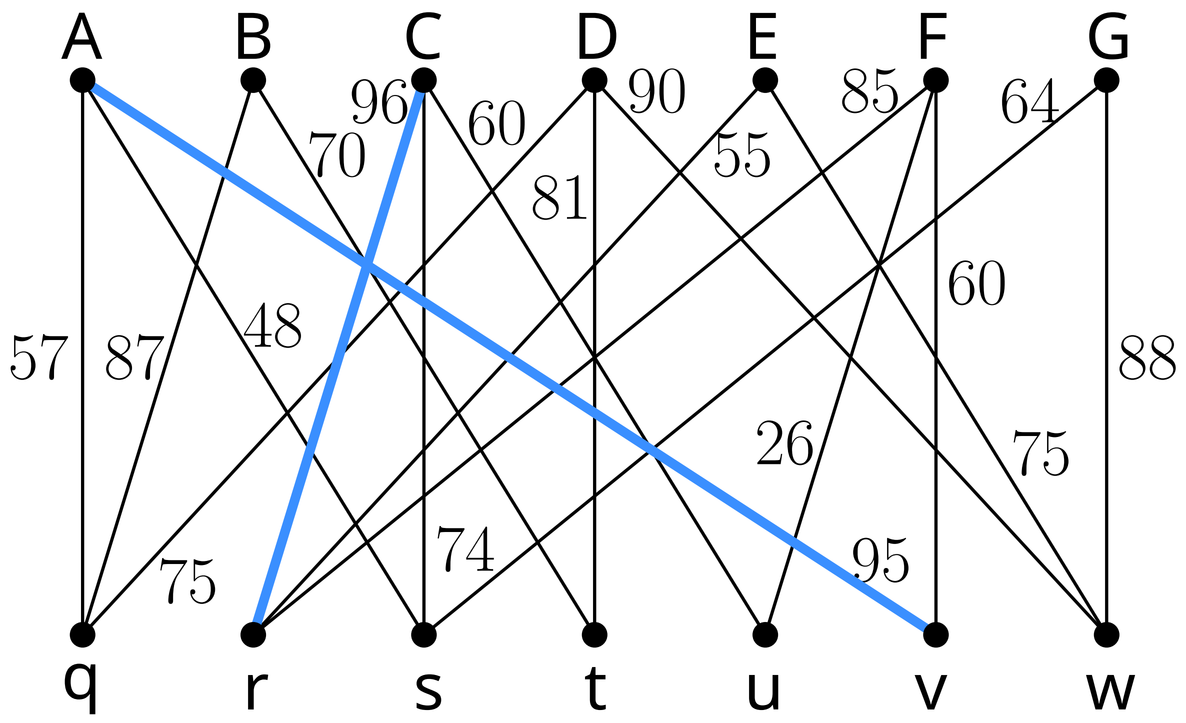
always assign largest possible score



Maximum weight perfect matching

Greedy approach

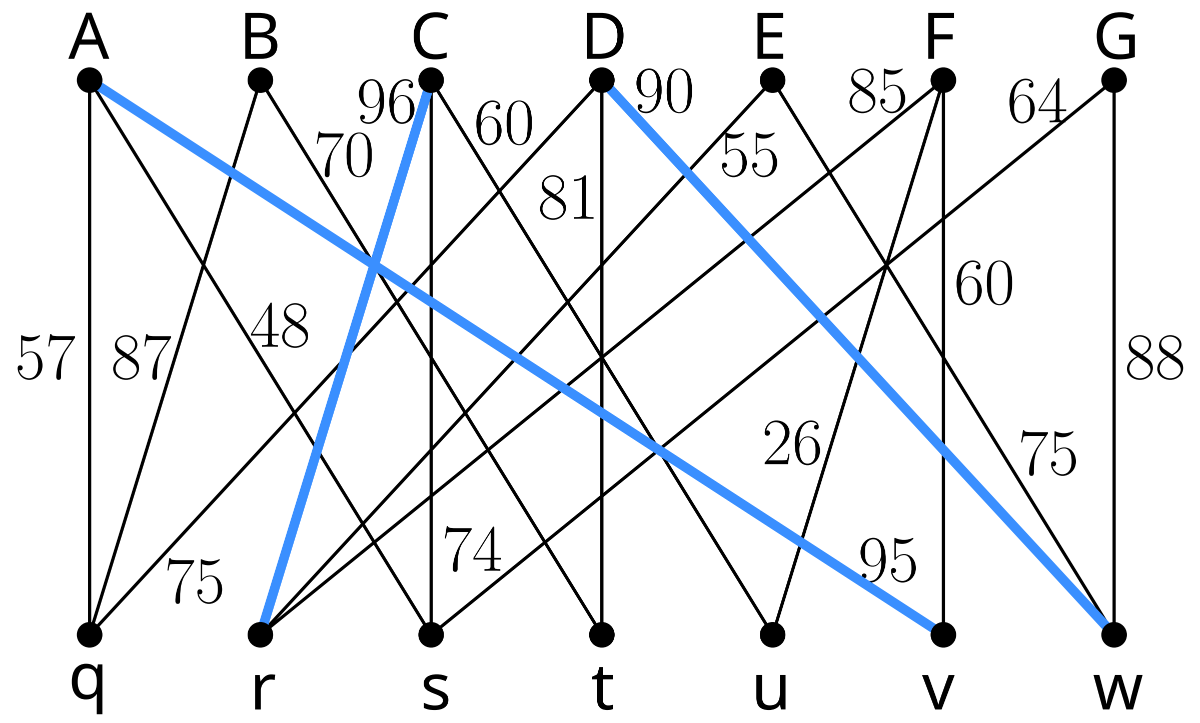
always assign largest possible score



Maximum weight perfect matching

Greedy approach

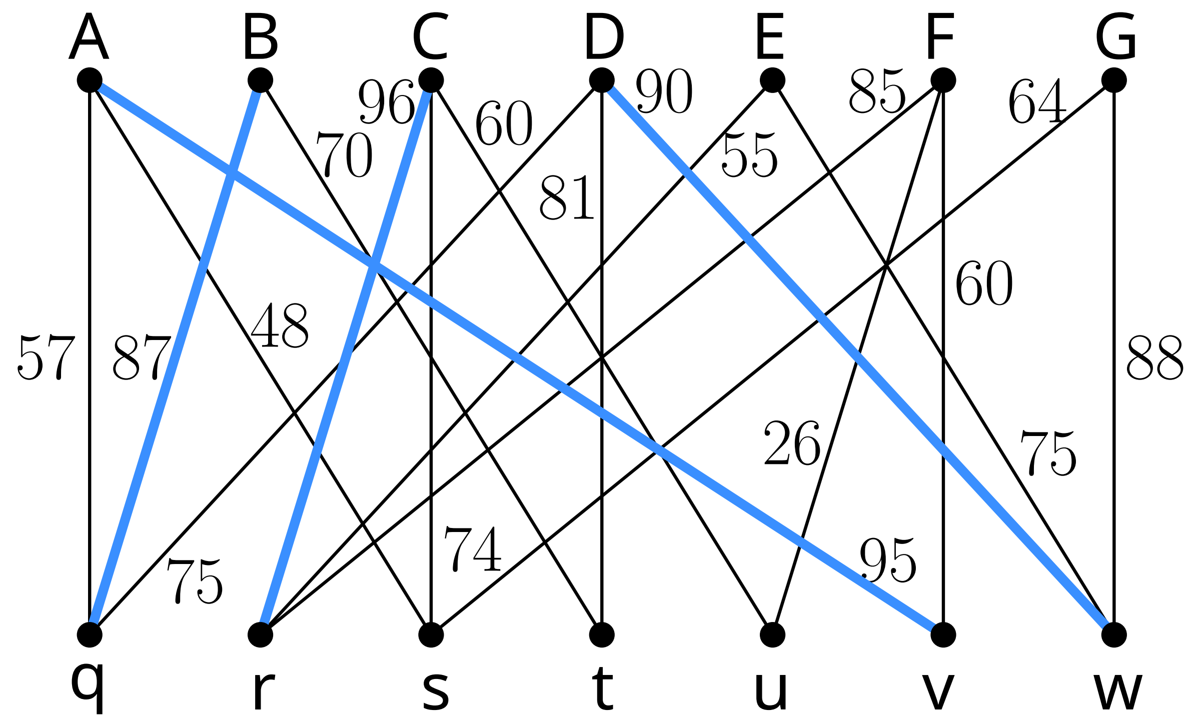
always assign largest possible score



Maximum weight perfect matching

Greedy approach

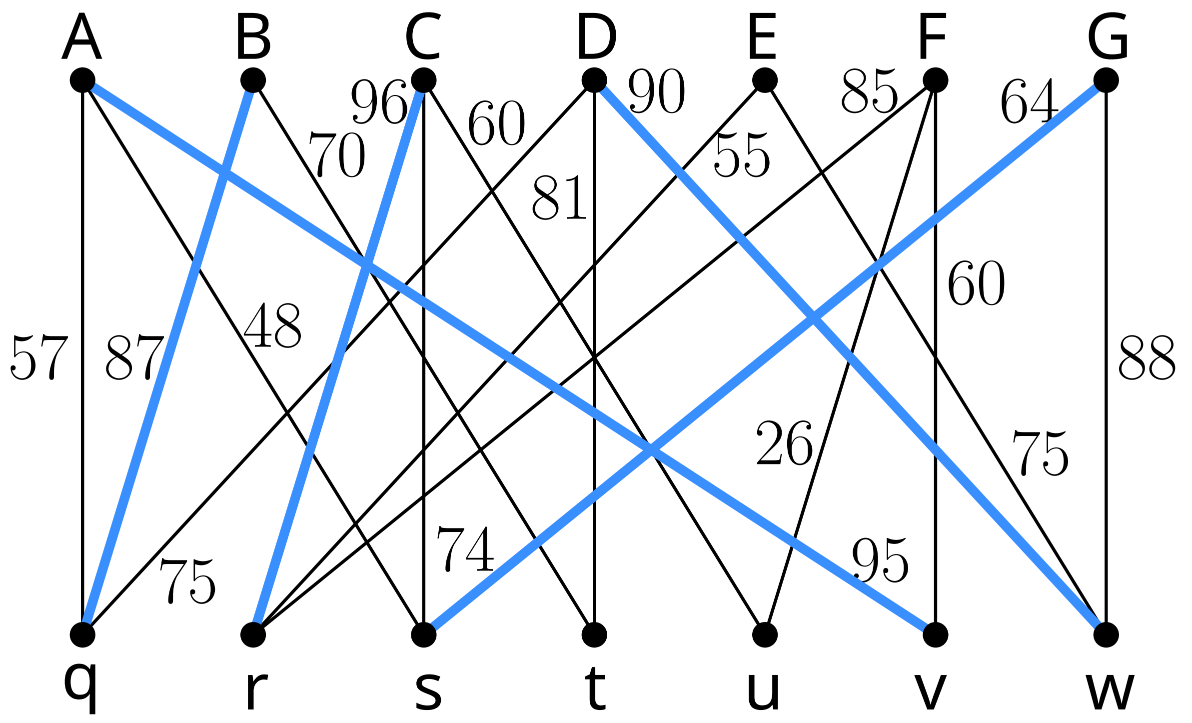
always assign largest possible score



Maximum weight perfect matching

Greedy approach

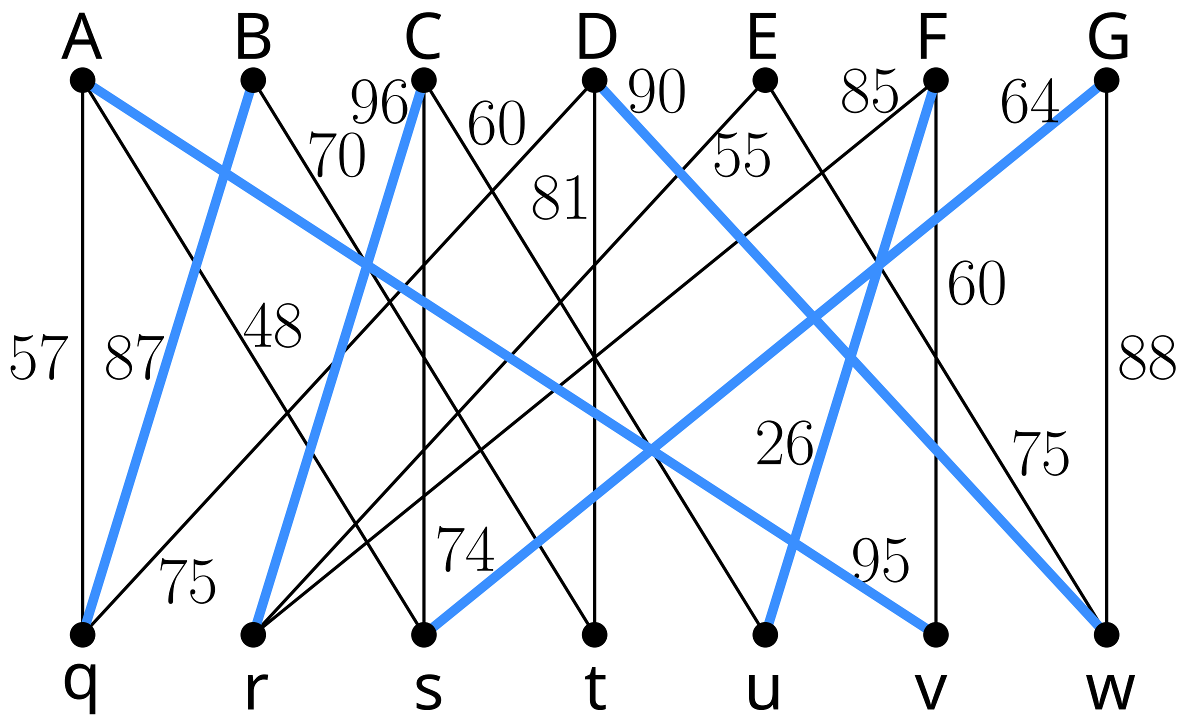
always assign largest possible score



Maximum weight perfect matching

Greedy approach

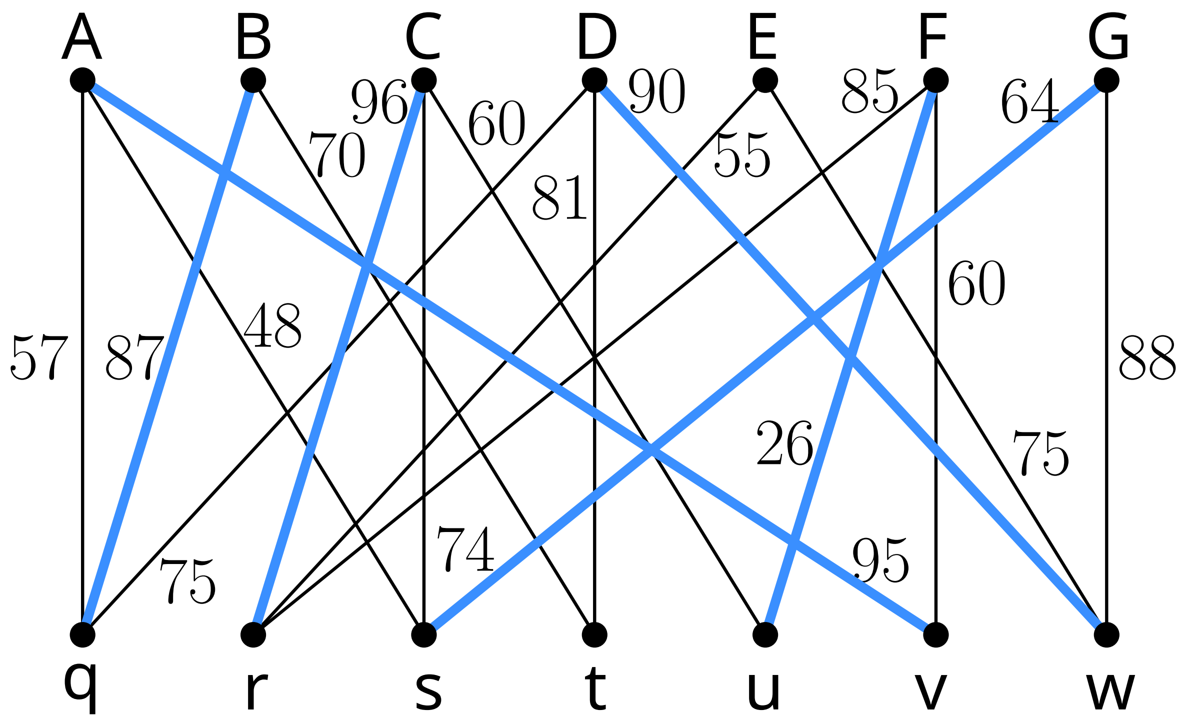
always assign largest possible score



Maximum weight perfect matching

Greedy approach

always assign largest possible score



total score:

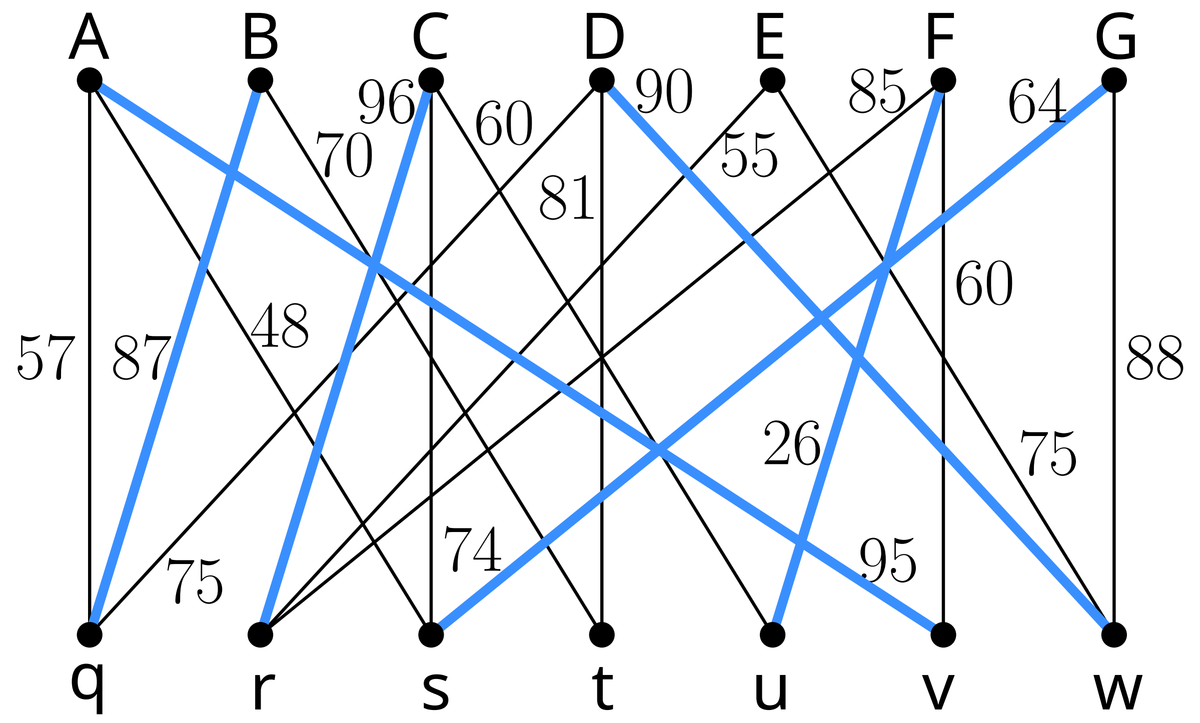
$$95 + 87 + 96 + 90 + 0 + 26 + 64 = 458$$

Eleanor is not assigned to any job!

Maximum weight perfect matching

Greedy approach

always assign largest possible score

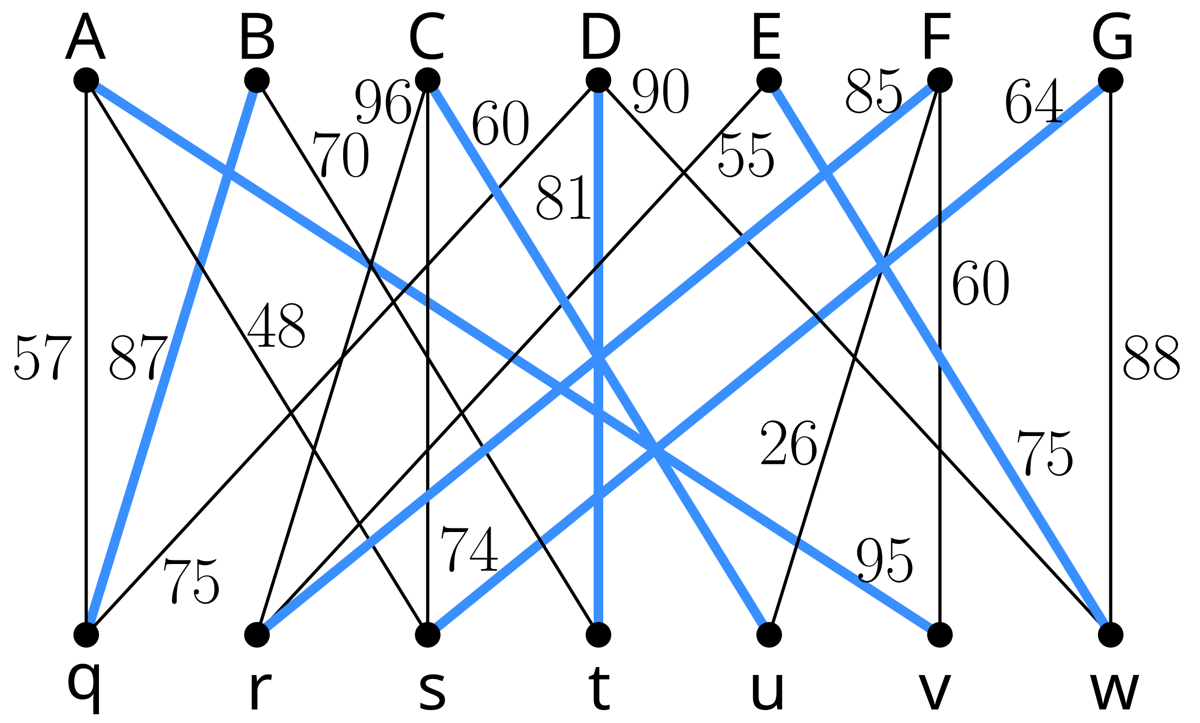


total score:

$$95 + 87 + 96 + 90 + 0 + 26 + 64 = 458$$

Eleanor is not assigned to any job!

Optimal solution



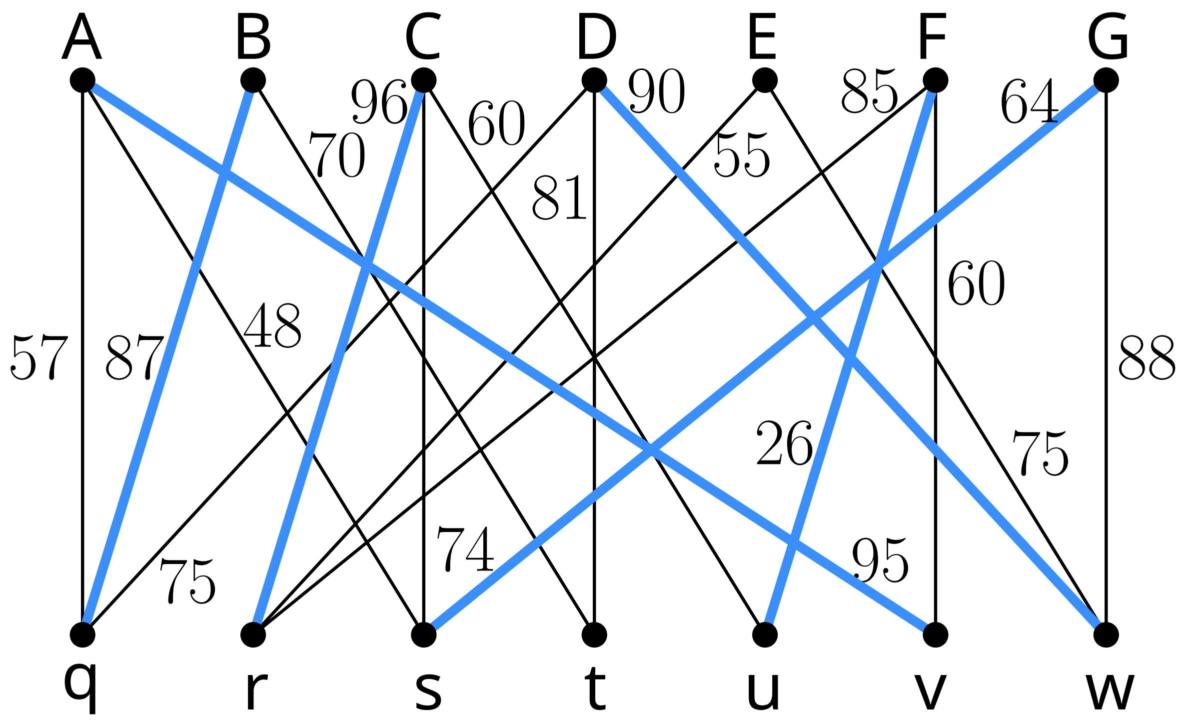
total score:

$$95 + 87 + 60 + 81 + 75 + 85 + 64 = 547$$

Maximum weight perfect matching

Greedy approach

always assign largest possible score

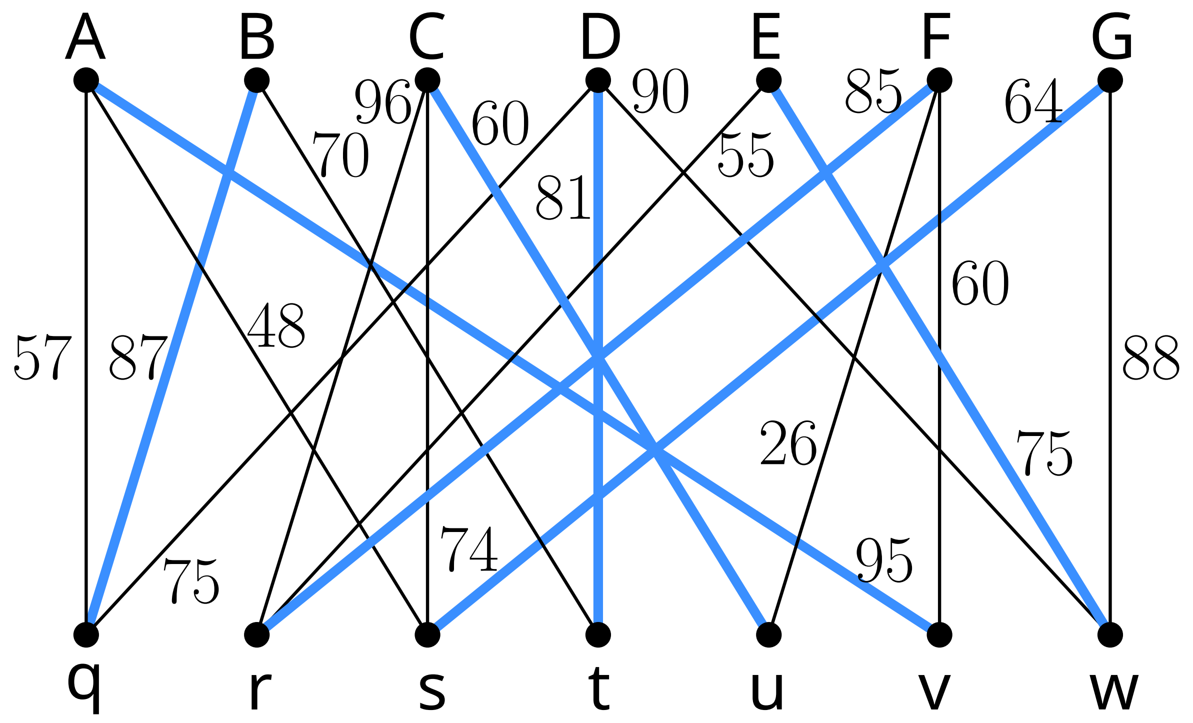


total score:

$$95 + 87 + 96 + 90 + 0 + 26 + 64 = 458$$

Eleanor is not assigned to any job!

Optimal solution



total score:

$$95 + 87 + 60 + 81 + 75 + 85 + 64 = 547$$

How to model this as an ILP?

Maximum weight perfect matching

Integer program

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$x_e \in \{0, 1\}$ for each edge $e \in E$.

variable x_e encodes whether edge e is contained in the matching

Maximum weight perfect matching

Integer program

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$x_e \in \{0, 1\}$ for each edge $e \in E$.

variable x_e encodes whether edge e is contained in the matching

if we "relax" the requirement that the x_e are integral we get . . .

Maximum weight perfect matching

Integer program

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} w_e x_e \\
\text{subject to} \quad & \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V \text{, and} \\
& x_e \in \{0, 1\} \text{ for each edge } e \in E.
\end{aligned}$$

variable x_e encodes whether edge e is contained in the matching

LP relaxation

$$\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} w_e x_e \\
\text{subject to} \quad & \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V \text{, and} \\
& 0 \leq x_e \leq 1 \text{ for each edge } e \in E.
\end{aligned}$$

Maximum weight perfect matching

Integer program

$$\text{maximize} \sum_{e \in E} w_e x_e$$

$$\text{subject to} \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V\text{, and}$$

$$x_e \in \{0, 1\} \text{ for each edge } e \in E.$$

LP relaxation

$$\text{maximize} \sum_{e \in E} w_e x_e$$

$$\text{subject to} \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V\text{, and}$$

$$0 \leq x_e \leq 1 \text{ for each edge } e \in E.$$

What do we learn from solving the LP-relaxation?

Maximum weight perfect matching

Integer program

$$\text{maximize} \sum_{e \in E} w_e x_e$$

$$\text{subject to} \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V \text{, and}$$

$$x_e \in \{0, 1\} \text{ for each edge } e \in E.$$

LP relaxation

$$\text{maximize} \sum_{e \in E} w_e x_e$$

$$\text{subject to} \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V \text{, and}$$

$$0 \leq x_e \leq 1 \text{ for each edge } e \in E.$$

What do we learn from solving the LP-relaxation?

If the LP is infeasible so is the IP.

Maximum weight perfect matching

Integer program

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$x_e \in \{0, 1\}$ for each edge $e \in E$.

LP relaxation

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$0 \leq x_e \leq 1$ for each edge $e \in E$.

What do we learn from solving the LP-relaxation?

An optimal solution to the LP relaxation is a bound on an optimal solution to the IP.

Maximum weight perfect matching

Integer program

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$x_e \in \{0, 1\}$ for each edge $e \in E$.

LP relaxation

maximize $\sum_{e \in E} w_e x_e$

subject to $\sum_{e \in E : v \in e} x_e = 1$ for each vertex $v \in V$, and

$0 \leq x_e \leq 1$ for each edge $e \in E$.

What do we learn from solving the LP-relaxation?

An optimal solution to the LP relaxation is a bound on an optimal solution to the IP.

(How) can we get an optimal integral solution from a fractional one?

Maximum weight perfect matching

Integer program

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V\text{, and} \\ & x_e \in \{0, 1\} \text{ for each edge } e \in E. \end{aligned}$$

LP relaxation

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} w_e x_e \\ \text{subject to} \quad & \sum_{e \in E : v \in e} x_e = 1 \text{ for each vertex } v \in V\text{, and} \\ & 0 \leq x_e \leq 1 \text{ for each edge } e \in E. \end{aligned}$$

What do we learn from solving the LP-relaxation?

An optimal solution to the LP relaxation is a bound on an optimal solution to the IP.

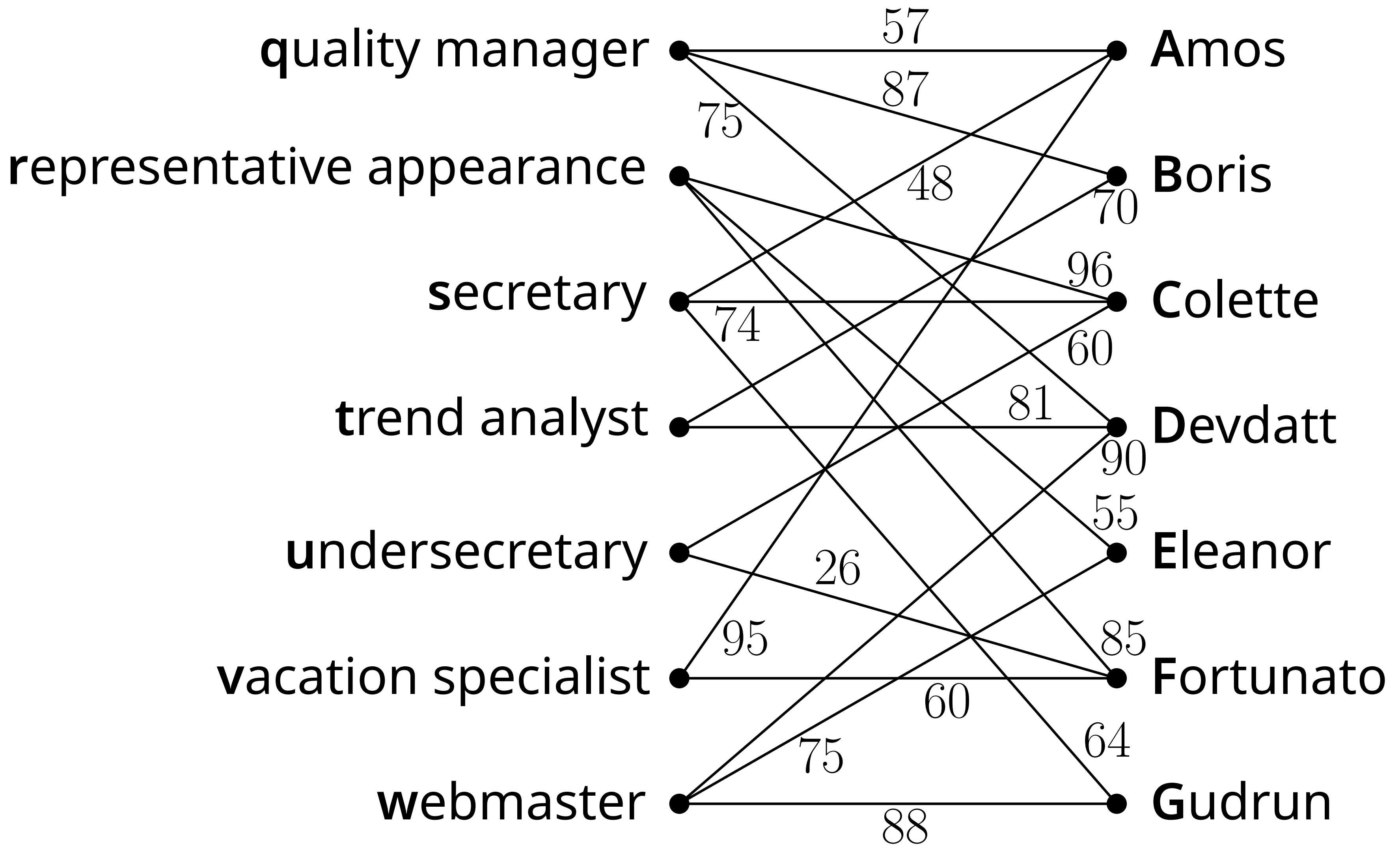
(How) can we get an optimal integral solution from a fractional one?

Here this works out nicely.

Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x_e^*$ its value and $k(x^*)$ its number of non-integral components.

Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x^*_e$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) = 0$, then we are done.

Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x_e^*$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Maximum weight perfect matching

Theorem 3.2.1

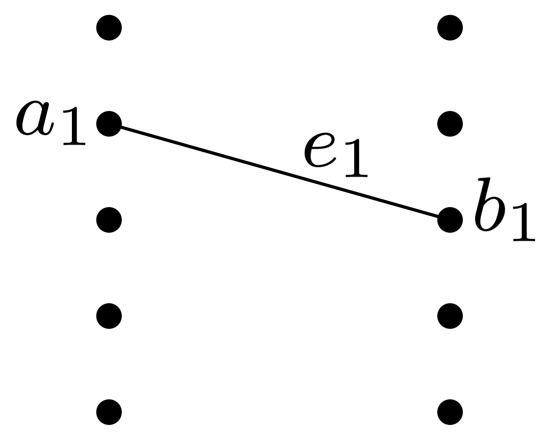
Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x_e^*$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x_{e_1}^*$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

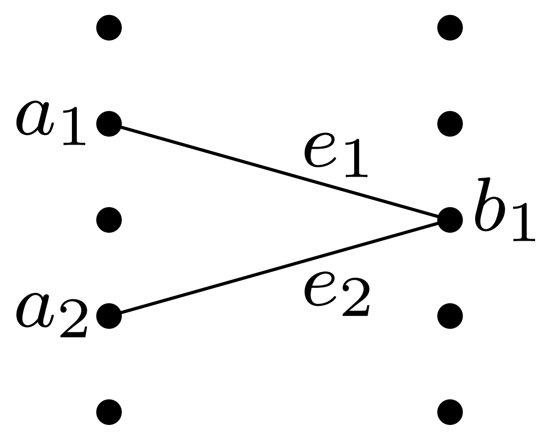
Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e\in E} w_e x^*_e$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x^*_{e_1}$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.

Since $0 < x^*_{e_1} < 1$ and $\sum_{b_1 \in e} x^*_e = 1$ there exists $e_2 = (a_2, b_1) \neq e_1$ s.t. $x^*_{e_2}$ non-integral.



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

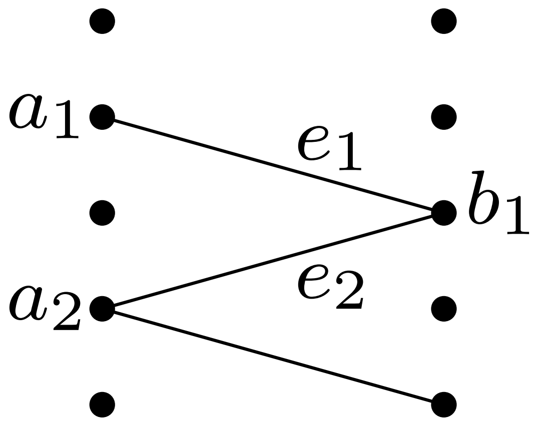
Let $w(x^*) = \sum_{e \in E} w_e x^*_e$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x^*_{e_1}$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.

Since $0 < x^*_{e_1} < 1$ and $\sum_{b_1 \in e} x^*_e = 1$ there exists $e_2 = (a_2, b_1) \neq e_1$ s.t. $x^*_{e_2}$ non-integral.

Similarly, there exists edge e_3 and so on, until we find a cycle.



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

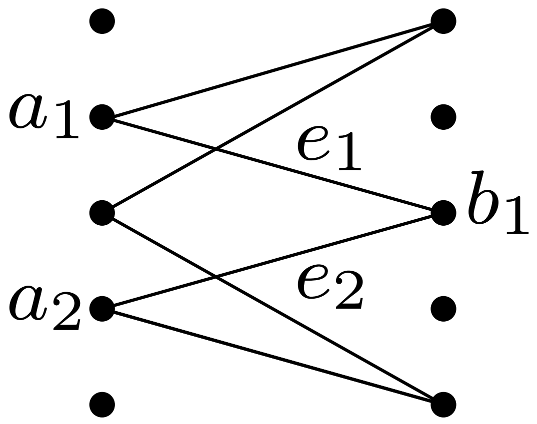
Let $w(x^*) = \sum_{e \in E} w_e x_e^*$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x_{e_1}^*$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.

Since $0 < x_{e_1}^* < 1$ and $\sum_{b_1 \in e} x_e^* = 1$ there exists $e_2 = (a_2, b_1) \neq e_1$ s.t. $x_{e_2}^*$ non-integral.

Similarly, there exists edge e_3 and so on, until we find a cycle.



Maximum weight perfect matching

Theorem 3.2.1
Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x^*_e$ its value and $k(x^*)$ its number of non-integral components.

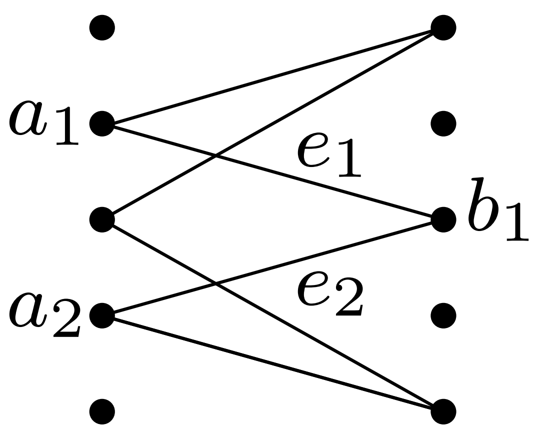
If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x^*_{e_1}$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.

Since $0 < x^*_{e_1} < 1$ and $\sum_{b_1 \in e} x^*_e = 1$ there exists $e_2 = (a_2, b_1) \neq e_1$ s.t. $x^*_{e_2}$ non-integral.

Similarly, there exists edge e_3 and so on, until we find a cycle.

Since G is bipartite, the cycle – say $e_1, \ldots, e_t$ – has even length t.



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof: Let x^* be an optimal solution of the LP relaxation.

Let $w(x^*) = \sum_{e \in E} w_e x_e^*$ its value and $k(x^*)$ its number of non-integral components.

If $k(x^*) > 0$, we find an optimal solution $\tilde{x}$ with $k(x^*) > k(\tilde{x}^*)$.

Let $x_{e_1}^*$ be a non-integral component of x for edge $e_1 = (a_1, b_1)$.

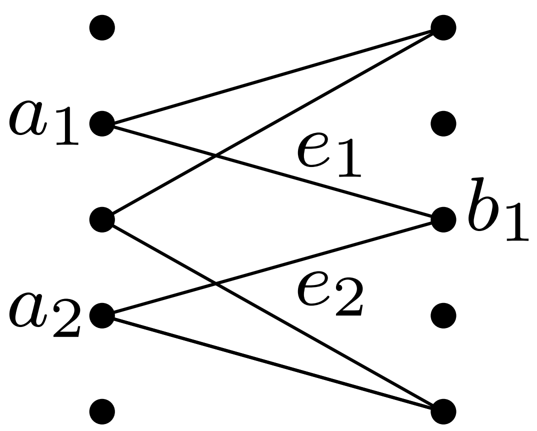
Since $0 < x_{e_1}^* < 1$ and $\sum_{b_1 \in e} x_e^* = 1$ there exists $e_2 = (a_2, b_1) \neq e_1$ s.t. $x_{e_2}^*$ non-integral.

Similarly, there exists edge e_3 and so on, until we find a cycle.

Since G is bipartite, the cycle – say $e_1, \ldots, e_t$ – has even length t.

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

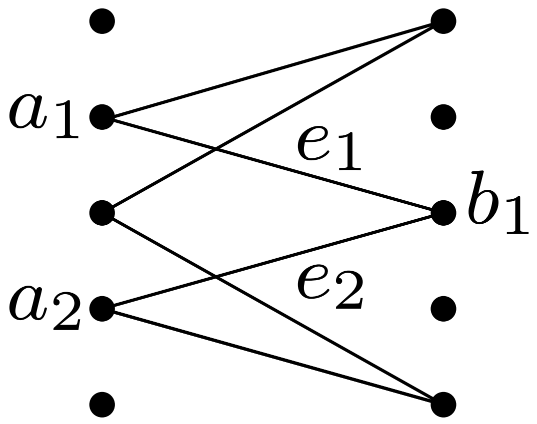
Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

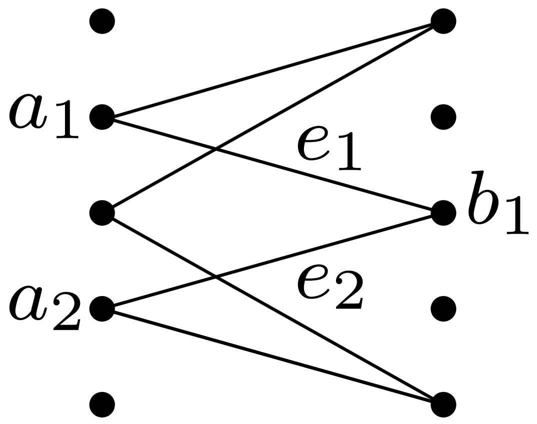
Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

For sufficiently small ε also $0 \le \tilde{x}_e \le 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \dots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \dots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1
Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

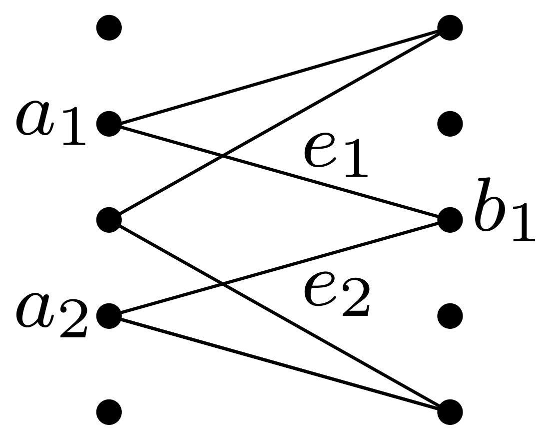
It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

What about $w(\tilde{x})$?

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

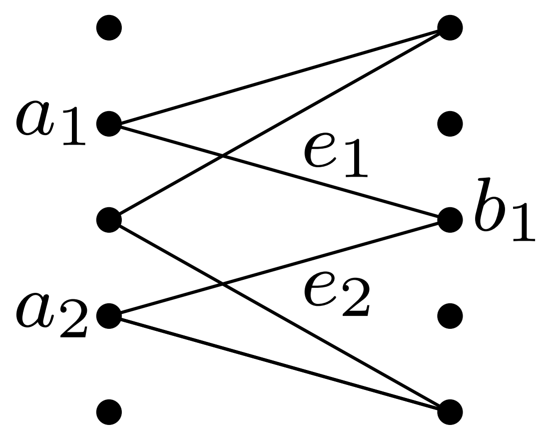
It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

We have $w(\tilde{x}) = \sum_{e\in E} w_e\tilde{x}_e = w(x^*) + \varepsilon \underbrace{\sum_{i=1}^{t} (-1)^i w_{e_i}}_{=:\,\Delta} = w(x^*) + \varepsilon\Delta$

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

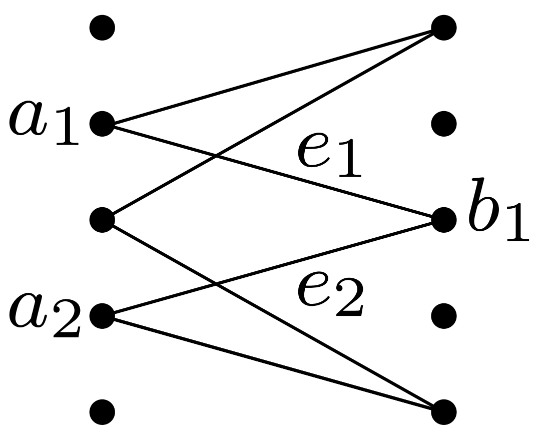
For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

We have $w(\tilde{x}) = \sum_{e\in E} w_e \tilde{x}_e = w(x^*) + \varepsilon \underbrace{\sum_{i=1}^{t} (-1)^i w_{e_i}}_{=:\,\Delta} = w(x^*) + \varepsilon\Delta$

$=: \Delta = 0$, else optimality of x^* violated

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

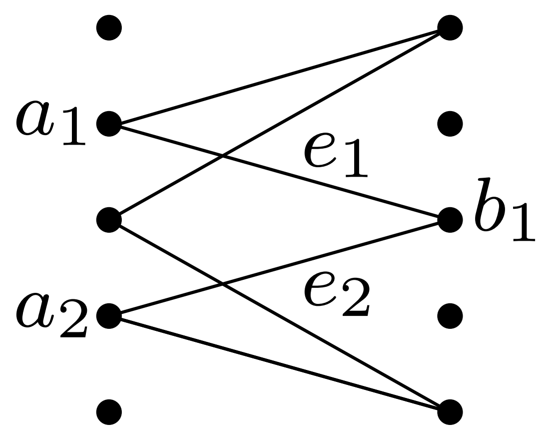
For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

We have $w(\tilde{x}) = \sum_{e\in E} w_e\tilde{x}_e = w(x^*) + \varepsilon \underbrace{\sum_{i=1}^{t} (-1)^i w_{e_i}}_{=:\,\Delta} = w(x^*) + \varepsilon\Delta = w(x^*)$

$=: \Delta = 0$, else optimality of x^* violated

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

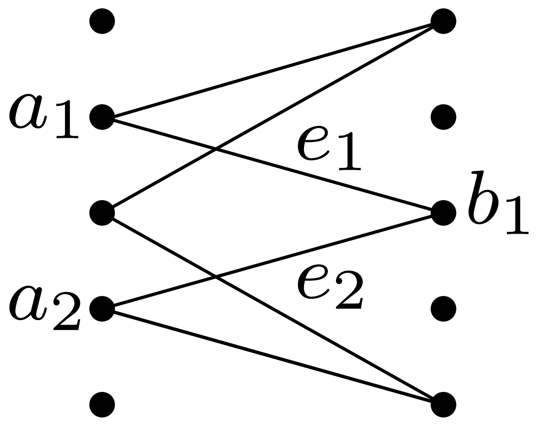
For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

We have $w(\tilde{x}) = \sum_{e\in E} w_e\tilde{x}_e = w(x^*) + \varepsilon \sum_{i=1}^{t}(-1)^i w_{e_i} = w(x^*) + \varepsilon\Delta = w(x^*)$

How do we choose ε?

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

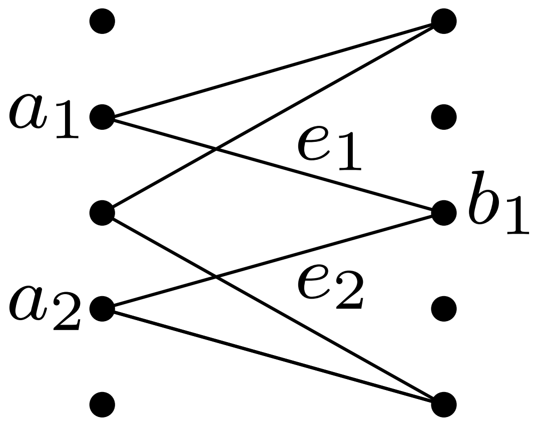
For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

We have $w(\tilde{x}) = \sum_{e\in E} w_e\tilde{x}_e = w(x^*) + \varepsilon \sum_{i=1}^{t} (-1)^i w_{e_i} = w(x^*) + \varepsilon\Delta = w(x^*)$

We choose ε largest s.t. $\tilde{x}$ is still feasible. Then $k(\tilde{x}) < k(x^*)$.

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \dots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \dots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Proof:

It is easy to see that $\tilde{x}$ still satisfies $\sum_{e:v\in e} \tilde{x}_e = 1$ for all $v \in V$.

For sufficiently small ε also $0 \leq \tilde{x}_e \leq 1$ for all $e \in E$, hence $\tilde{x}$ is feasible.

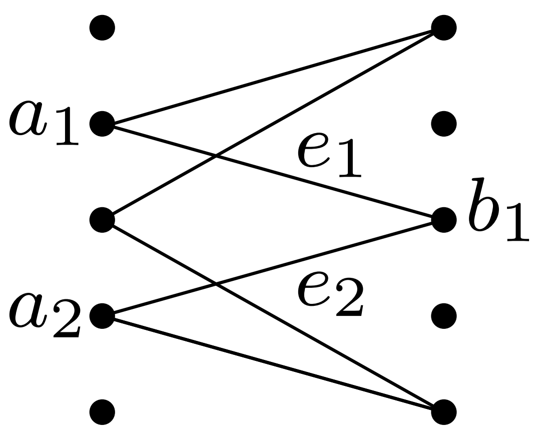
We have $w(\tilde{x}) = \sum_{e\in E} w_e \tilde{x}_e = w(x^*) + \varepsilon \sum_{i=1}^{t} (-1)^i w_{e_i} = w(x^*) + \varepsilon\Delta = w(x^*)$

We choose ε largest s.t. $\tilde{x}$ is still feasible. Then $k(\tilde{x}) < k(x^*)$.

We continue this procedure until all components are integral.

For small $\varepsilon > 0$ we define

$$\tilde{x}_e = \begin{cases} x_e^* - \varepsilon & \text{for } e \in \{e_1, \ldots, e_{t-1}\} \\ x_e^* + \varepsilon & \text{for } e \in \{e_2, \ldots, e_t\} \\ x_e^* & \text{else} \end{cases}$$



Maximum weight perfect matching

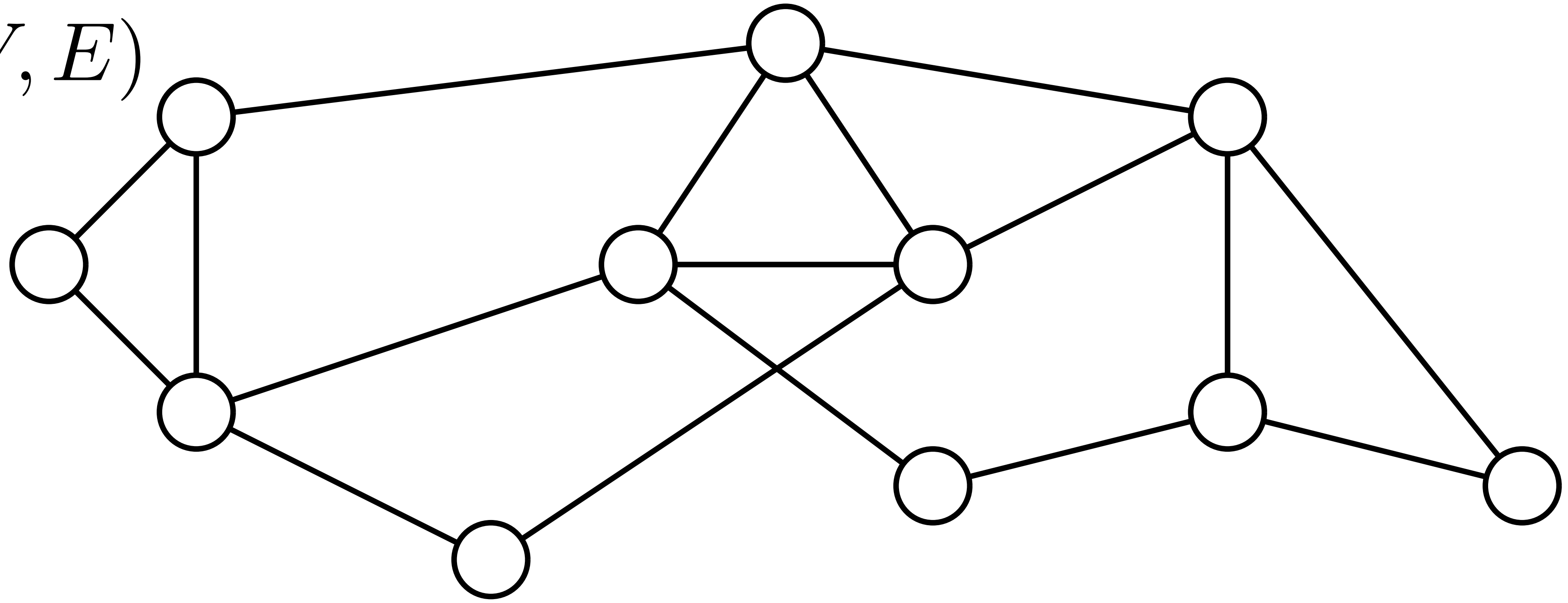
Theorem 3.2.1

Let G be an arbitrary weighted bipartite graph. The LP relaxation from before has an integer optimal solution (which also solves the integer program).

Moral: Sometimes solving an integer program is no harder than solving a linear program.

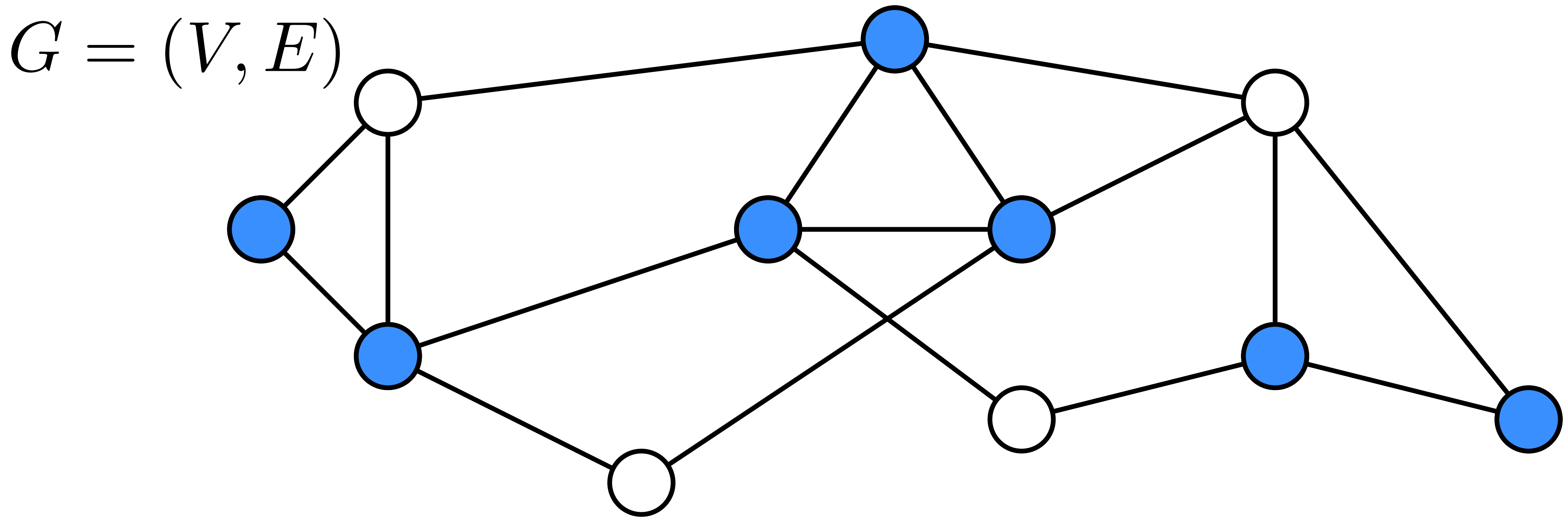
Minimum vertex cover

$G = (V, E)$



Recall: a minimum vertex cover, is a smallest possible subset $V' \subseteq V$ such that for every edge $uv \in E$, it holds that $u \in V'$ or $v \in V'$.

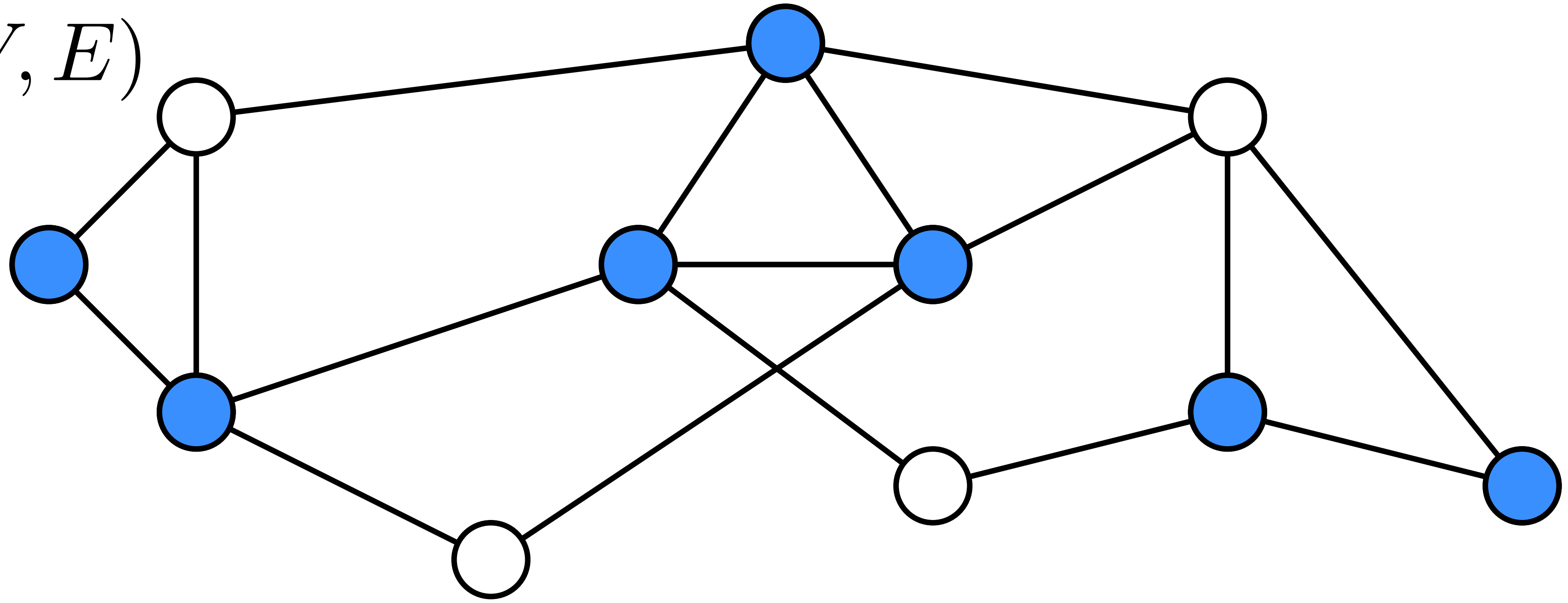
Minimum vertex cover



Recall: a minimum vertex cover, is a smallest possible subset $V' \subseteq V$ such that for every edge $uv \in E$, it holds that $u \in V'$ or $v \in V'$.

Minimum vertex cover

$G = (V, E)$

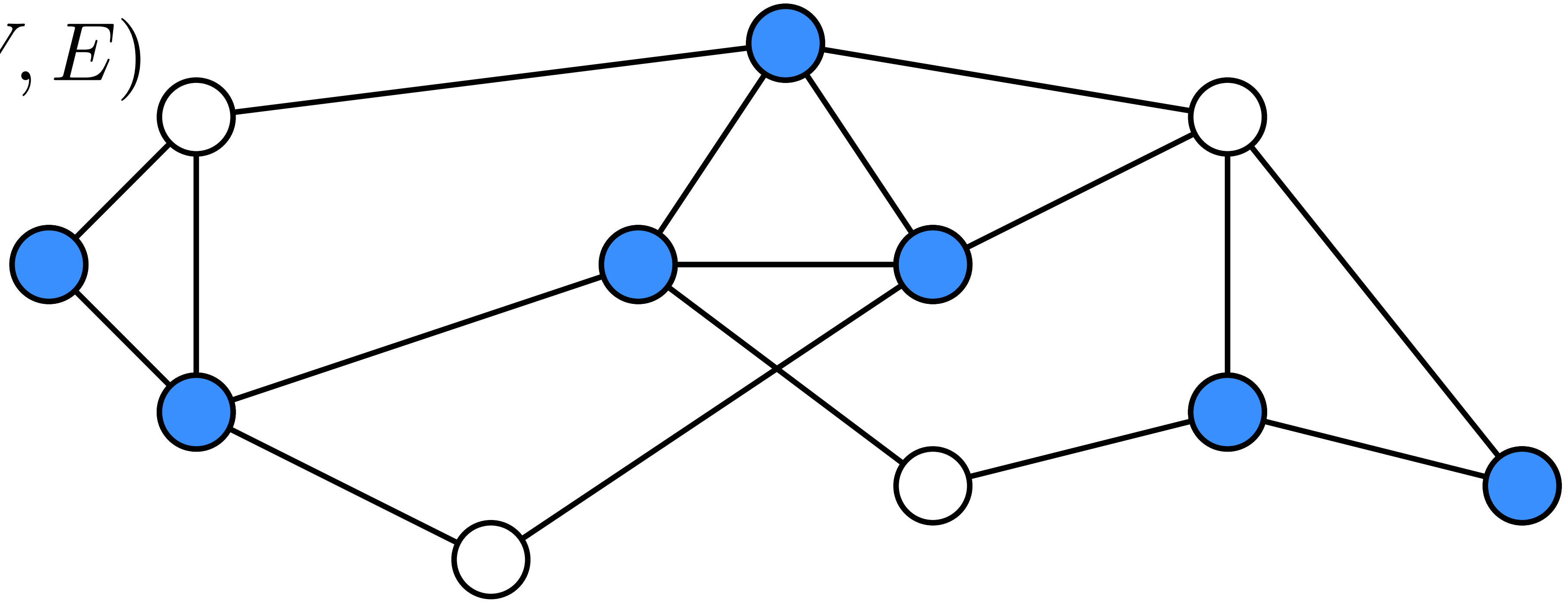


Recall: a minimum vertex cover, is a smallest possible subset $V' \subseteq V$ such that for every edge $uv \in E$, it holds that $u \in V'$ or $v \in V'$.

How can we formulate this as an ILP?

Minimum vertex cover

$G = (V, E)$



minimize $\sum_{v \in V} x_v$

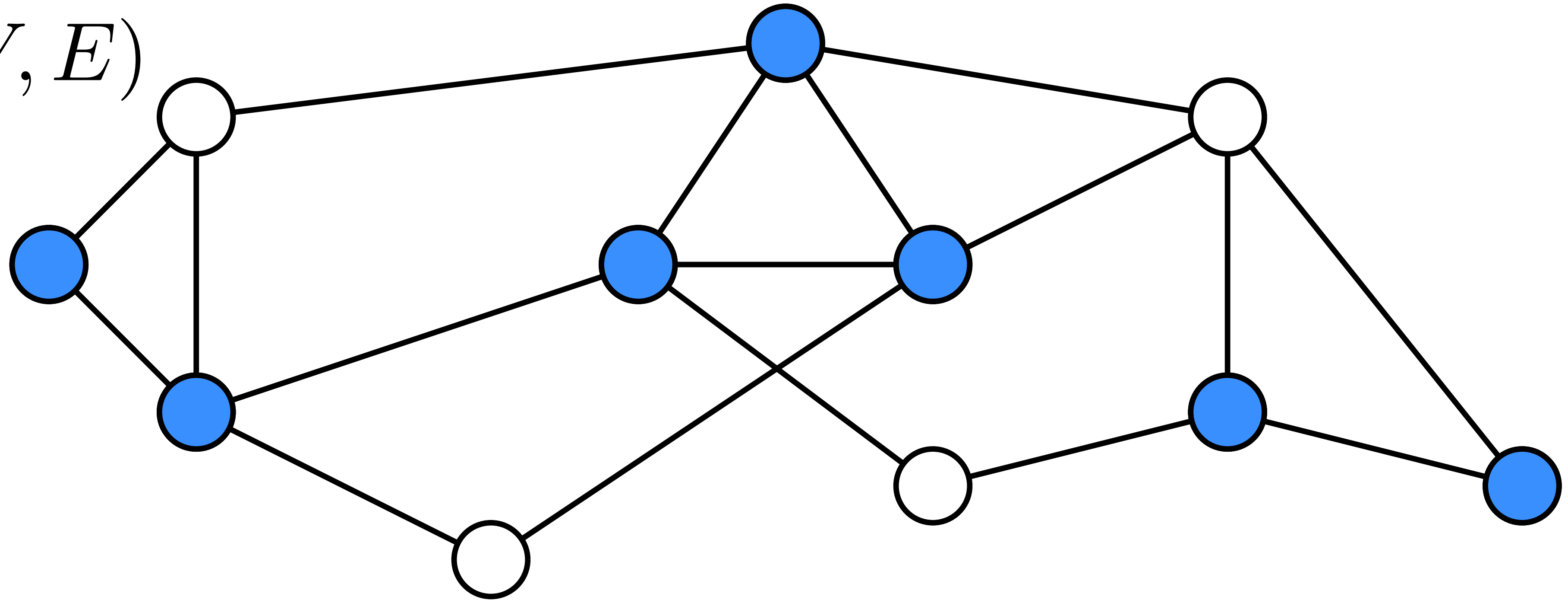
subject to $x_u + x_v \geq 1$ for every edge $\{u, v\} \in E$

$x_v \in \{0, 1\}$ for all $v \in V$.

variable x_v encodes whether vertex v is contained in the cover

Minimum vertex cover

$G = (V, E)$



minimize $\sum_{v \in V} x_v$

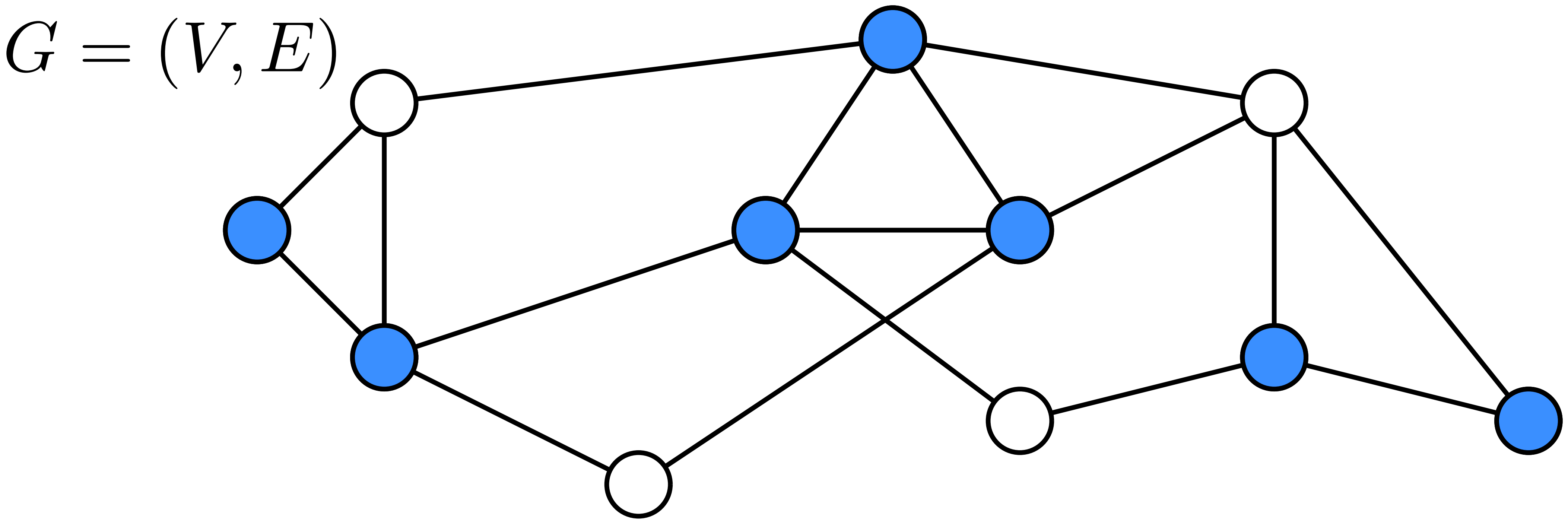
subject to $x_u + x_v \geq 1$ for every edge $\{u, v\} \in E$

$x_v \in \{0, 1\}$ for all $v \in V$.

variable x_v encodes whether vertex v is contained in the cover

LP relaxation: $0 \leq x_v \leq 1$

Minimum vertex cover



minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for every edge $\{u, v\} \in E$

$x_v \in \{0, 1\}$ for all $v \in V$.

variable x_v encodes whether vertex v is contained in the cover

LP relaxation: $0 \leq x_v \leq 1$ How does this help?

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

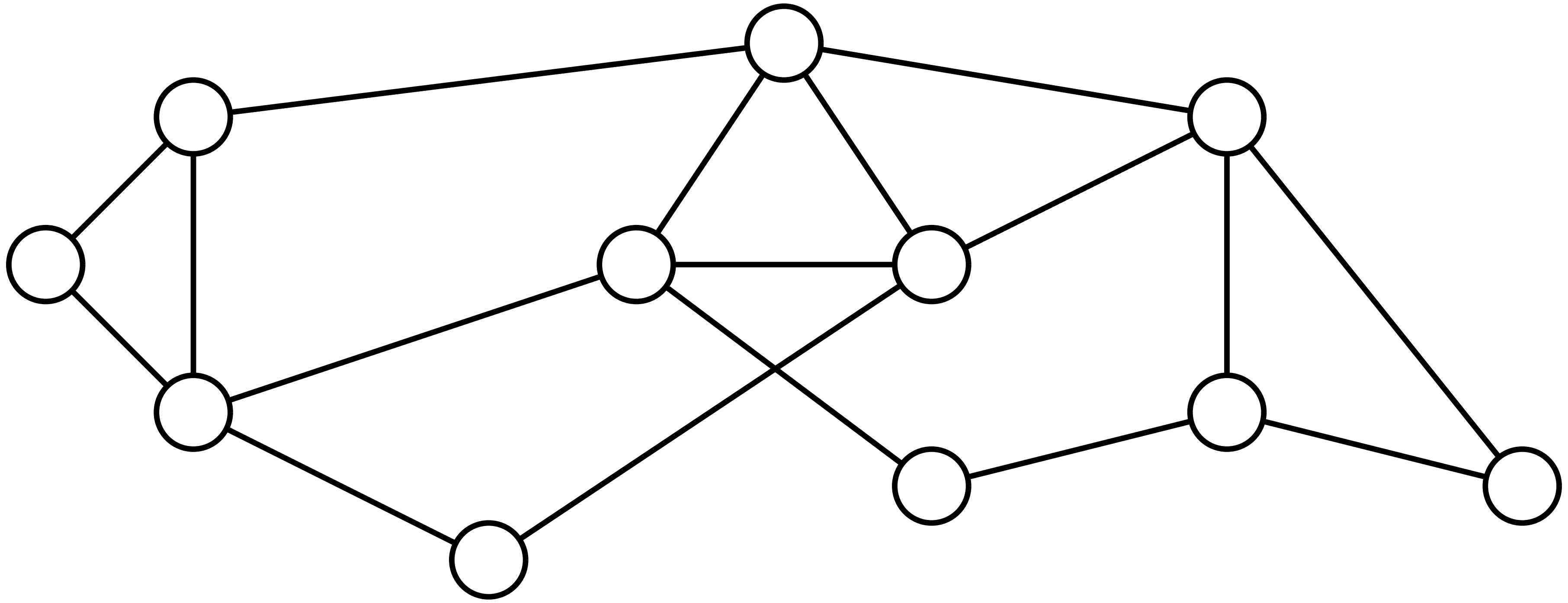
Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

What solution do we get here?

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.



Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

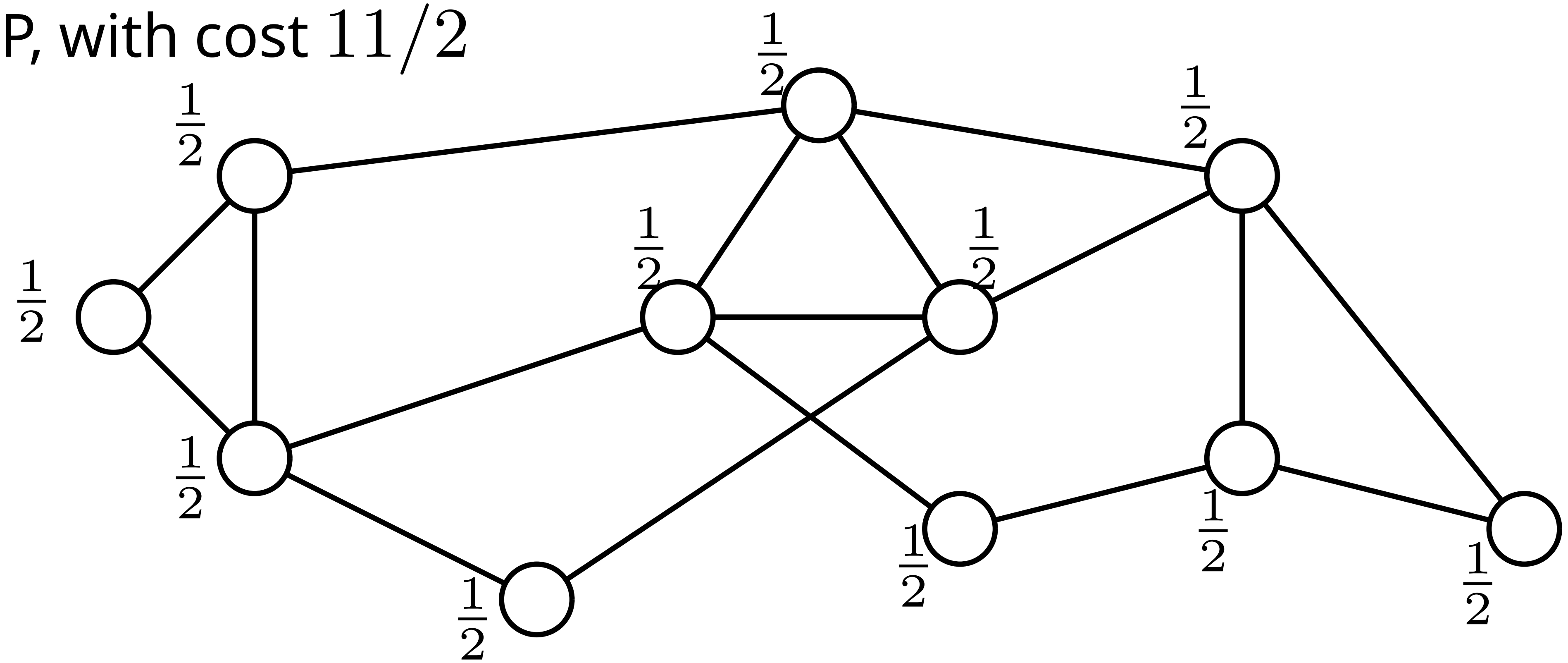
What solution do we get here?

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

solution to LP, with cost $11/2$



Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

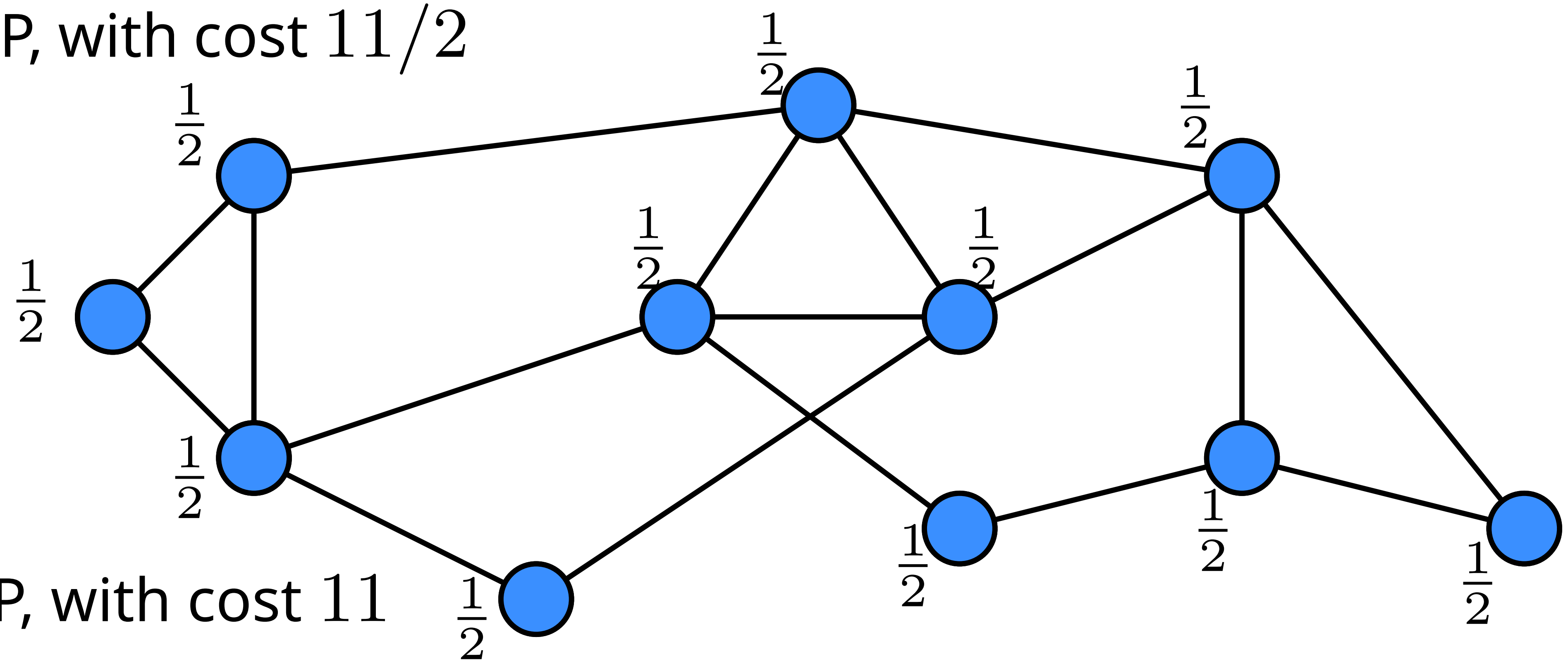
$$\text{minimize} \sum_{v \in V} x_v$$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

What solution do we get here?

solution to LP, with cost $11/2$



solution to IP, with cost 11

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

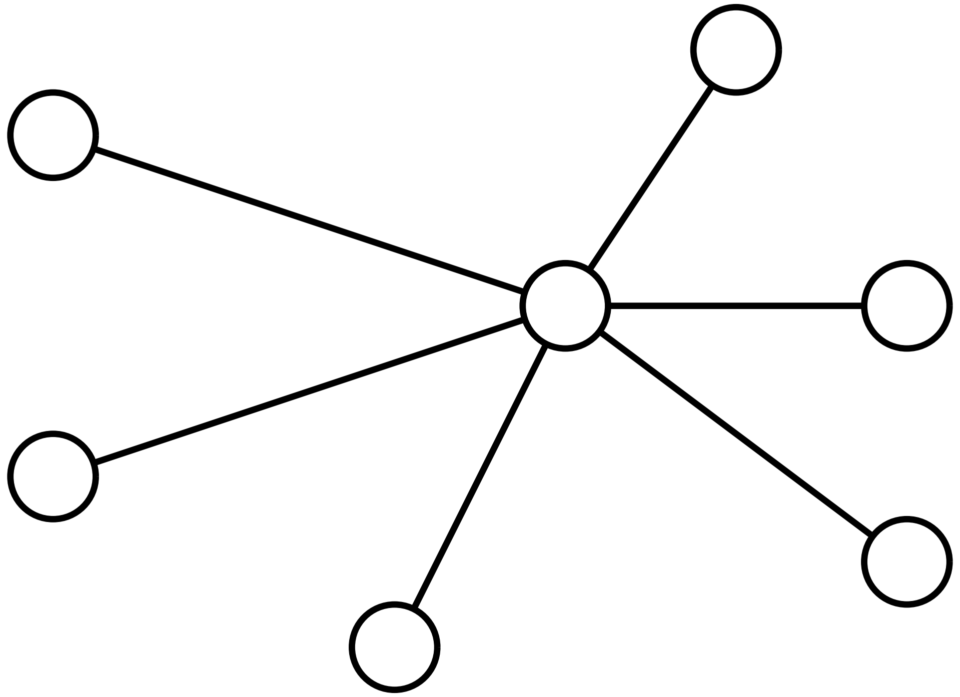
Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP, obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

What solution do we get here?

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.



Minimum vertex cover

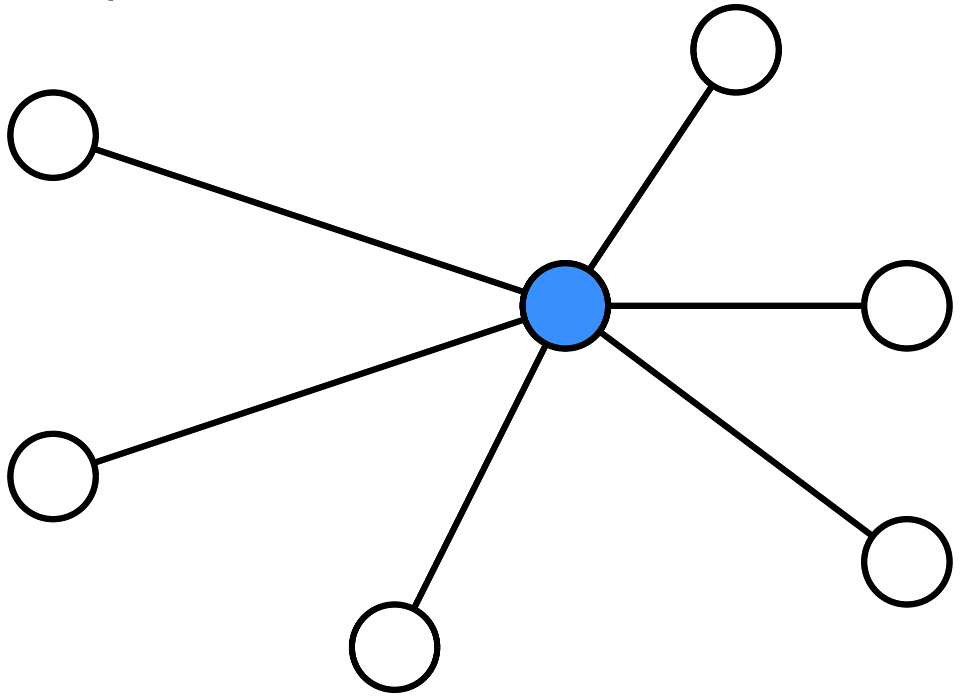
Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

What solution do we get here?

solution to LP also to IP, with cost 1



minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.

Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

Fact: $|S_{\text{IP}}| \leq |S_{\text{LP}}| \leq 2|S_{\text{IP}}|$

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.
Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

$$\text{minimize } \sum_{v \in V} x_v$$

$$\text{subject to } x_u + x_v \geq 1 \text{ for all } \{u, v\} \in E$$

$$x_v \in [0, 1] \text{ for all } v \in V.$$

Fact: $|S_{\text{IP}}| \leq |S_{\text{LP}}| \leq 2|S_{\text{IP}}|$

x_v^*: optimal LP solution,
$\tilde{x}_v$: optimal ILP solution

Proof of second inequality

$$|S_{\text{LP}}| \leq 2\sum_{v \in V} x_v^* \leq 2\sum_{v \in V} \tilde{x}_v = 2 \cdot |S_{\text{IP}}|$$

by definition of S_{LP}

since any solution to IP is also a feasible solution for the LP relaxation.

Minimum vertex cover

Rounding:

Let $S_{\text{IP}} \subseteq V$ be a vertex cover solving IP.
Let $S_{\text{LP}} \subseteq V$ be a vertex cover solving LP,
obtained by $S_{\text{LP}} = \{v \in V \mid x_v \geq \frac{1}{2}\}$.

minimize $\sum_{v \in V} x_v$

subject to $x_u + x_v \geq 1$ for all $\{u, v\} \in E$

$x_v \in [0, 1]$ for all $v \in V$.

Fact: $|S_{\text{IP}}| \leq |S_{\text{LP}}| \leq 2|S_{\text{IP}}|$

Proof of second inequality

$$|S_{\text{LP}}| \leq 2 \sum_{v \in V} x_v^* \leq 2 \sum_{v \in V} \tilde{x}_v = 2 \cdot |S_{\text{IP}}|$$

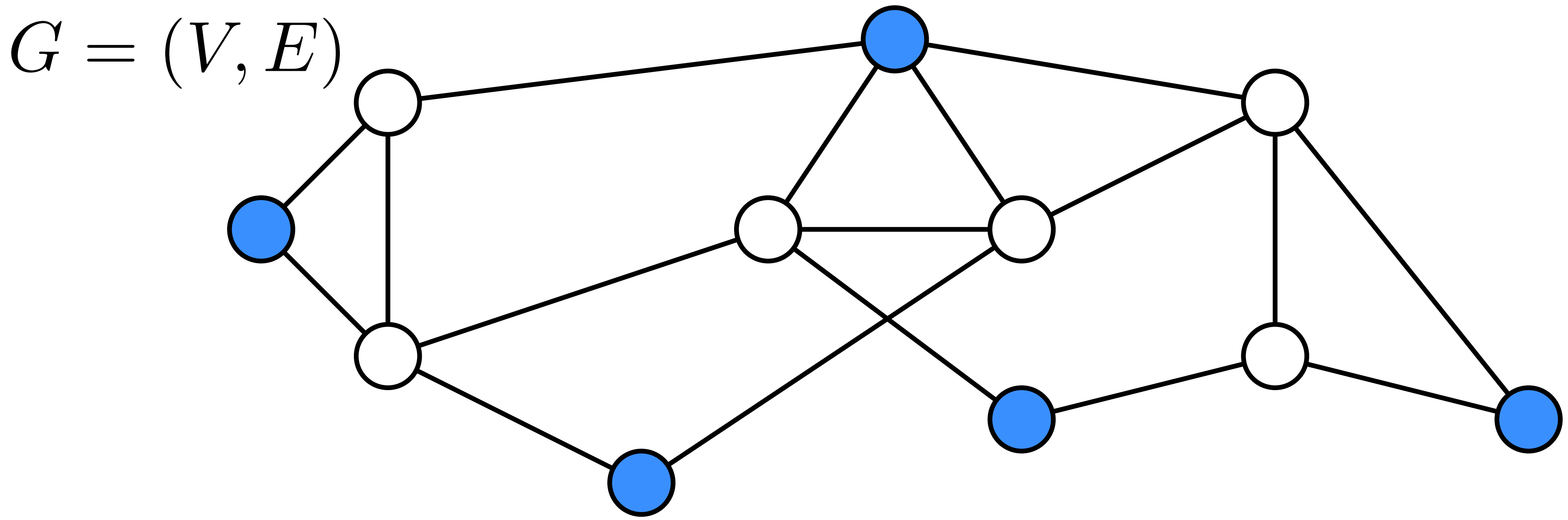
x_v^*: optimal LP solution,
$\tilde{x}_v$: optimal ILP solution

by definition of S_{LP}

since any solution to IP is also a feasible solution for the LP relaxation.

Moral: Sometimes solving an LP relaxation gives an approximate solution to an **NP**-hard integer program.

Maximum independent set

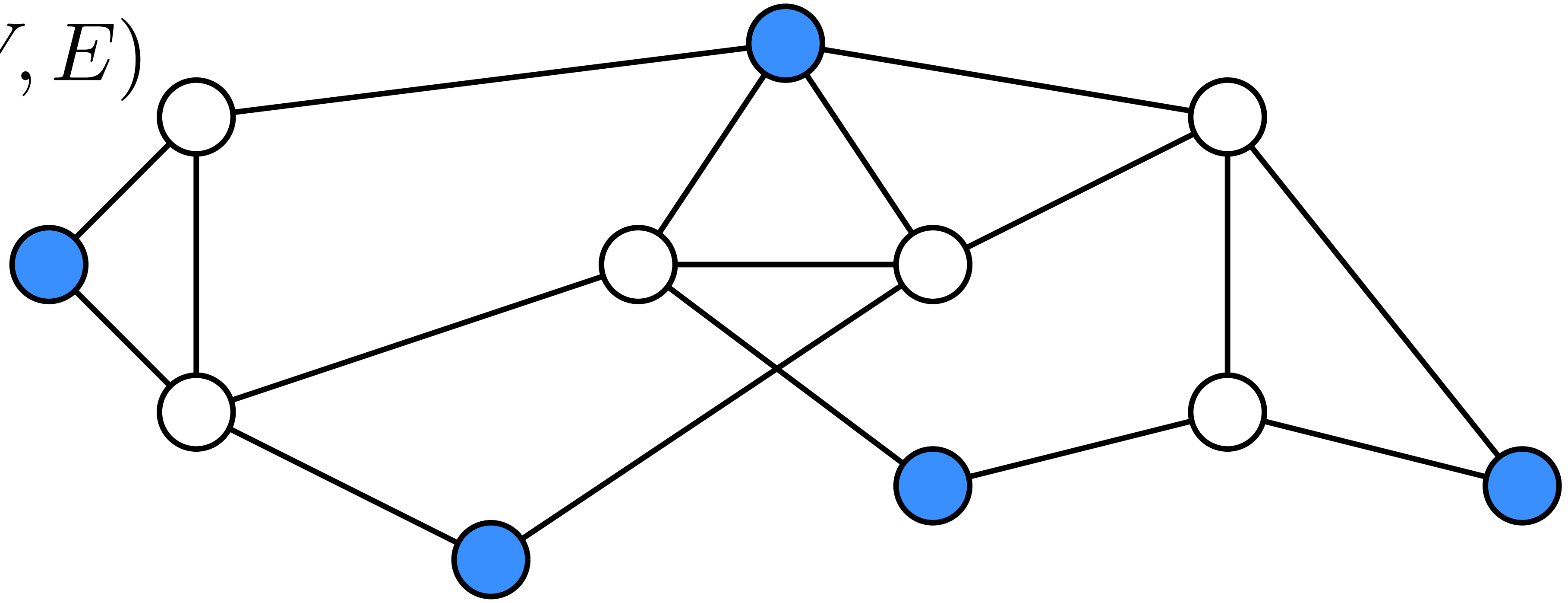


Recall: a maximum independent, is a largest possible subset $V' \subseteq V$ such that for any $u, v \in V'$, it holds that $(u, v) \notin E$.

How can we formulate this as an ILP?

Maximum independent set

$G = (V, E)$



maximize $\sum_{v \in V} x_v$

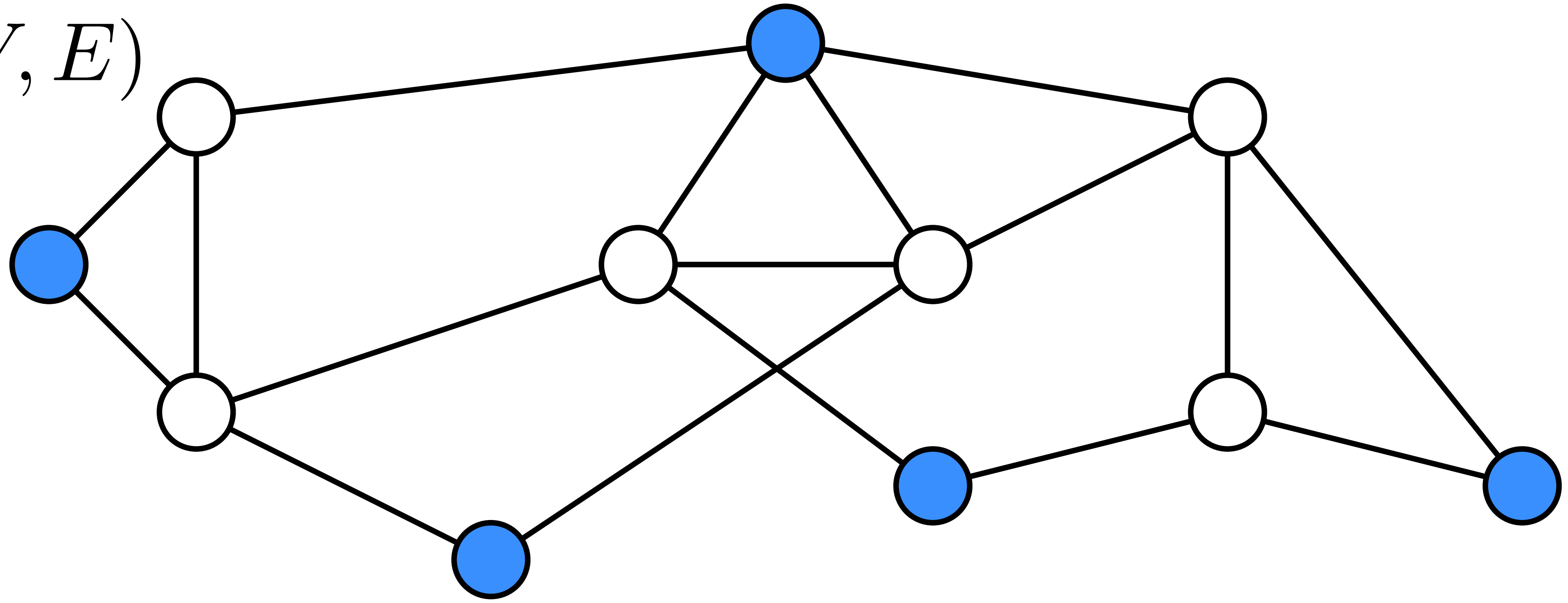
subject to $x_u + x_v \leq 1$ for each edge $\{u, v\} \in E$

$x_v \in \{0, 1\}$ for all $v \in V$.

variable x_v encodes whether vertex v is contained in the set

Maximum independent set

$G = (V, E)$



maximize $\sum_{v \in V} x_v$

subject to $x_u + x_v \leq 1$ for each edge $\{u, v\} \in E$

$x_v \in \{0, 1\}$ for all $v \in V$.

variable x_v encodes whether vertex v is contained in the set

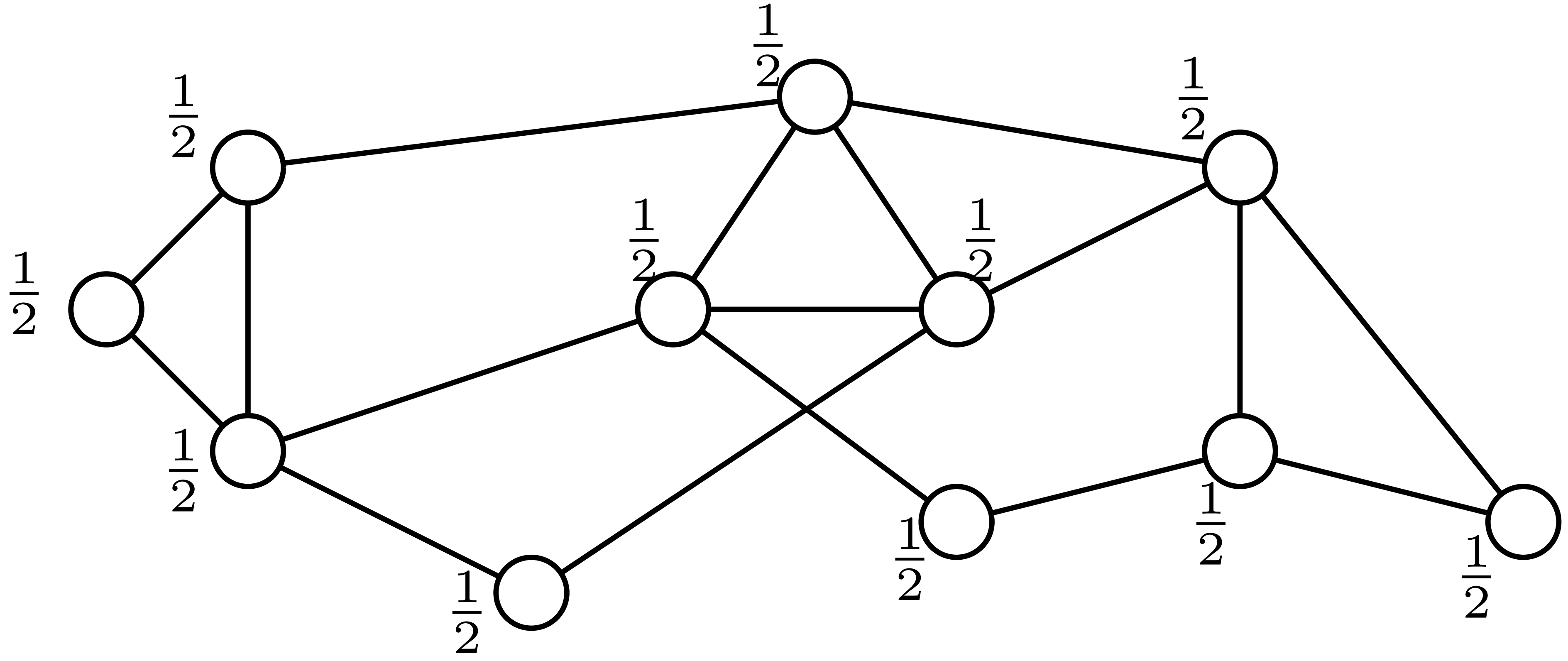
LP relaxation: $0 \leq x_v \leq 1$

Maximum independent set

In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

Maximum independent set

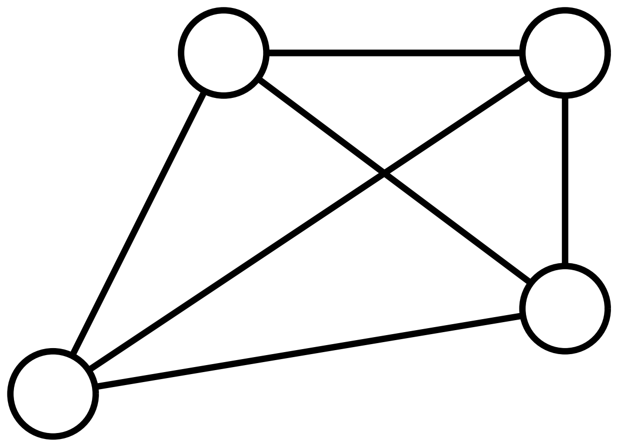
In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.



Maximum independent set

In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

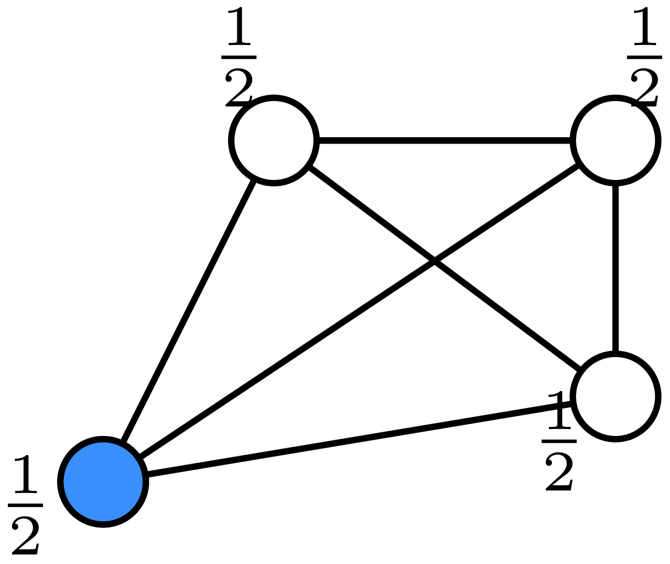
In a complete graph, what are the optimum LP and IP solution?



Maximum independent set

In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

In a complete graph, the LP has optimum $n/2$, and the IP optimum 1.



Maximum independent set

In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

In a complete graph, the LP has optimum $n/2$, and the IP optimum 1.

Moral: Sometimes an LP relaxation tells us next to nothing about the integer program.

Maximum independent set

In the LP relaxation ($0 \leq x_v \leq 1$) we always have the feasible solution $x_v = \frac{1}{2}$ for all v, meaning the optimum is at least $\frac{1}{2}|V|$.

In a complete graph, the LP has optimum $n/2$, and the IP optimum 1.

Moral: Sometimes an LP relaxation tells us next to nothing about the integer program.

Approximation for this problem is known to be hard:

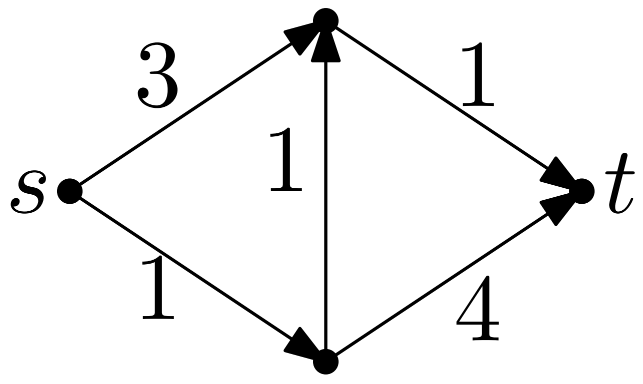
J. Håstad: Clique is hard to approximate within $n^{1-\epsilon}$,
Acta Math. 182(1999): 105-142

Bonus Examples

Shortest Path and TSP

Shortest Path Problem

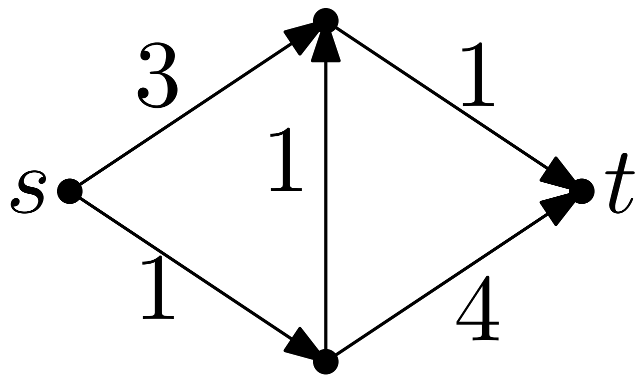
Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP?

Shortest Path Problem

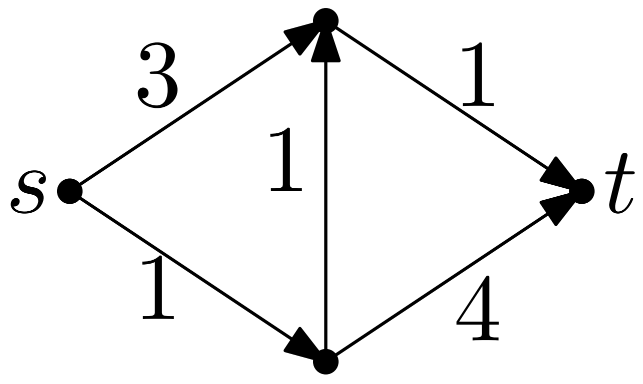
Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.

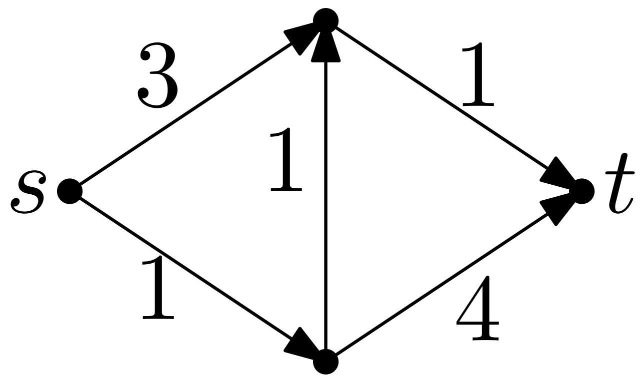


Can we also model this as ILP? Yes! How?

Idea 1: use variable x_{uv} for whether edge (u, v) is used.

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 1: use variable x_{uv} for whether edge (u, v) is used.

minimize $\displaystyle\sum_{(u,v)\in E} w(u,v)x_{uv}$

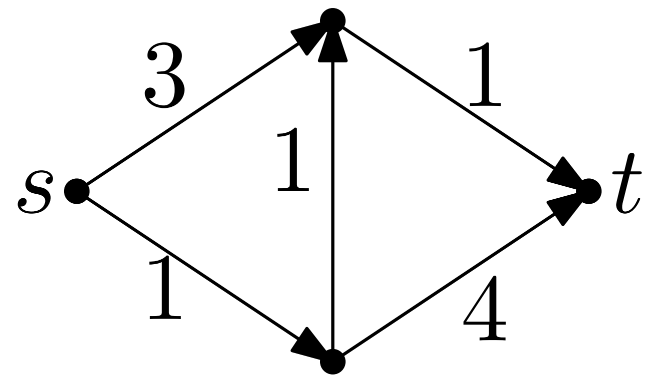
subject to $\displaystyle\sum_{(u,v)\in E} x_{uv} = \sum_{(v,w)\in E} x_{vw}$ for each vertex $v \in V \setminus \{s,t\}$, and

$$\sum_{(u,t)\in E} x_{ut} = 1.$$

$x_{uv} \in \{0, 1\}$ for each edge $(u, v) \in E$.

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 1: use variable x_{uv} for whether edge (u, v) is used.

minimize $\displaystyle\sum_{(u,v)\in E} w(u,v)x_{uv}$

subject to $\displaystyle\sum_{(u,v)\in E} x_{uv} = \sum_{(v,w)\in E} x_{vw}$ for each vertex $v \in V \setminus \{s,t\}$, and

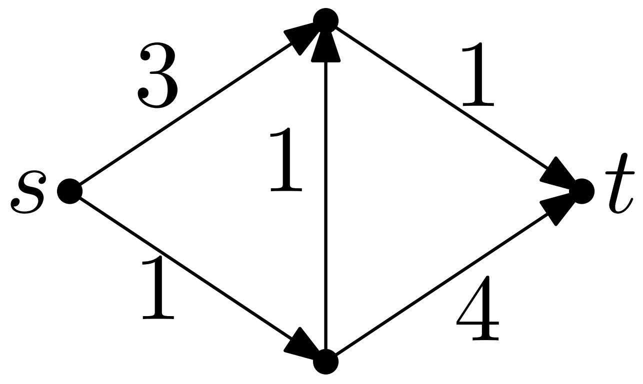
$\displaystyle\sum_{(u,t)\in E} x_{ut} = 1.$

$x_{uv} \in \{0, 1\}$ for each edge $(u, v) \in E$.

LP-relaxation has $\{0, 1\}$-solution, like MinWeight Perfect Matching

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.

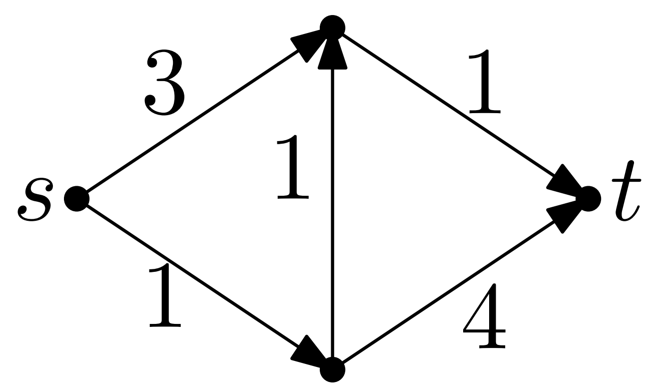


Can we also model this as ILP? Yes! How?

Idea 2: use variable d_v for distance from s to v.
Then we want that $d_v \leq d_u + w(u, v)$ for all edges (u, v).

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w,
we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 2: use variable d_v for distance from s to v.
Then we want that $d_v \leq d_u + w(u, v)$ for all edges (u, v).

maximize d_t

subject to

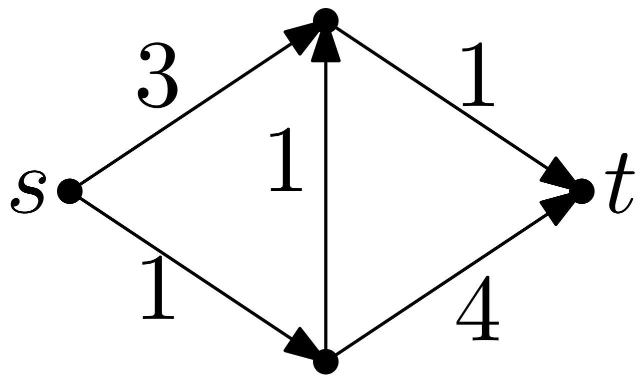
$$d_v - d_u \leq w(u, v) \;\; \forall (u, v) \in E$$

$$d_s = 0$$

Actually an LP!

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 2: use variable d_v for distance from s to v.
Then we want that $d_v \leq d_u + w(u, v)$ for all edges (u, v).

maximize d_t

subject to

$$d_v - d_u \leq w(u, v) \;\; \forall (u, v) \in E$$

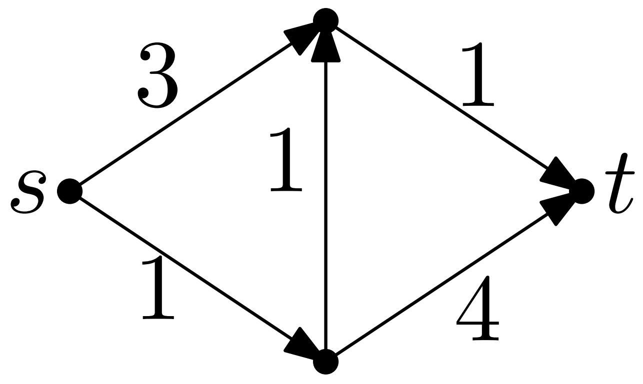
$$d_s = 0$$

Why do we maximize?

Actually an LP!

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w, we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 2: use variable d_v for distance from s to v.
Then we want that $d_v \leq d_u + w(u, v)$ for all edges (u, v).

maximize d_t

subject to

$d_v - d_u \leq w(u, v) \ \forall (u, v) \in E$

$d_s = 0$

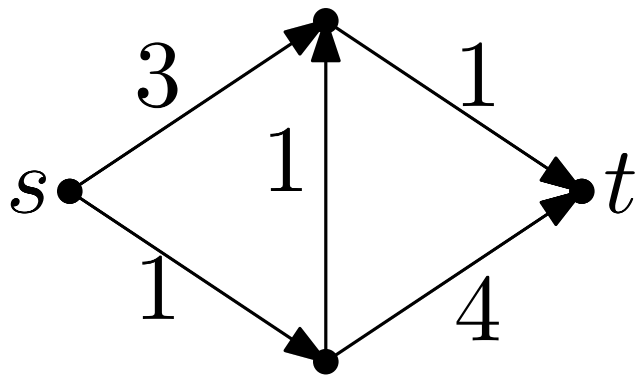
Why do we maximize?

otherwise we could set all d_v to zero

Actually an LP!

Shortest Path Problem

Given a directed graph $G = (V, E)$ with edge weights w,
we are looking for a shortest path from s to t.



Can we also model this as ILP? Yes! How?

Idea 2: use variable d_v for distance from s to v.
Then we want that $d_v \leq d_u + w(u, v)$ for all edges (u, v).

maximize d_t

subject to

$d_v - d_u \leq w(u, v) \;\; \forall (u, v) \in E$

$d_s = 0$

Why do we maximize?

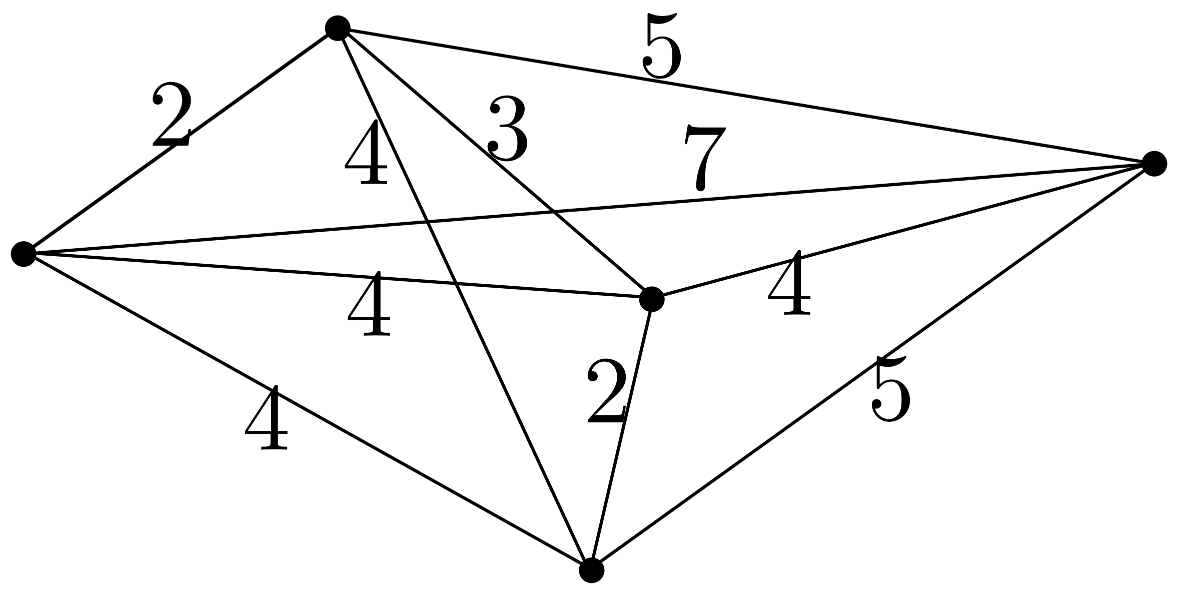
otherwise we could set all d_v to zero

simulating a $\min(\cdot)$ by $\leq$ and maximize

Actually an LP!

Traveling Salesperson Tour

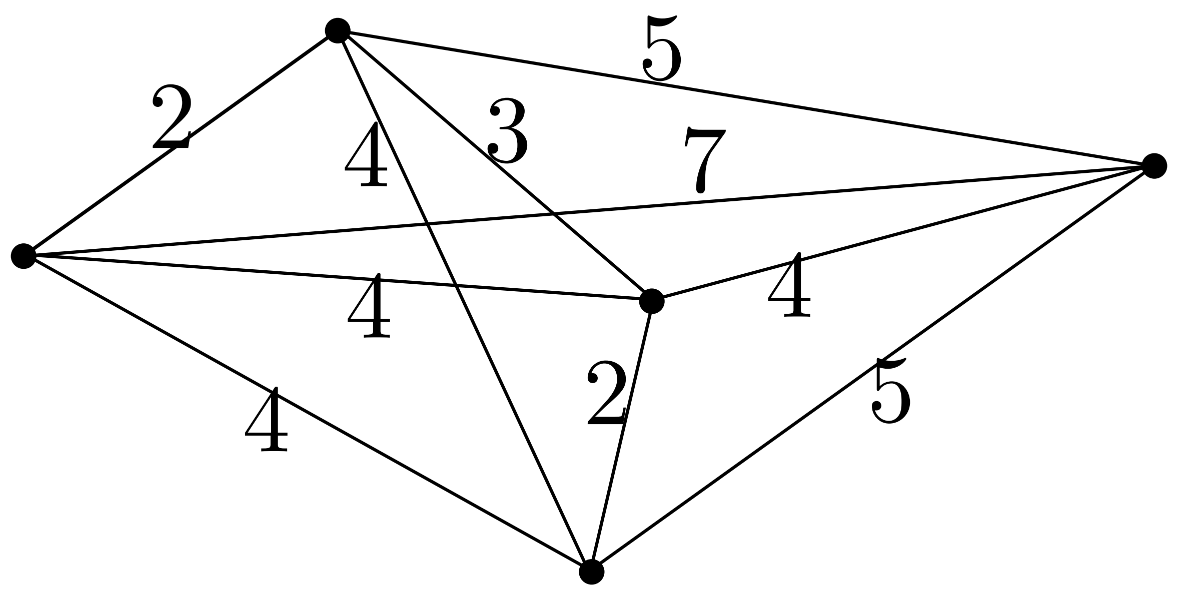
Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.

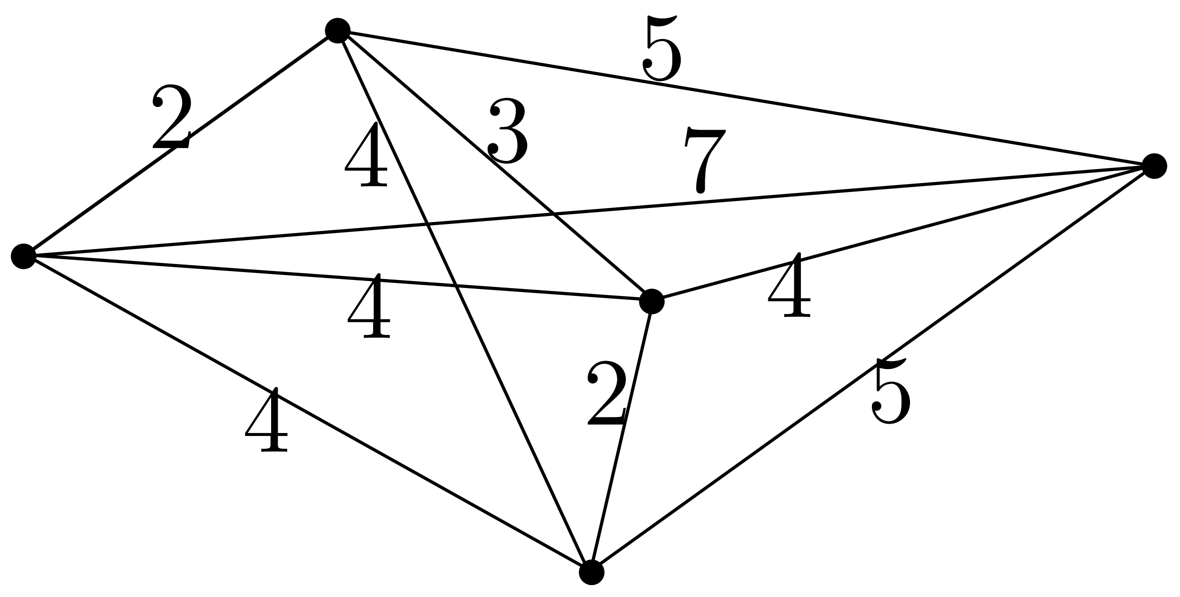
Can we also model this as ILP?



Traveling Salesperson Tour

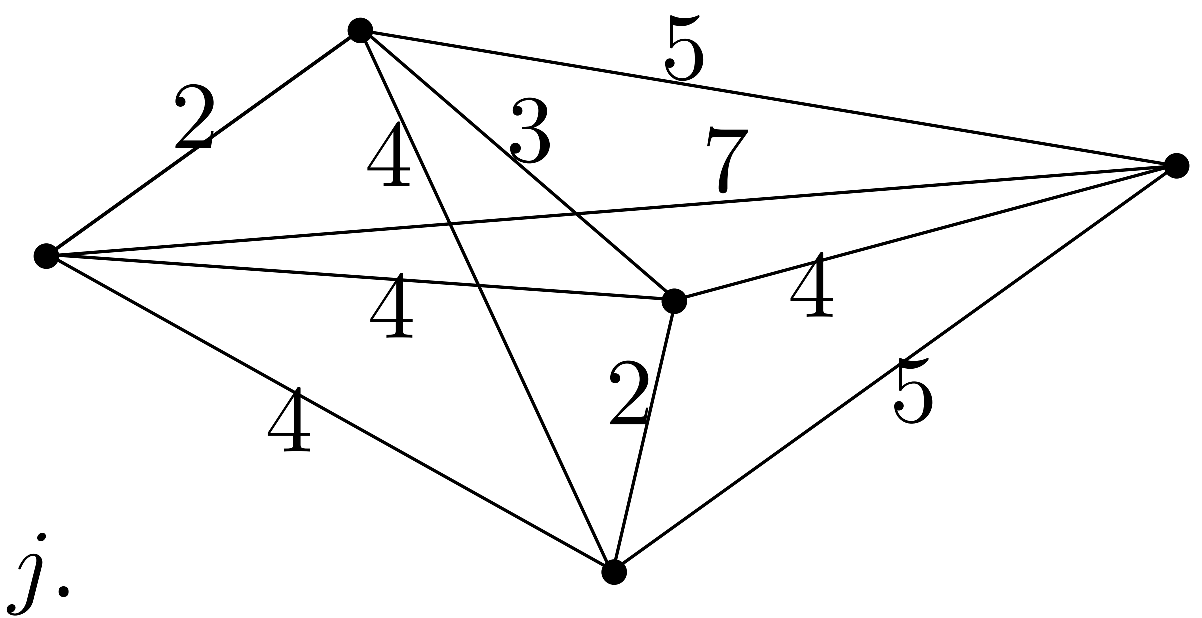
Given an undirected complete graph $G = (V, E)$ with edge weights $c \colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.

Can we also model this as ILP? Yes! How?



Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.

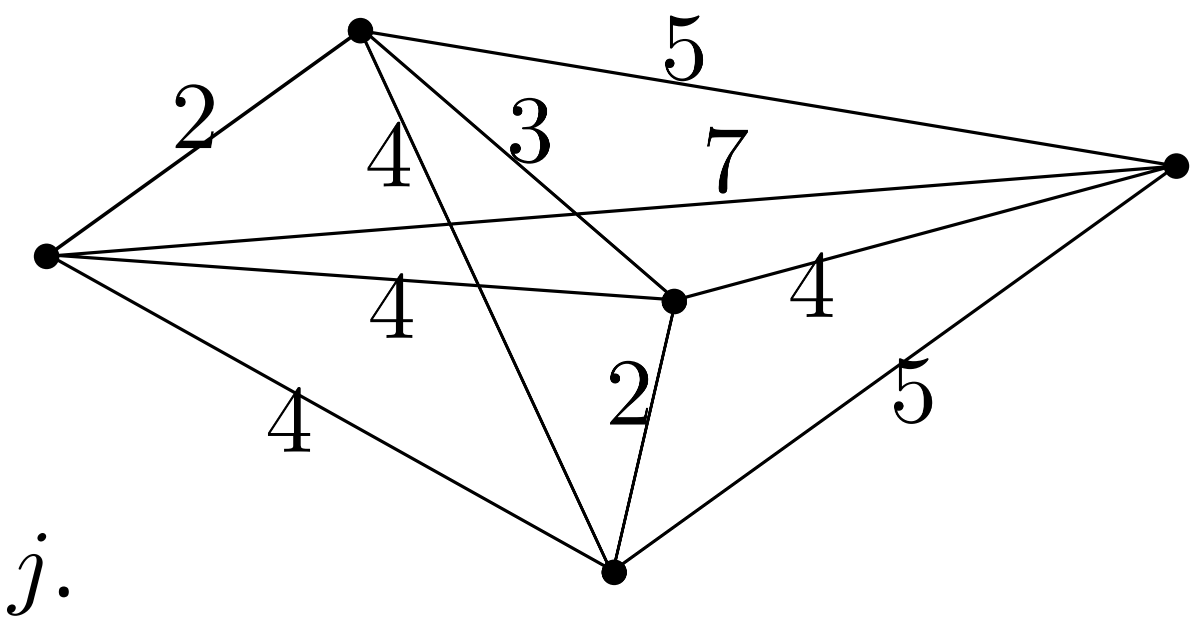


Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1\leq i\leq n} \sum_{1\leq j\leq n} x_{ij} c(i,j)$

subject to

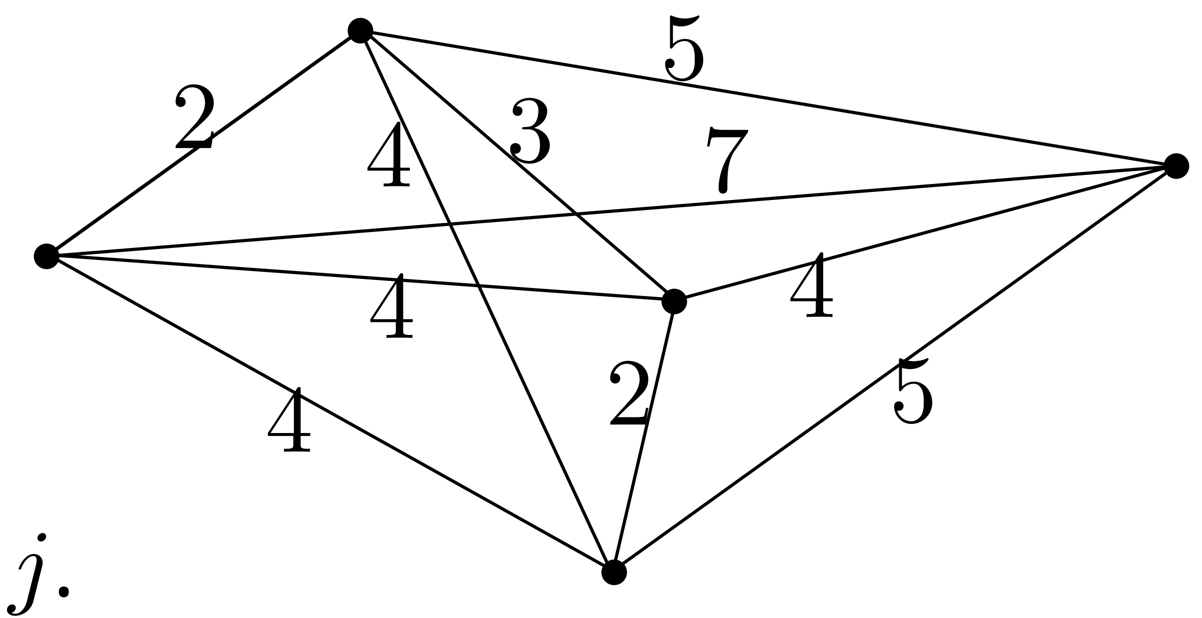
$\sum_{1\leq j\leq n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1\leq i\leq n} x_{ij} = 1$ for $j = 1, \ldots, n$

$x_{ij} \in \{0, 1\}$ for $i, j = 1, \ldots, n$

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1\le i\le n}\sum_{1\le j\le n} x_{ij}c(i,j)$

subject to

$\sum_{1\le j\le n} x_{ij} = 1$ for $i = 1,\dots,n$

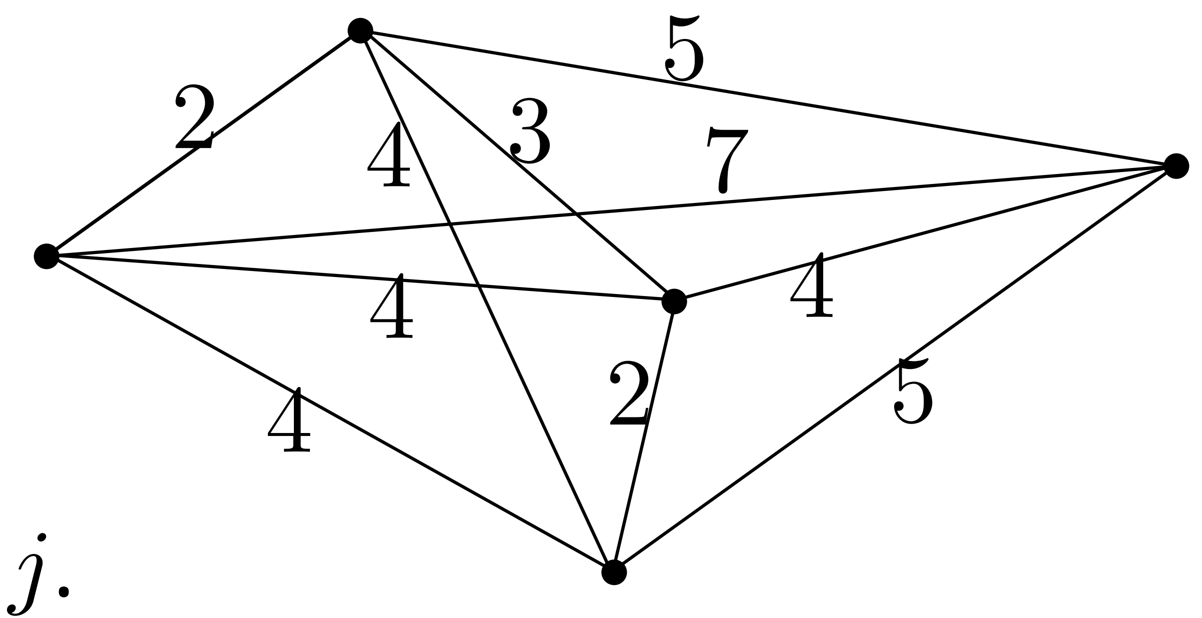
$\sum_{1\le i\le n} x_{ij} = 1$ for $j = 1,\dots,n$

$x_{ij} \in \{0,1\}$ for $i,j = 1,\dots,n$

Does this correctly solve TSP?

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1 \le i \le n} \sum_{1 \le j \le n} x_{ij} c(i, j)$

subject to

$\sum_{1 \le j \le n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1 \le i \le n} x_{ij} = 1$ for $j = 1, \ldots, n$

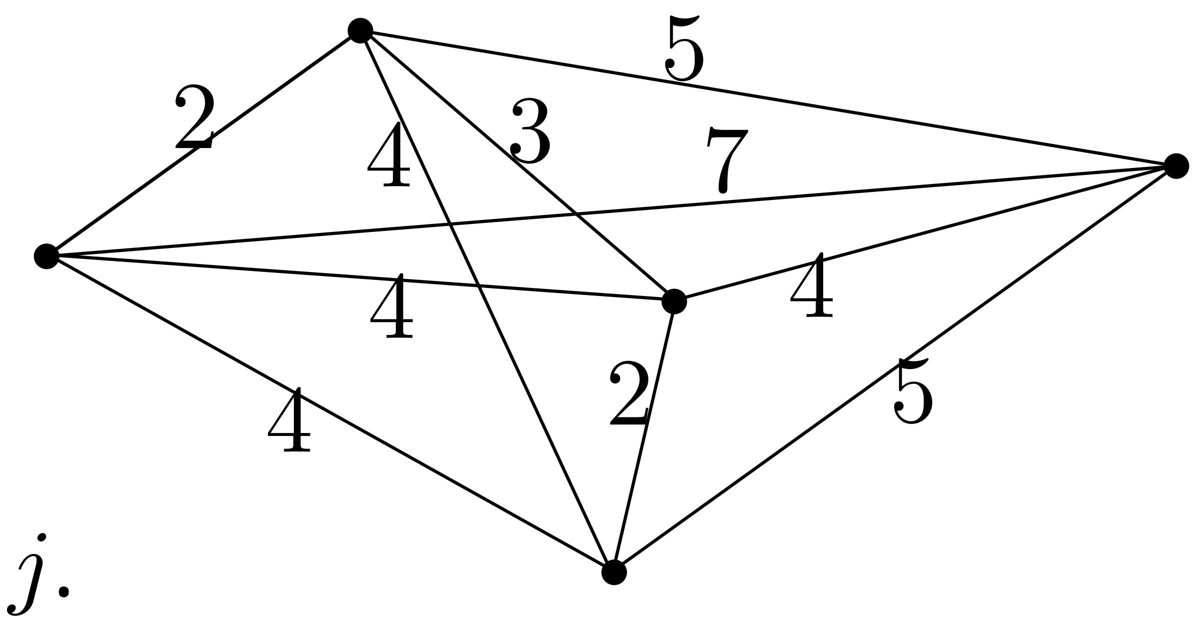
$x_{ij} \in \{0, 1\}$ for $i, j = 1, \ldots, n$

Does this correctly solve TSP? No!

A tour can “decompose” into many smaller tours.

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c \colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1 \le i \le n} \sum_{1 \le j \le n} x_{ij} c(i, j)$

subject to

$\sum_{1 \le j \le n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1 \le i \le n} x_{ij} = 1$ for $j = 1, \ldots, n$

$x_{ij} \in \{0, 1\}$ for $i, j = 1, \ldots, n$

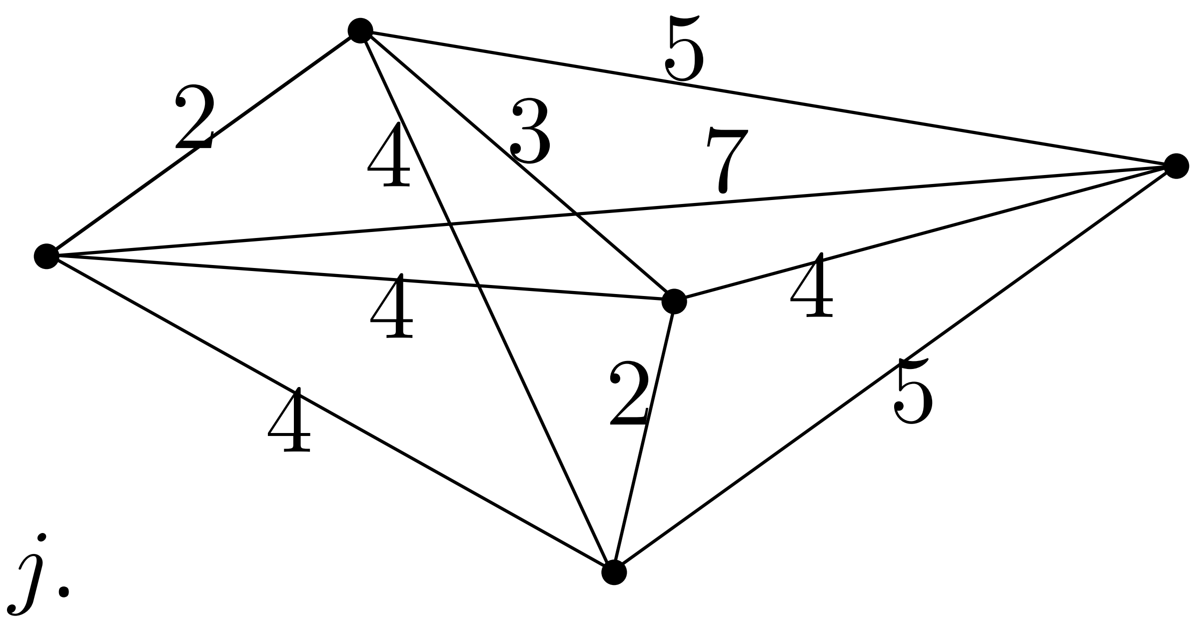
Does this correctly solve TSP? No!

A tour can "decompose" into many smaller tours.

How to avoid this?

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1 \le i \le n} \sum_{1 \le j \le n} x_{ij} c(i, j)$

subject to

$\sum_{1 \le j \le n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1 \le i \le n} x_{ij} = 1$ for $j = 1, \ldots, n$

$x_{ij} \in \{0, 1\}$ for $i, j = 1, \ldots, n$

Does this correctly solve TSP? No!

A tour can "decompose" into many smaller tours.

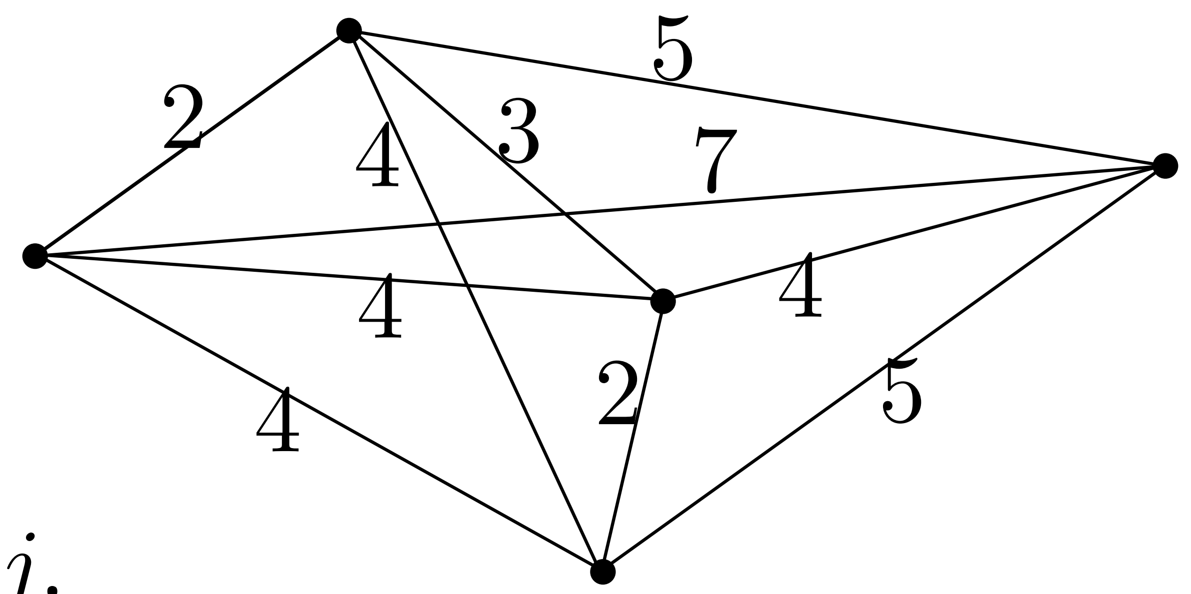
How to avoid this?

Use variables $u_j \mathrel{\widehat{=}}$ position of city j on tour

Require: $u_j \ge u_i + 1$ if $x_{ij} = 1$

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c \colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1\le i\le n} \sum_{1\le j\le n} x_{ij} c(i,j)$

subject to

$\sum_{1\le j\le n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1\le i\le n} x_{ij} = 1$ for $j = 1, \ldots, n$

$x_{ij} \in \{0,1\}$ for $i, j = 1, \ldots, n$

Does this correctly solve TSP? No!

A tour can “decompose” into many smaller tours.

How to avoid this?

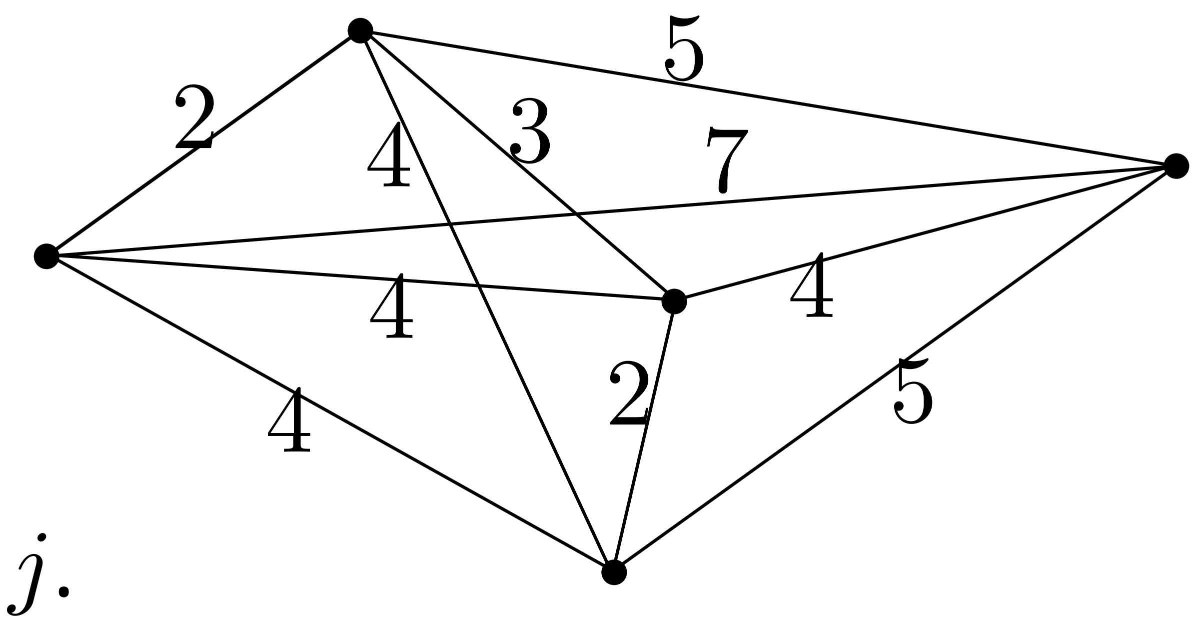
Use variables $u_j \mathrel{\widehat{=}}$ position of city j on tour

Require: $u_j \ge u_i + 1$ if $x_{ij} = 1$

by $u_j + (n-1) \ge u_i + n x_{ij}$
for all $i \ne j, j > 1$

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1 \le i \le n} \sum_{1 \le j \le n} x_{ij} c(i, j)$

subject to

$\sum_{1 \le j \le n} x_{ij} = 1$ for $i = 1, \ldots, n$

$\sum_{1 \le i \le n} x_{ij} = 1$ for $j = 1, \ldots, n$

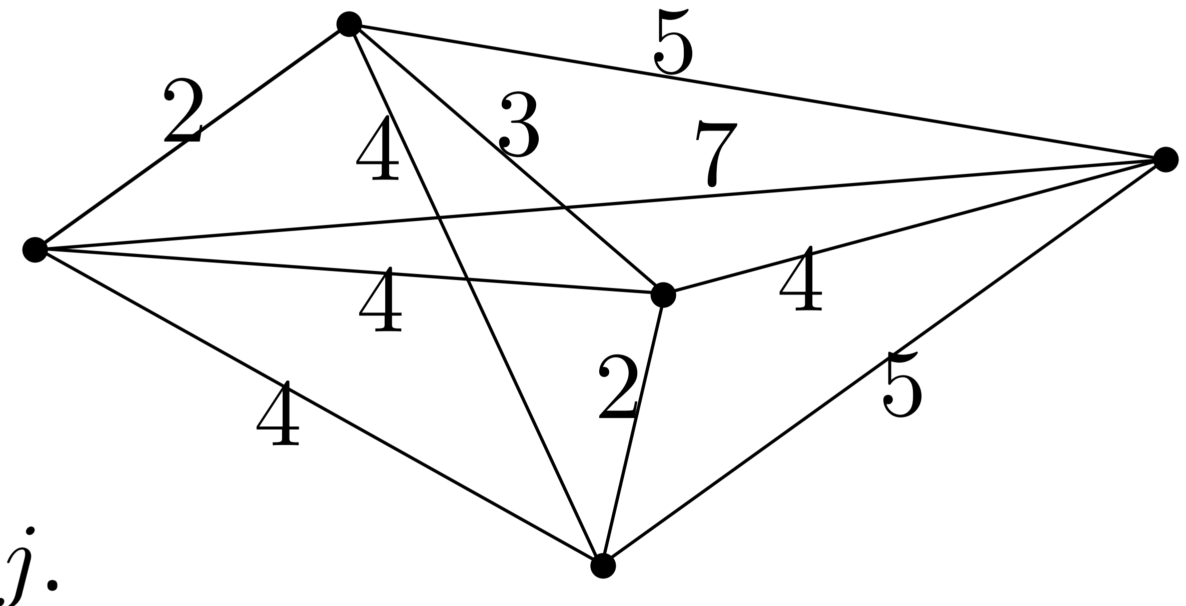
$u_i - u_j + n x_{ij} \le n - 1$ for $i \ne j, j > 1$

$x_{ij} \in \{0, 1\}$, $1 \le u_i \le n$ for $i, j = 1, .., n$

Miller-Tucker-Zemlin formulation

Traveling Salesperson Tour

Given an undirected complete graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}$, find a shortest Hamilton circuit in G.



Can we also model this as ILP? Yes! How?

Idea: use variable x_{ij} for edge from city i to city j.

minimize $\sum_{1\le i\le n} \sum_{1\le j\le n} x_{ij} c(i,j)$

subject to

$\sum_{1\le j\le n} x_{ij} = 1$ for $i = 1, \dots, n$

$\sum_{1\le i\le n} x_{ij} = 1$ for $j = 1, \dots, n$

$$\sum_{i\neq j\,:\,i,j\in Q} x_{ij} \le |Q| - 1$$

$$\forall Q \subsetneq \{1, \dots, n\}, |Q| \ge 2$$

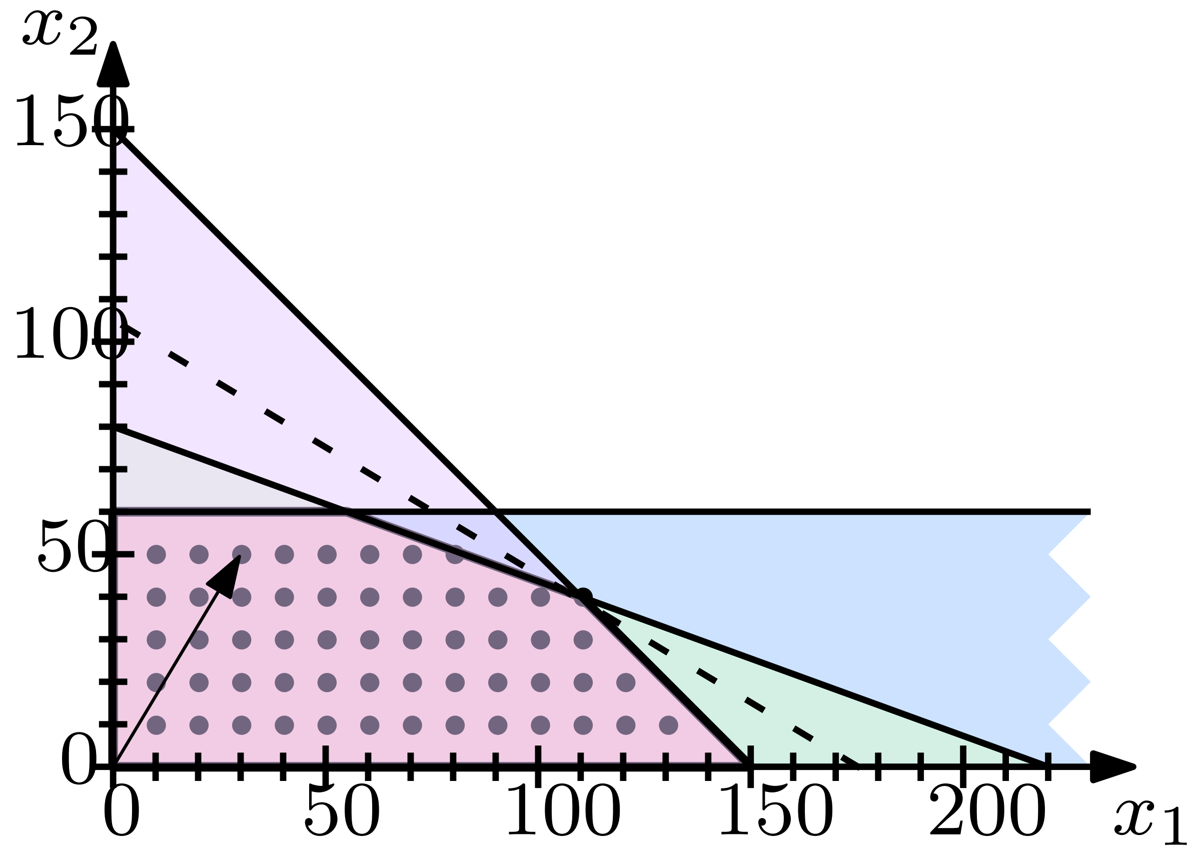
Miller-Tucker-Zemlin formulation

alternative:

Dantzig–Fulkerson–Johnson formulation with subtour-elimination constraint

Solving Integer Linear Programs

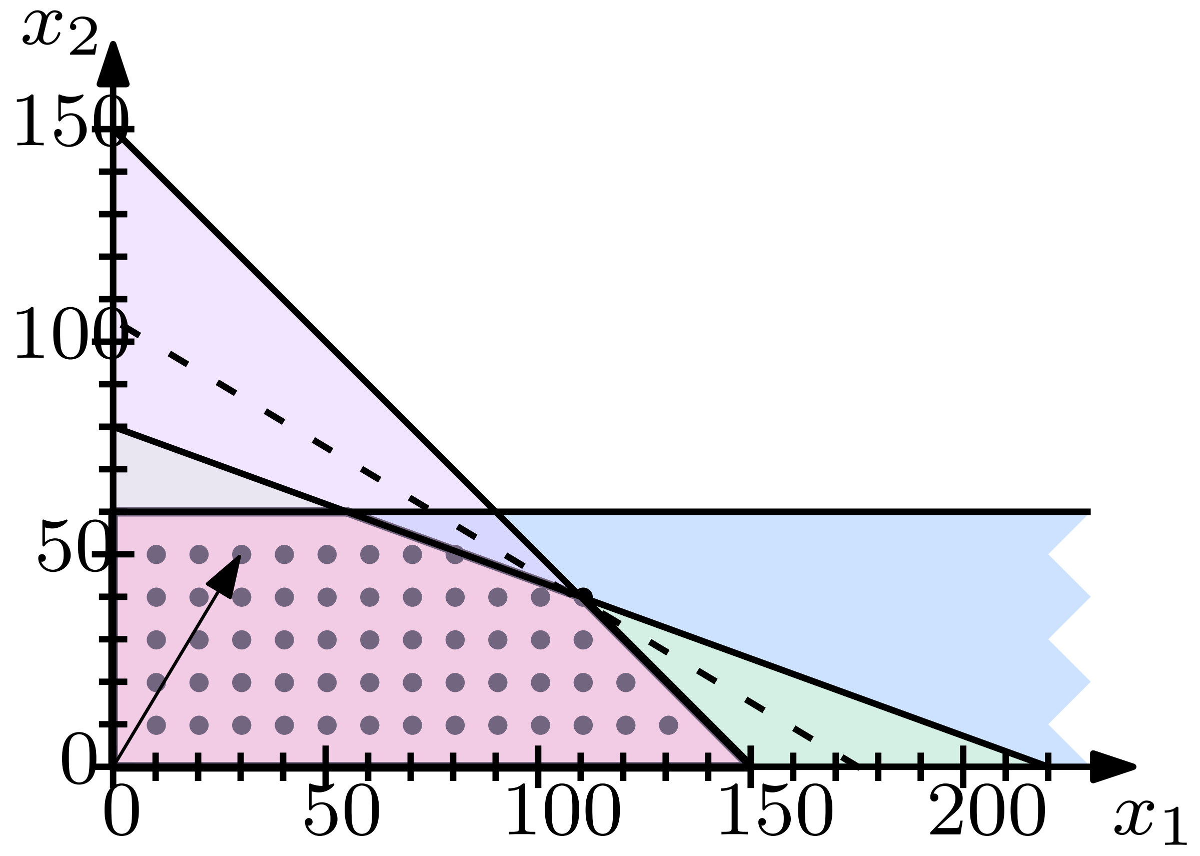
How can we solve an ILP?



Solving Integer Linear Programs

How can we solve an ILP?

Optimal integer solutions may be arbitrary far from relaxed LP solutions



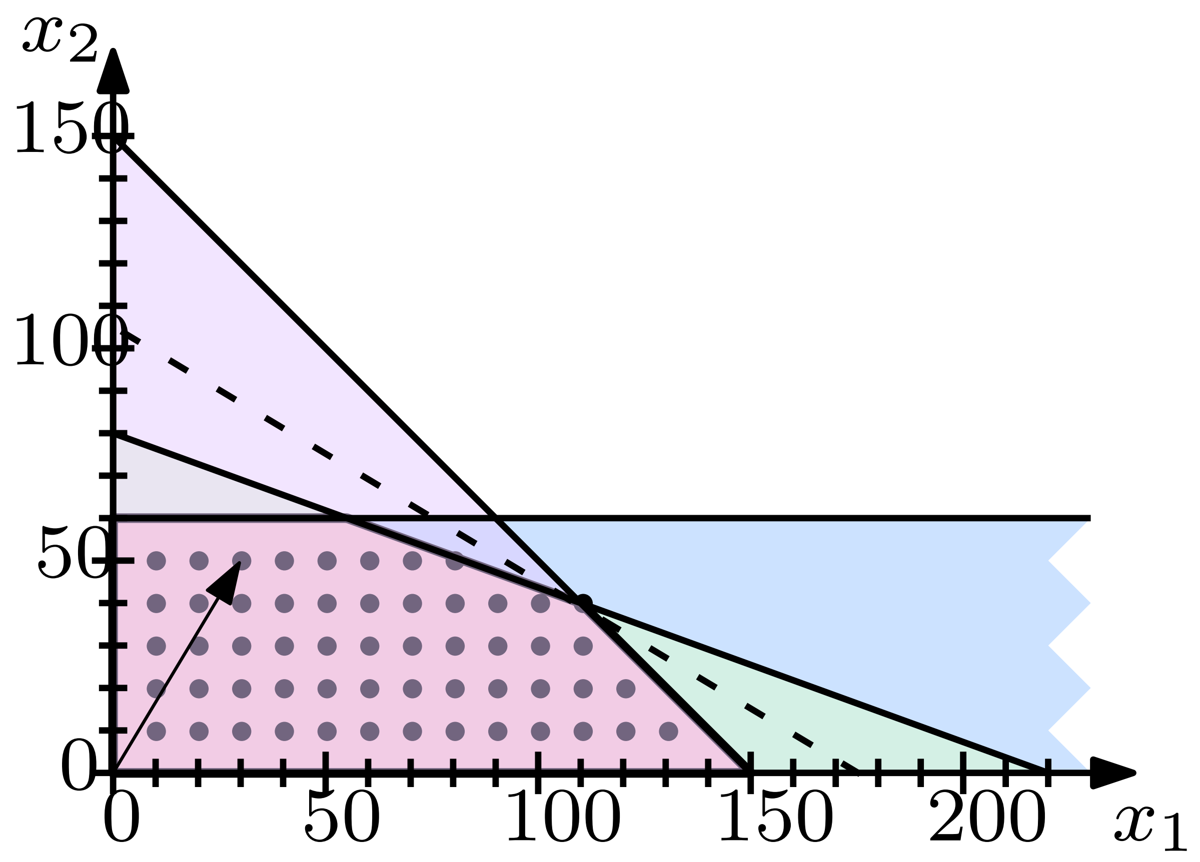
Solving Integer Linear Programs

How can we solve an ILP?

Optimal integer solutions may be arbitrary far from relaxed LP solutions

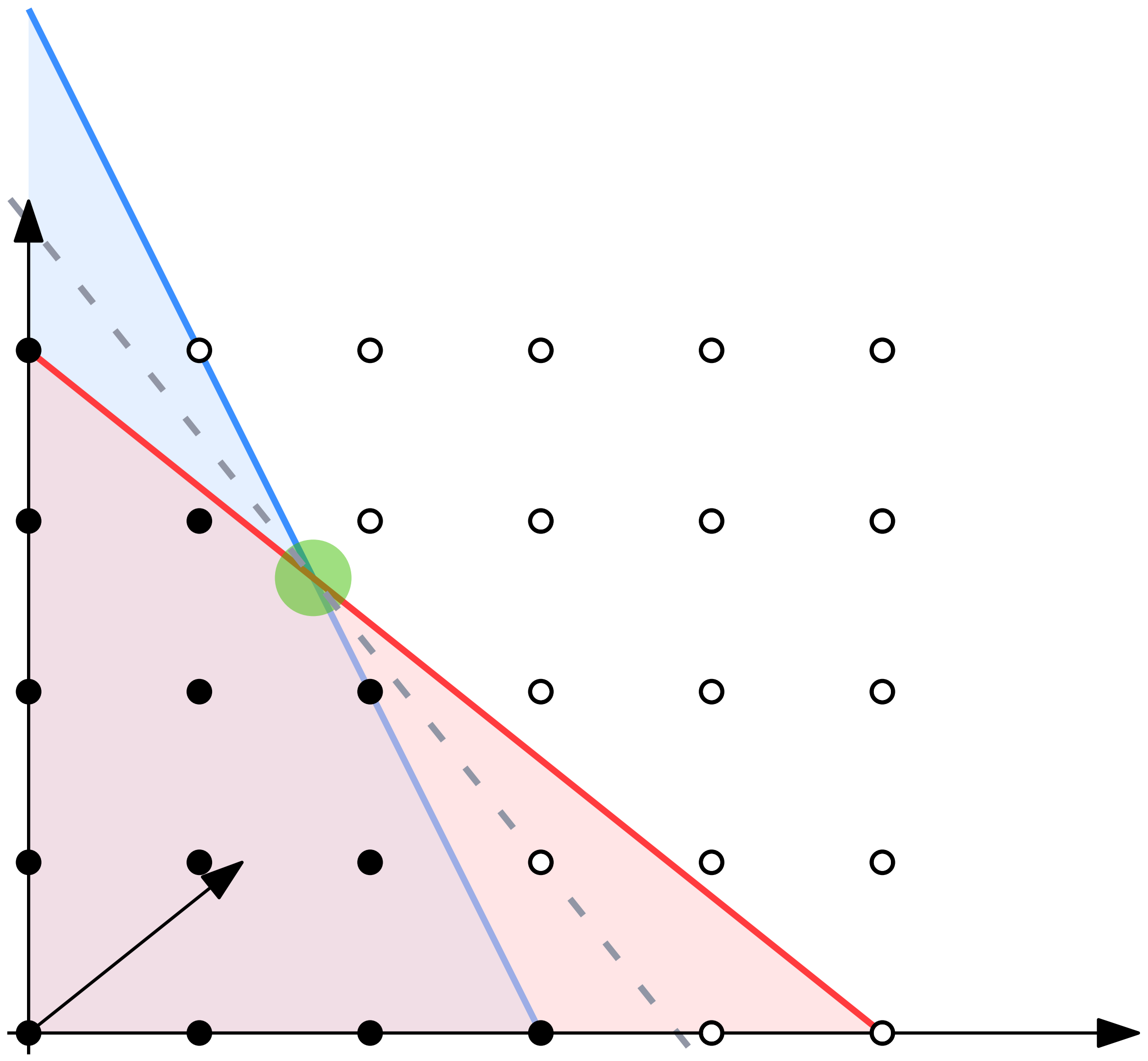
Techniques

- Branch-and-Bound
- Cutting Planes
- Branch-and-Cut



Branch and Bound for Solving ILP

Idea: branch: decompose in two subproblems
bound: discard if possible subproblems



Branch and Bound for Solving ILP

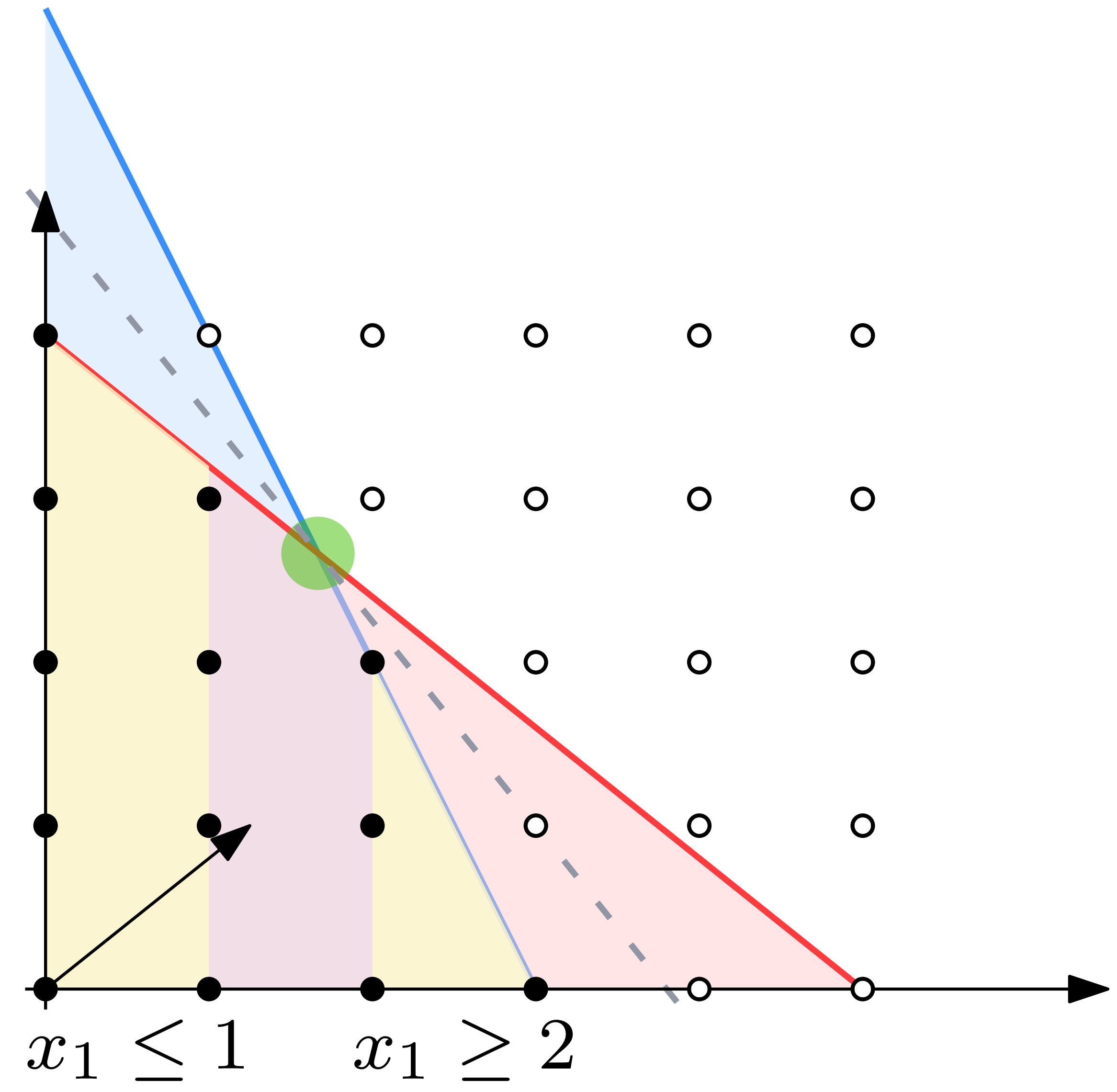
Idea: branch: decompose in two subproblems
bound: discard if possible subproblems

Sketch

- solve relaxed problem
- if solution non integer, choose variable x_i with non-integer value α_i and split into

problem P_1:
P with $x_i \leq \lfloor \alpha_i \rfloor$

problem P_2:
P with $x_i \geq \lceil \alpha_i \rceil$



Branch and Bound for Solving ILP

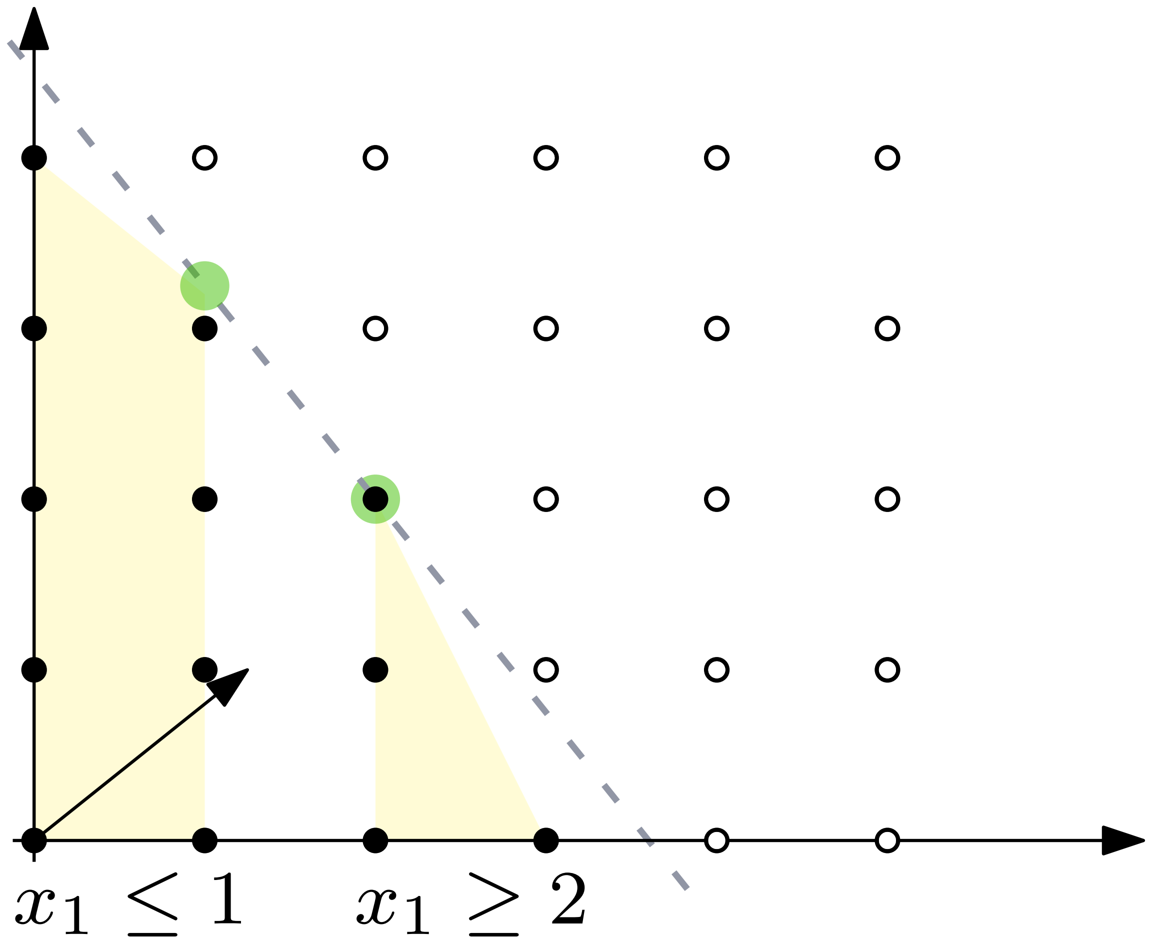
Idea: branch: decompose in two subproblems
bound: discard if possible subproblems

Sketch

- solve relaxed problem
- if solution non integer, choose variable x_i with non-integer value α_i and split into

problem P_1:
P with $x_i \leq \lfloor \alpha_i \rfloor$

problem P_2:
P with $x_i \geq \lceil \alpha_i \rceil$



Branch and Bound for Solving ILP

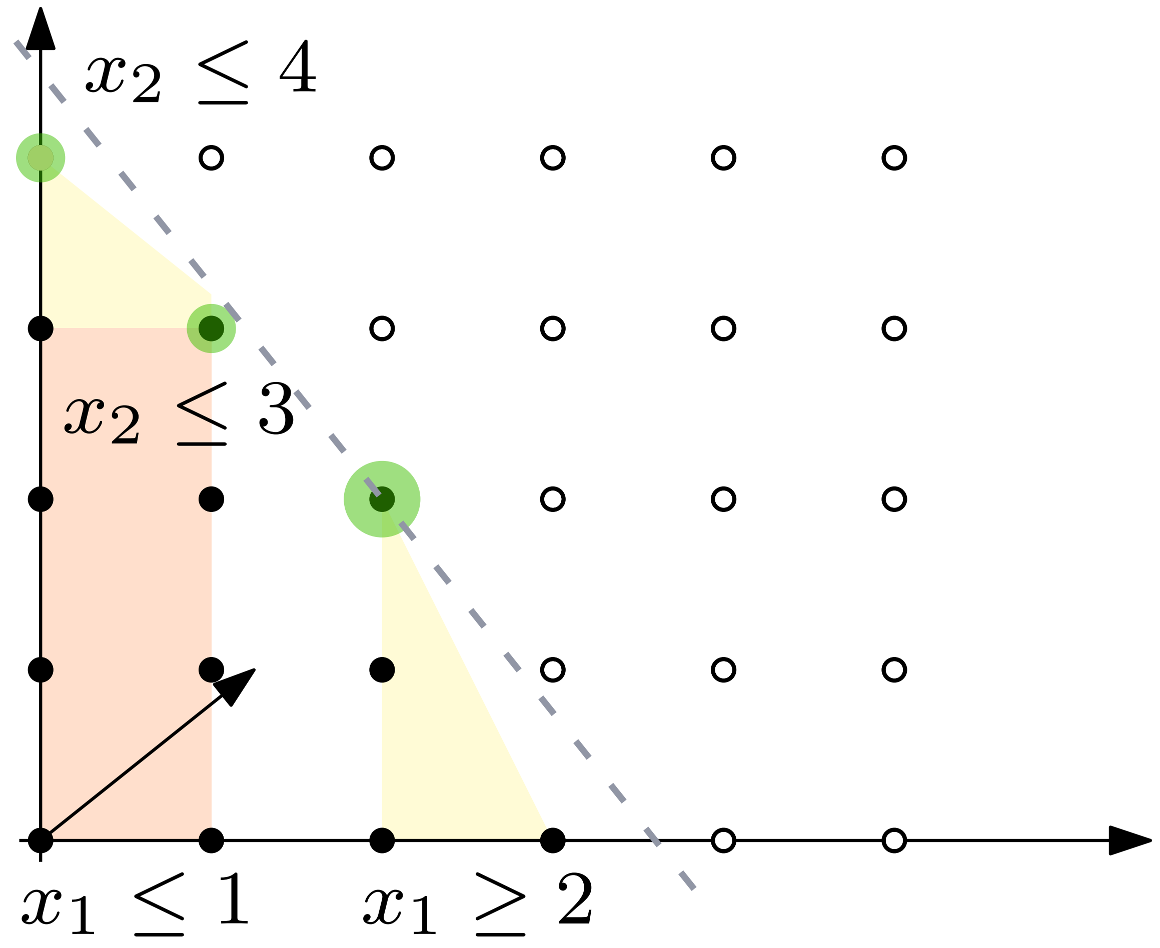
Idea: branch: decompose in two subproblems
bound: discard if possible subproblems

Sketch

- solve relaxed problem
- if solution non integer, choose variable x_i with non-integer value α_i and split into

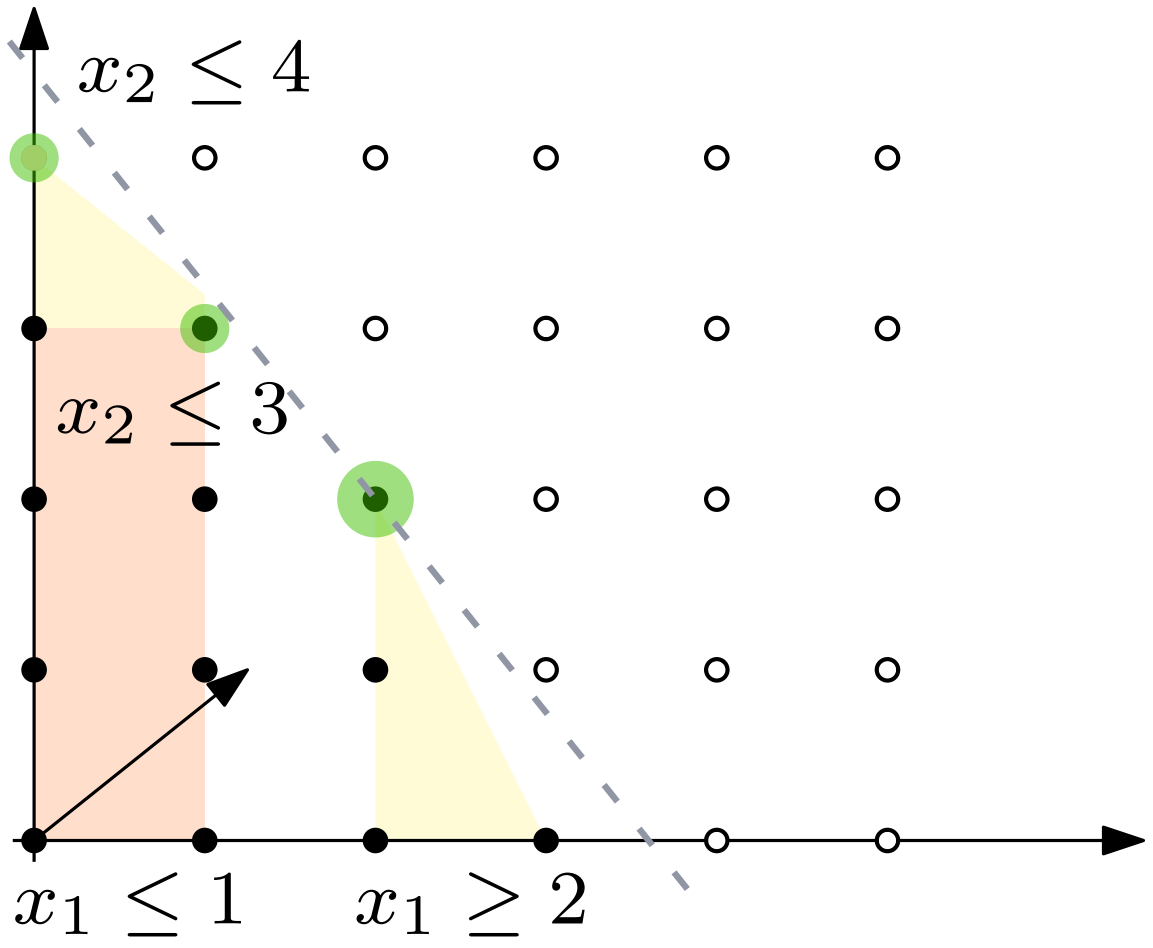
problem P_1: P with $x_i \leq \lfloor \alpha_i \rfloor$

problem P_2: P with $x_i \geq \lceil \alpha_i \rceil$



Branch and Bound for Solving ILP

Idea: branch: decompose in two subproblems
bound: discard if possible subproblems



Sketch

- solve relaxed problem
- if solution non integer, choose variable x_i with non-integer value α_i and split into

problem P_1: P with $x_i \leq \lfloor \alpha_i \rfloor$

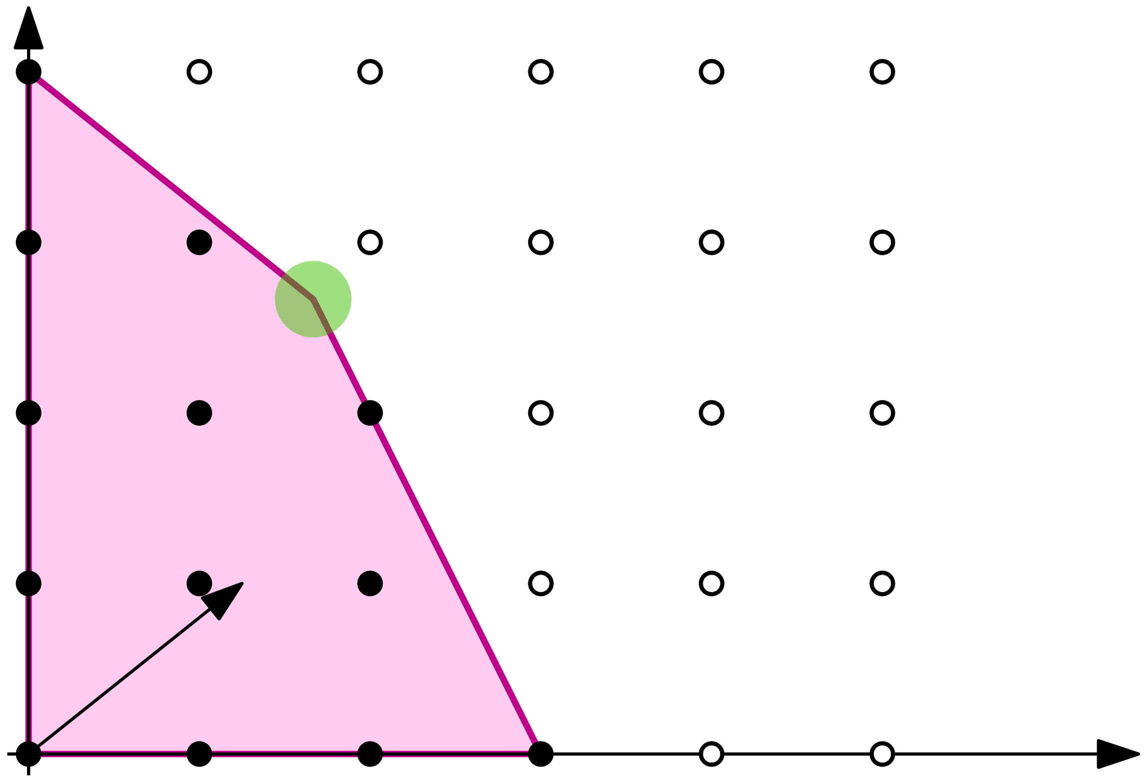
problem P_2: P with $x_i \geq \lceil \alpha_i \rceil$

three possibilities for subproblems:

- LP infeasible $\rightarrow$ discard branch
- solution integer $\rightarrow$ update OPT and discard branch
- solution non integer $\rightarrow$ stop if solution worse than OPT, else continue branching

Cutting Planes for Solving ILPs

Idea: solve relaxed LP and cut off non-integer solutions found

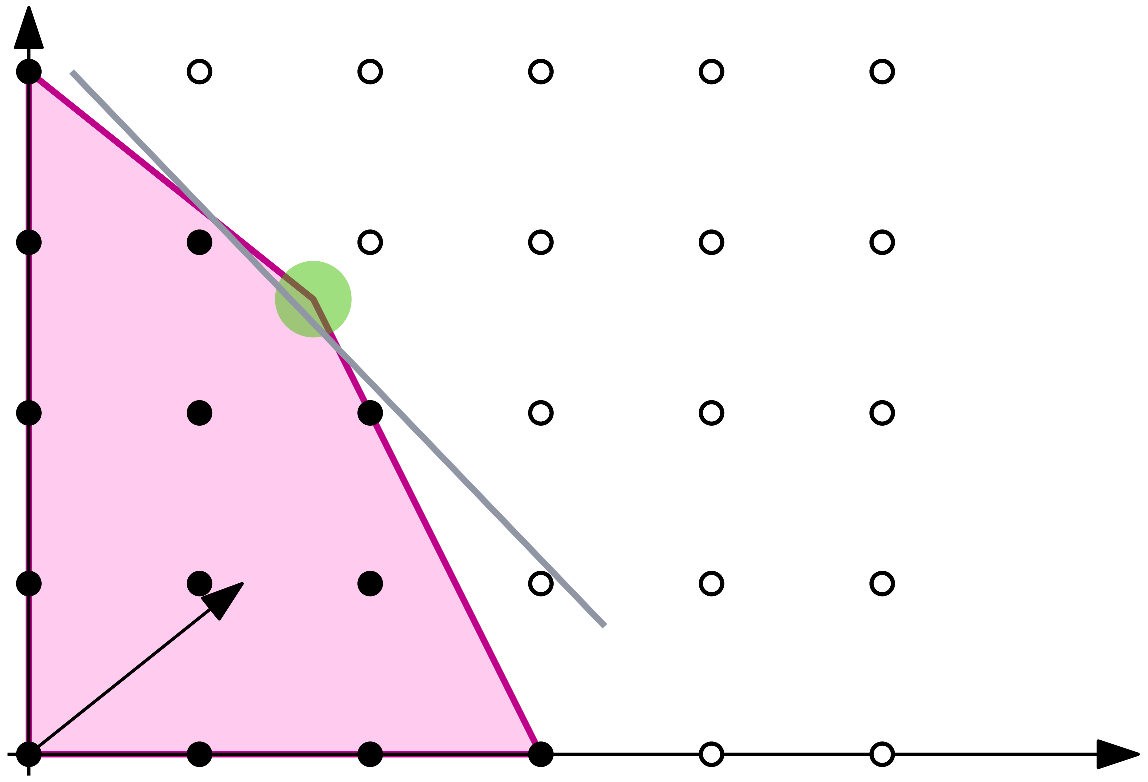


Cutting Planes for Solving ILPs

Idea: solve relaxed LP and cut off non-integer solutions found

Sketch

- solve relaxed problem
- if solution integer, return with this
- find a cut, which seperates the found solution from all integer solutions
- add the cut as constraint and restart

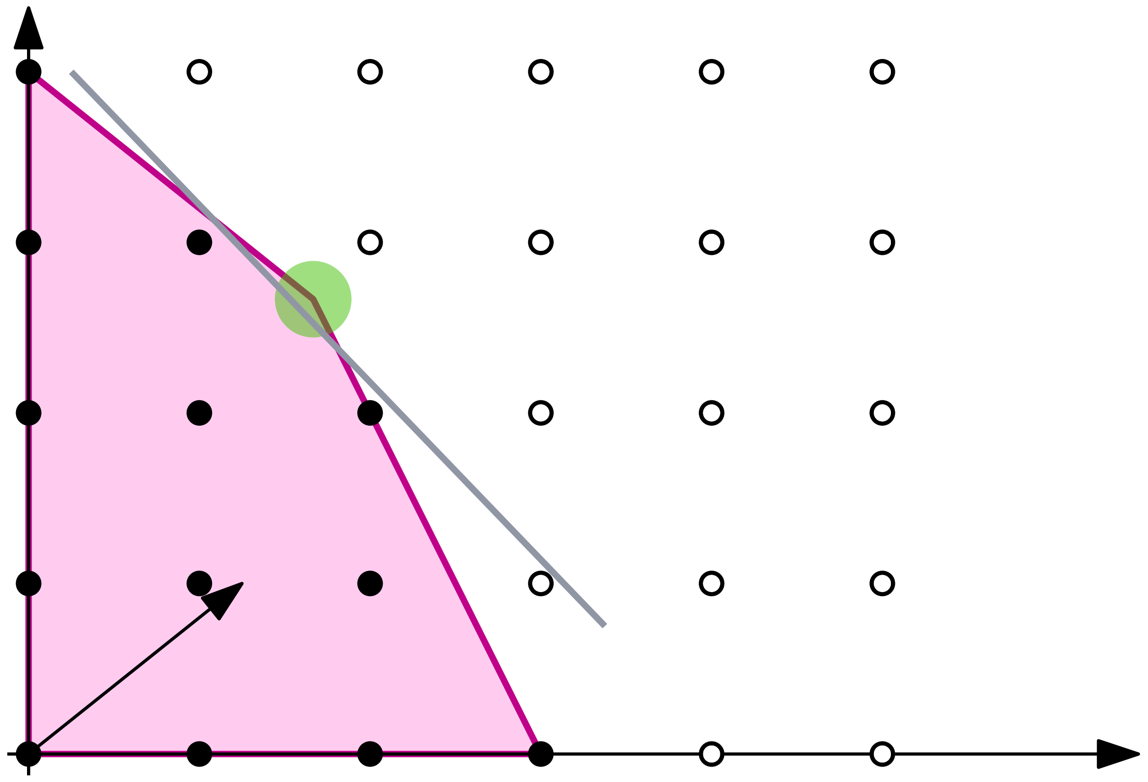


Cutting Planes for Solving ILPs

Idea: solve relaxed LP and cut off non-integer solutions found

Sketch

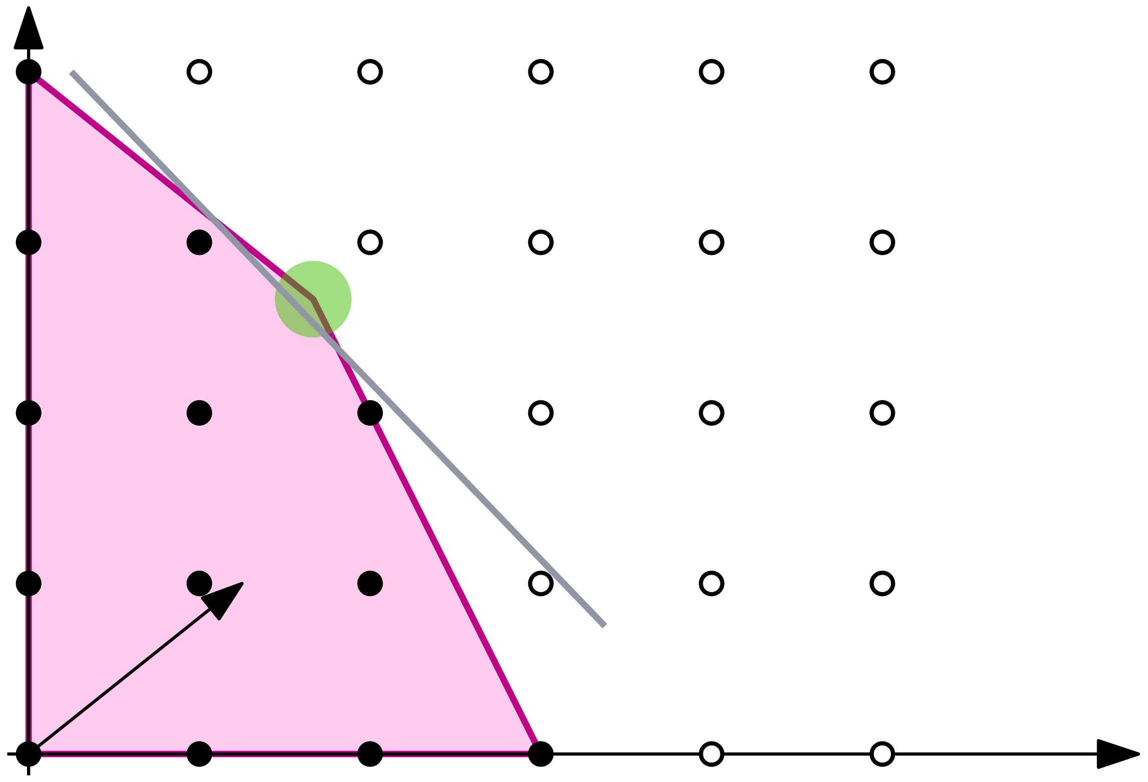
- solve relaxed problem
- if solution integer, return with this
- find a cut, which seperates the found solution from all integer solutions
- add the cut as constraint and restart



How to find a cut?

Cutting Planes for Solving ILPs

Idea: solve relaxed LP and cut off non-integer solutions found



Sketch

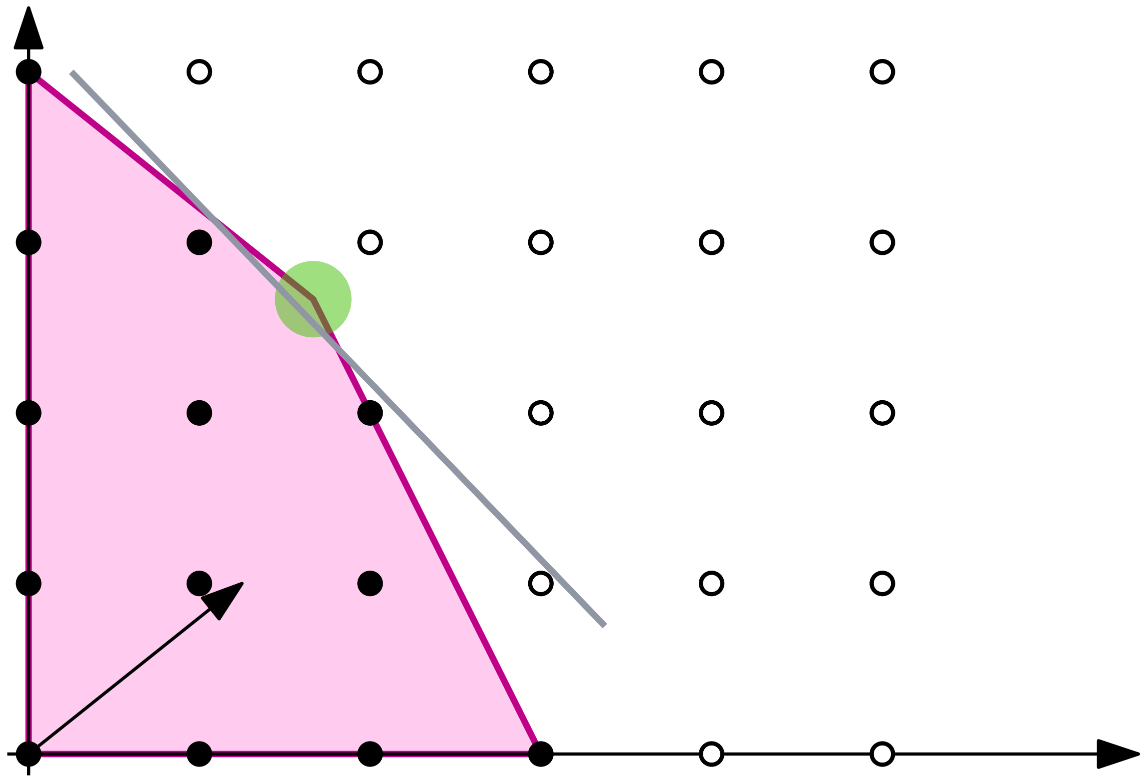
- solve relaxed problem
- if solution integer, return with this
- find a cut, which seperates the found solution from all integer solutions
- add the cut as constraint and restart

How to find a cut?

general-purpose cuts, e.g. Gomory Cuts
problem-specific cuts

Cutting Planes for Solving ILPs

Idea: solve relaxed LP and cut off non-integer solutions found



Sketch

- solve relaxed problem
- if solution integer, return with this
- find a cut, which seperates the found solution from all integer solutions
- add the cut as constraint and restart

How to find a cut?

general-purpose cuts, e.g. Gomory Cuts
problem-specific cuts

Even better is to combine both techniques:

Branch & Cut

Summary

An integer linear programm (IP) is of the form

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

Solving IPs in general is NP-hard

We formulated as IP

- Maximum weight matching
- Minimum vertex cover
- Maximum independent set
- Shortest path
- Traveling Salesperson Tour

Techniques: branch&bound, cutting planes, branch&cut

Summary

An integer linear programm (IP) is of the form

$$\begin{aligned} \text{maximize } & c^T x \\ \text{subject to } & Ax \le b \\ & x \in \mathbb{Z}^n \end{aligned}$$

Solving IPs in general is NP-hard

We formulated as IP

- Maximum weight matching
- Minimum vertex cover
- Maximum independent set
- Shortest path
- Traveling Salesperson Tour

Techniques: branch&bound, cutting planes, branch&cut

Next steps:

- basics for solving LPs
- simplex algorithm