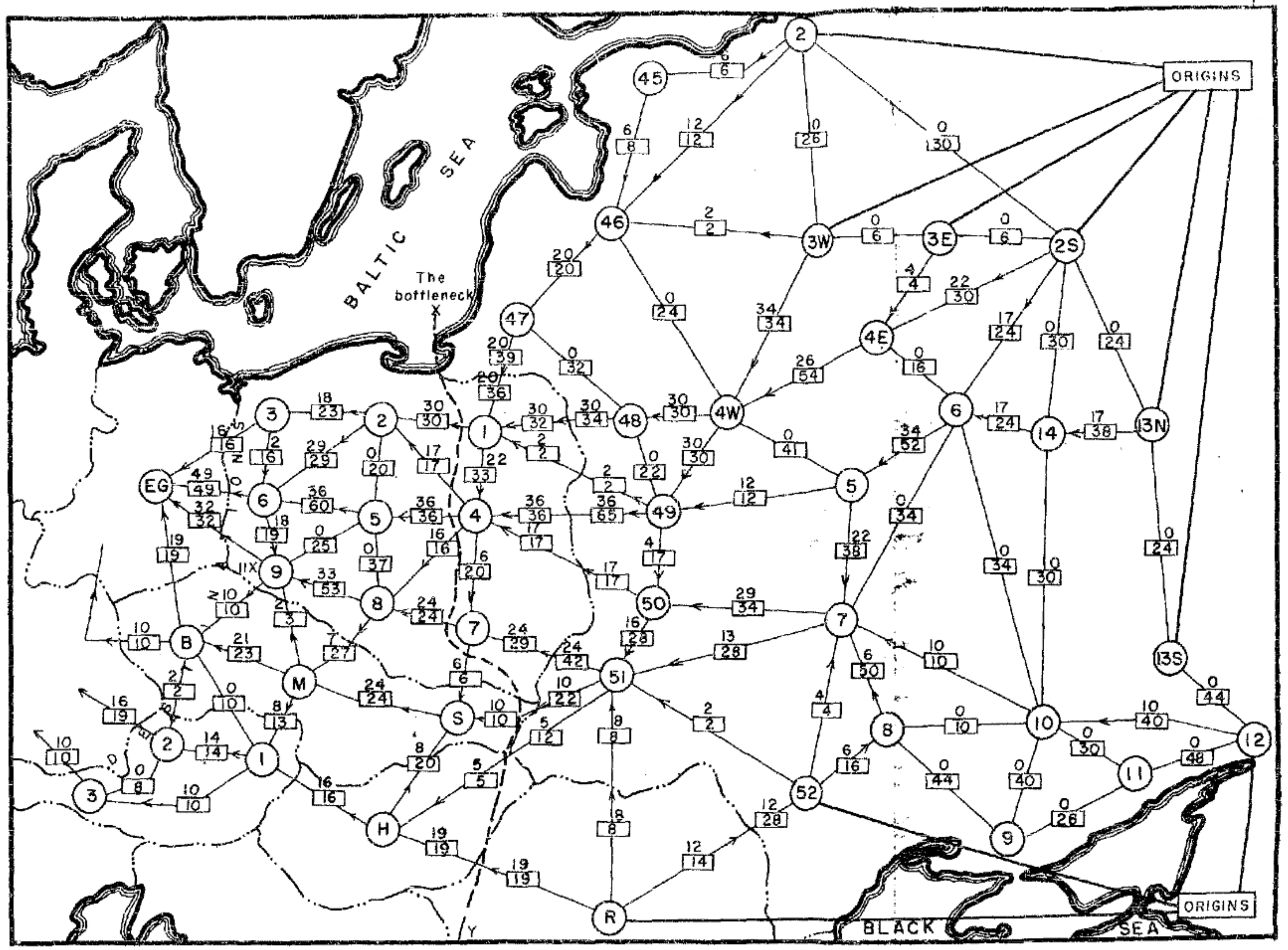
Advanced Algorithms

Network Flow & Linear Programming

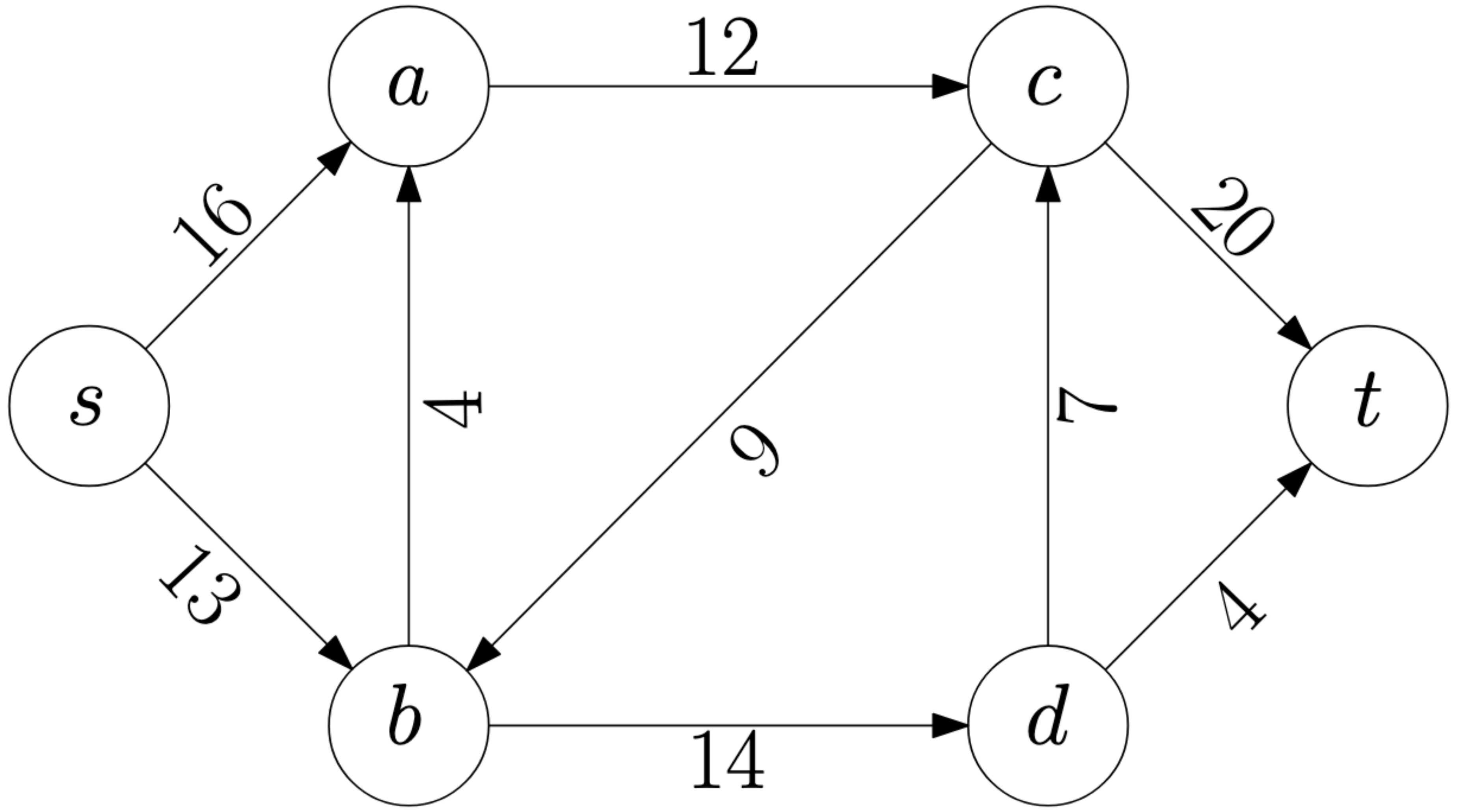
刘明谋 Nanjing University, Suzhou, 2025

Network Flow



Graph With Capacity

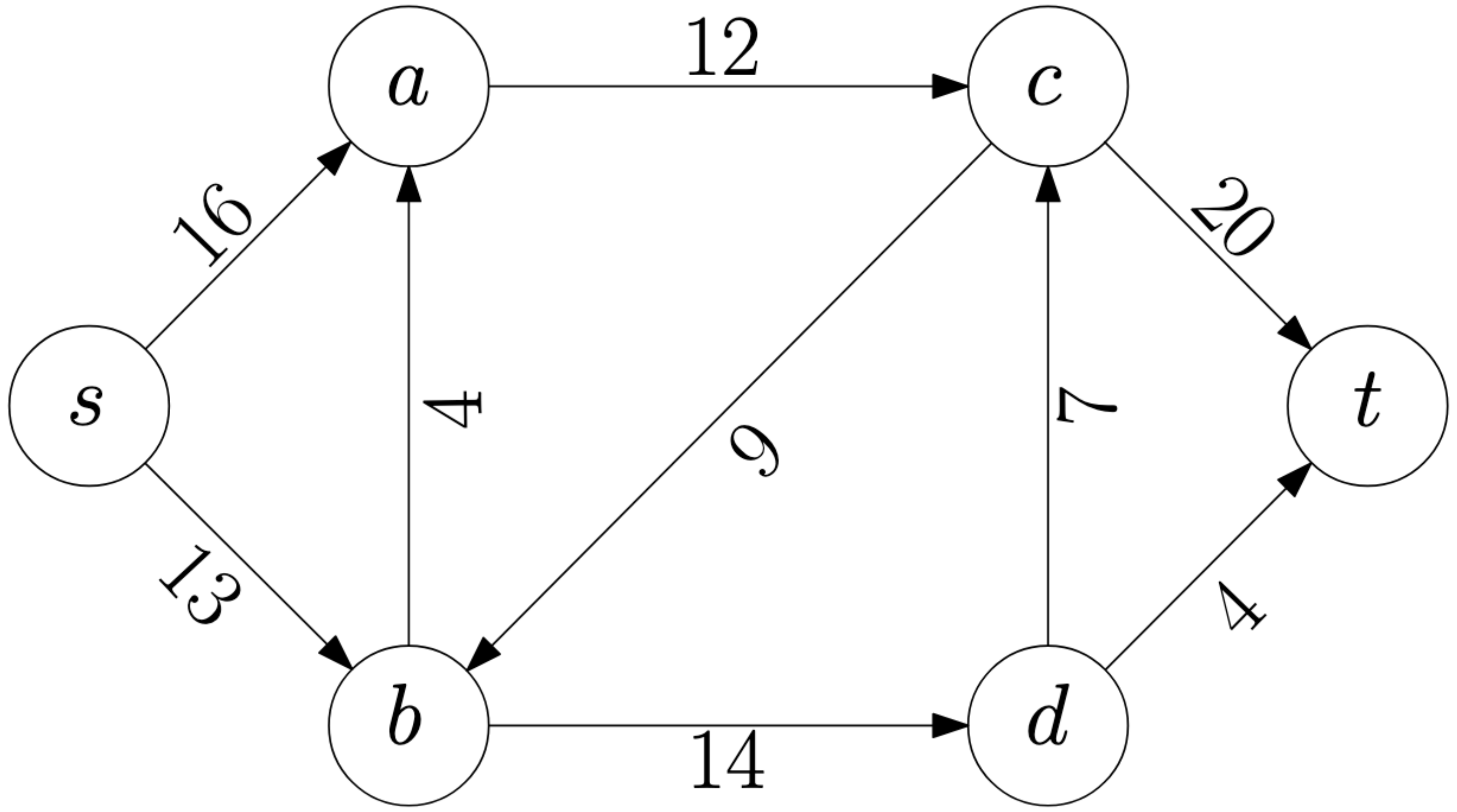
Flow



- Digraph $G = (V, E)$: a set of vertices V and a set of directed edges E
- Edge capacity $c_e \in \mathbb{R}^+$ for edge $e \in E$
- Flow function $f : E \to \mathbb{R}^+$
  - Valid flow: $\forall e \in E, 0 \leq f(e) \leq c_e$ & $\forall v \in V, \displaystyle\sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e)$

Flow Network

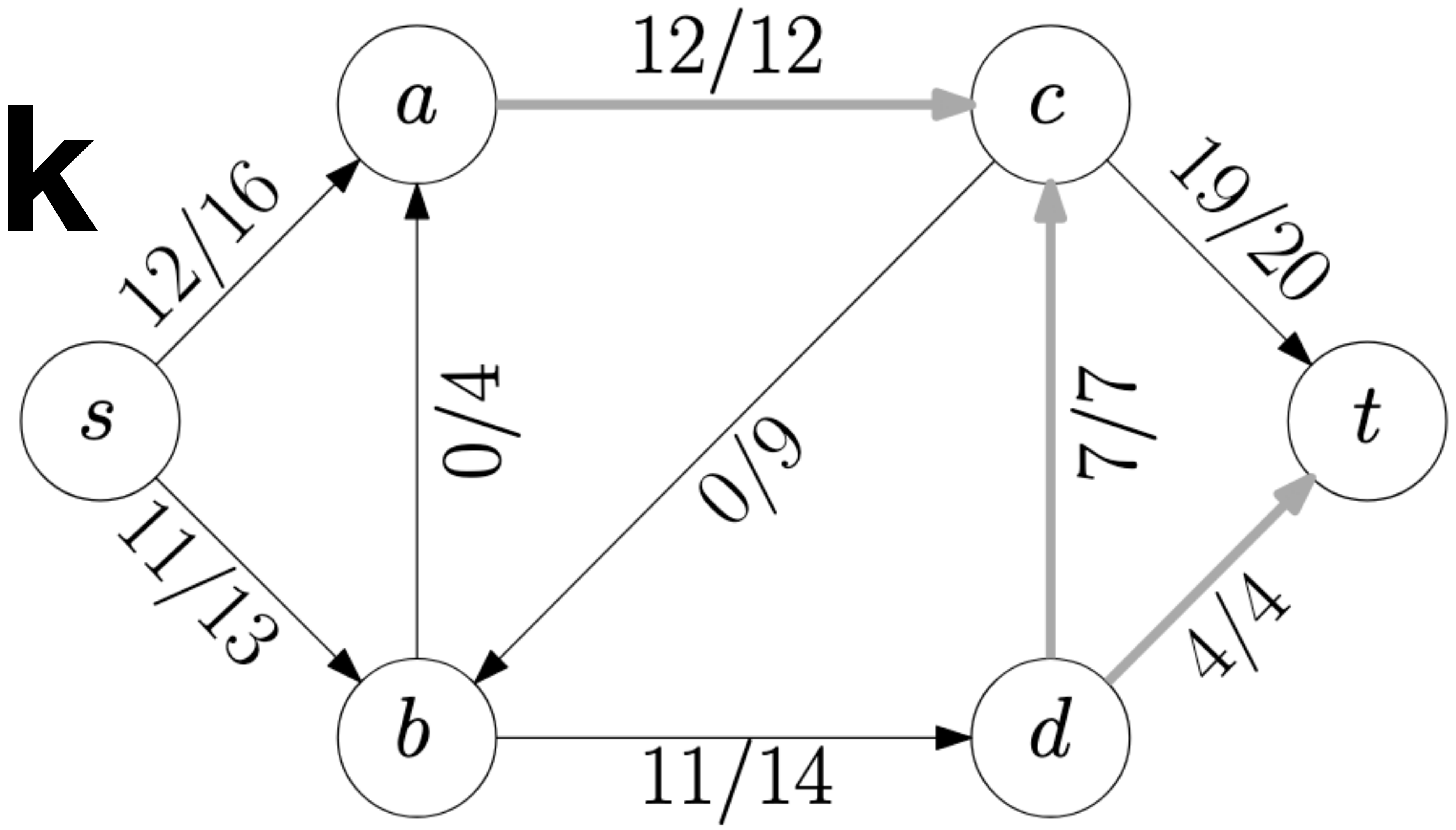
s-t flow



- Digraph $G = (V, E)$: a set of vertices V and a set of directed edges E
- Edge capacity $c_e \in \mathbb{R}^+$ for edge $e \in E$
- Flow function $f : E \to \mathbb{R}^+$, source $s \in V$, sink $t \in V$
  - Valid flow: $\forall e \in E, 0 \leq f(e) \leq c_e$ & $\forall v \in V \backslash \{s, t\}, \sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e)$

Flow Network

Maximum flow



- Digraph $G = (V, E)$: a set of vertices V and a set of directed edges E
- Edge capacity $c_e \in \mathbb{R}^+$ for edge $e \in E$
- Flow function $f : E \to \mathbb{R}^+$, source $s \in V$, sink $t \in V$
  - Valid flow: $\forall e \in E, 0 \le f(e) \le c_e$ & $\forall v \in V \backslash \{s, t\}$, $\sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e)$
- Maximum flow problem: find a flow f maximizes $\sum_{e \in \delta_{out}(s)} f(e)$

Army Transportation

SECRET RM-1573 10-24-55 -33-

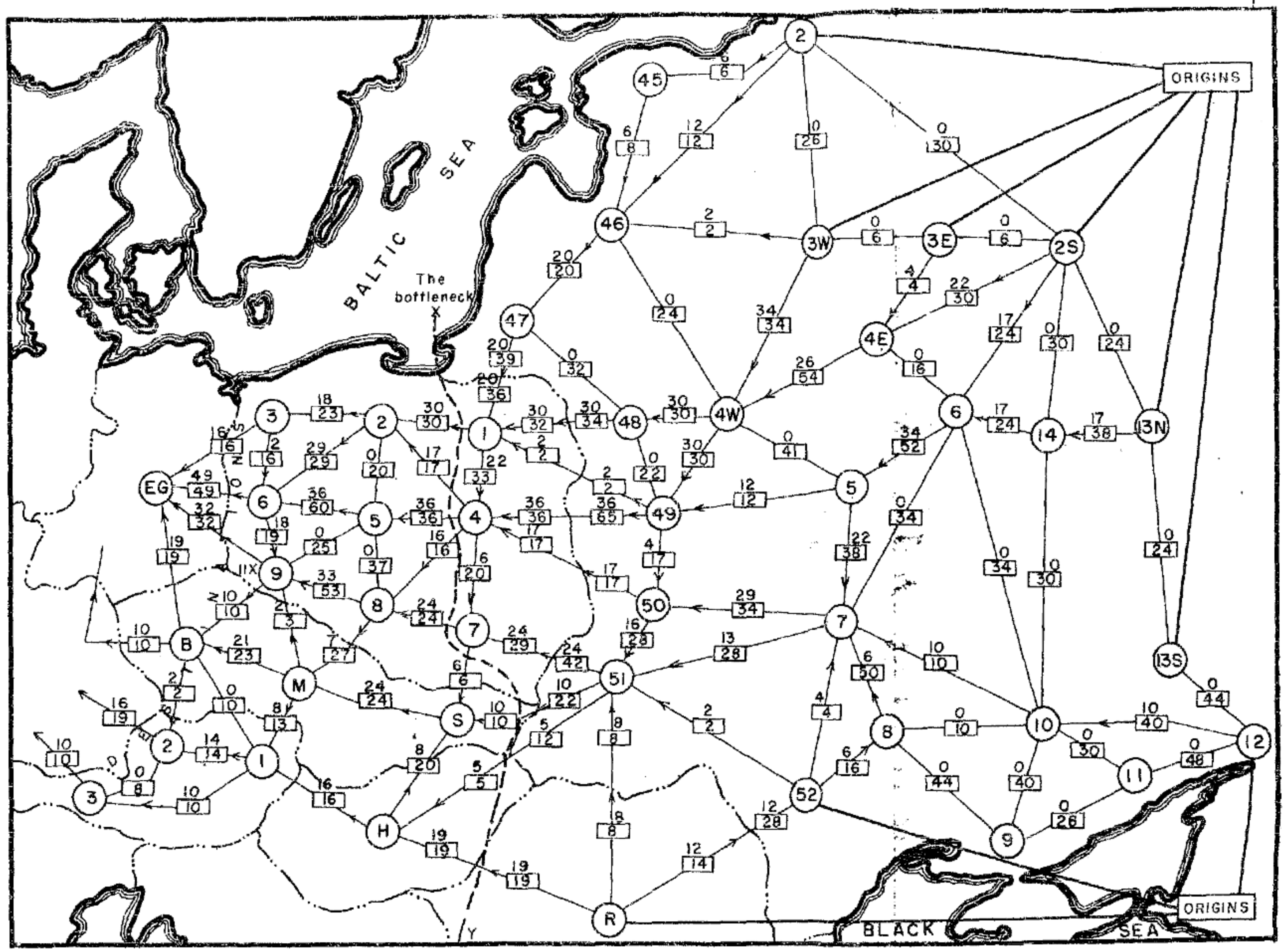


Fig. 7 — Traffic pattern: entire network available

Legend:

—··— International boundary

(8) Railway operating division

9 / [12] Capacity: 12 each way per day. Required flow of 9 per day toward destinations (in direction of arrow) with equivalent number of returning trains in opposite direction

All capacities in trains / √1000's of tons each way per day

Origins: Divisions 2, 3W, 3E, 2S, 13N, 13S, 12, 52 (USSR), and Roumania

Destinations: Divisions 3, 6, 9 (Poland); B (Czechoslovavakia); and 2, 3 (Austria)

Alternative destinations: Germany or East Germany

Note IIX at Division 9, Poland

Assumption:

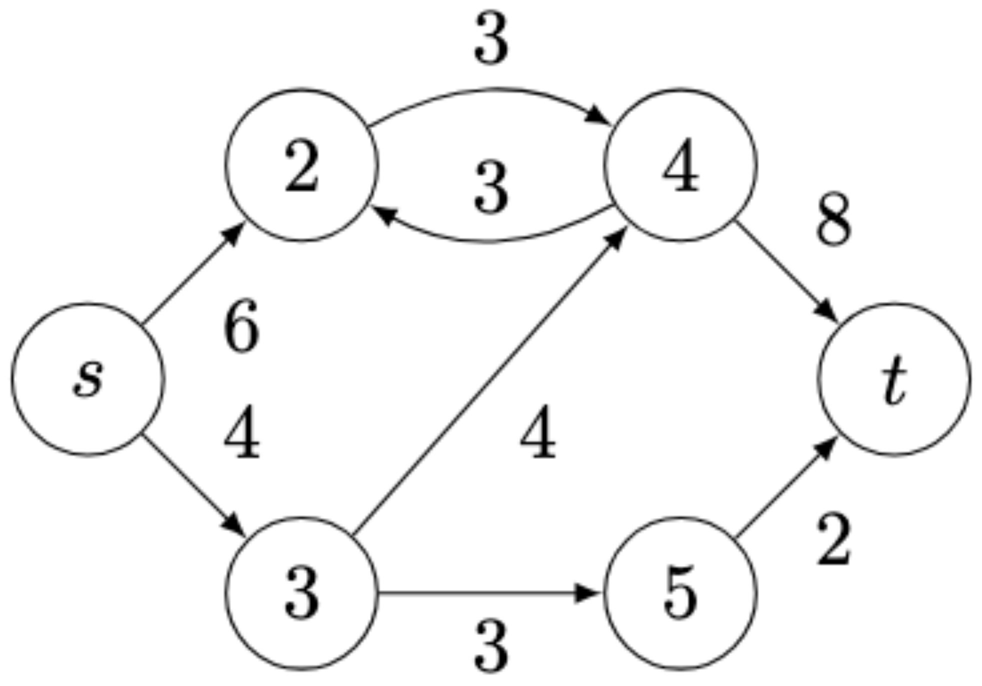
Entire network available for east—west traffic (no allowance for civilian or economic traffic)

Results:

(a) 163,000 tons per day can be delivered from points of origin to destinations.
(b) 147,000 tons per day can be delivered without using Austrian lines.
(c) 152,000 tons per day can be delivered into Germany by all lines.
(d) 126,000 tons per day can be delivered into East Germany without using Austrian lines.

Flow Decomposition

Maximum flow consists of paths



- Theorem: Any s-t flow can be decomposed into cycles and s-t paths.
- Constructive proof:
  - While s has out flow: exists valid s-t path since $\sum_{e \in \delta_{out}(s)} f(e) = \sum_{e \in \delta_{in}(t)} f(e)$
    - find s-t path $P \subseteq E$; let $\delta(P) \triangleq \min_{e \in P} f_e$; update $\forall e \in P, f_e \leftarrow f_e - \delta(P)$
  - While exists flow: exists flow cycles since $\sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e), \forall v$
    - find flow cycle $P \subseteq E$, remove it as well

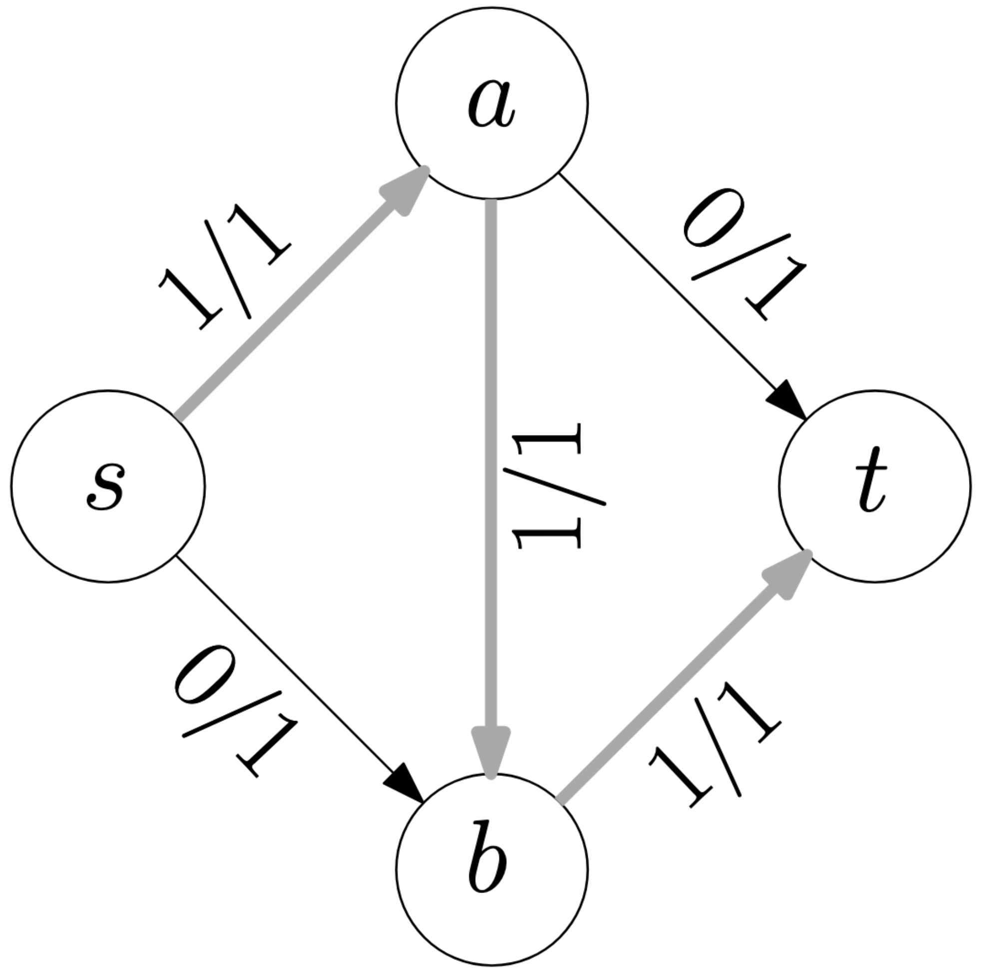
Augmenting Path

Maximum flow consists of paths

- Theorem: Any s-t flow can be decomposed into cycles and s-t paths.

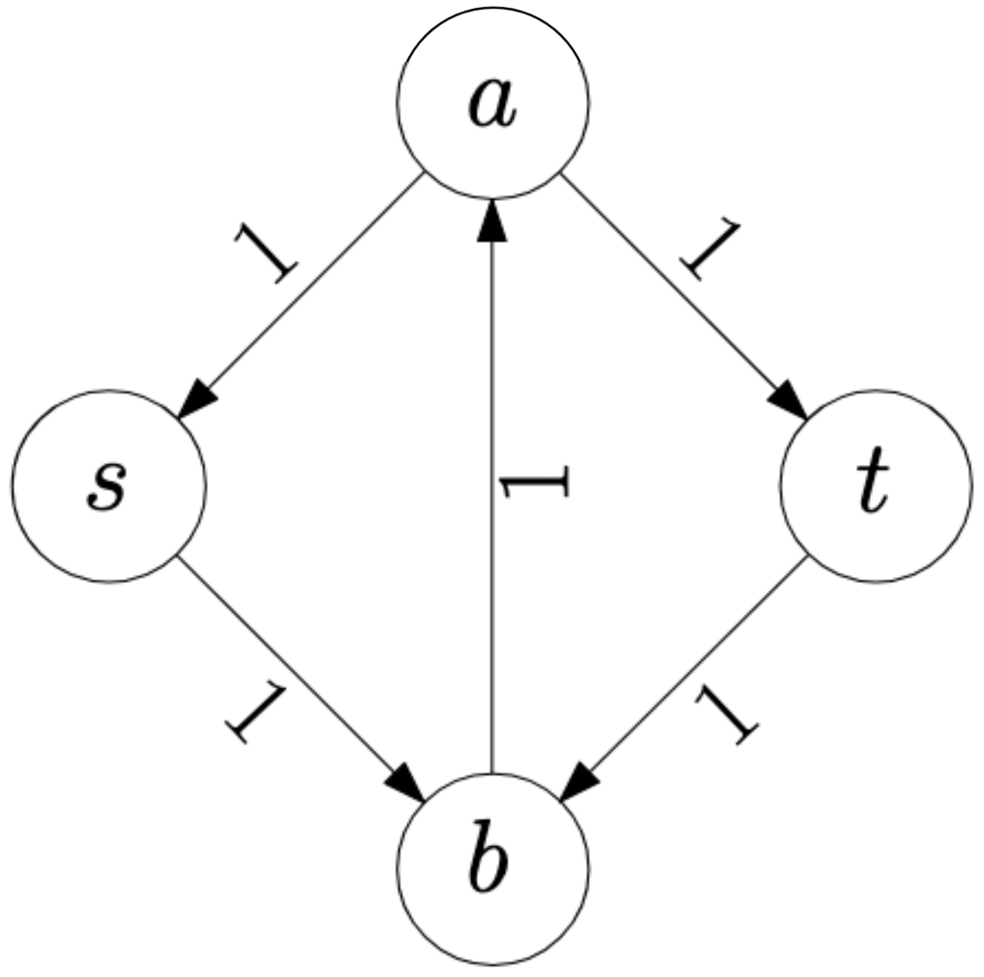
> **Greedy**: let $\delta(P) \triangleq \min_{e \in p} c_e - f_e$ be residual capacity
>
> While exists s-t path P with $\delta(P) > 0$:
>
> increase the flow in P by $\delta(P)$

- Any issue?



Observation:
implicit augmenting path

Residual graph

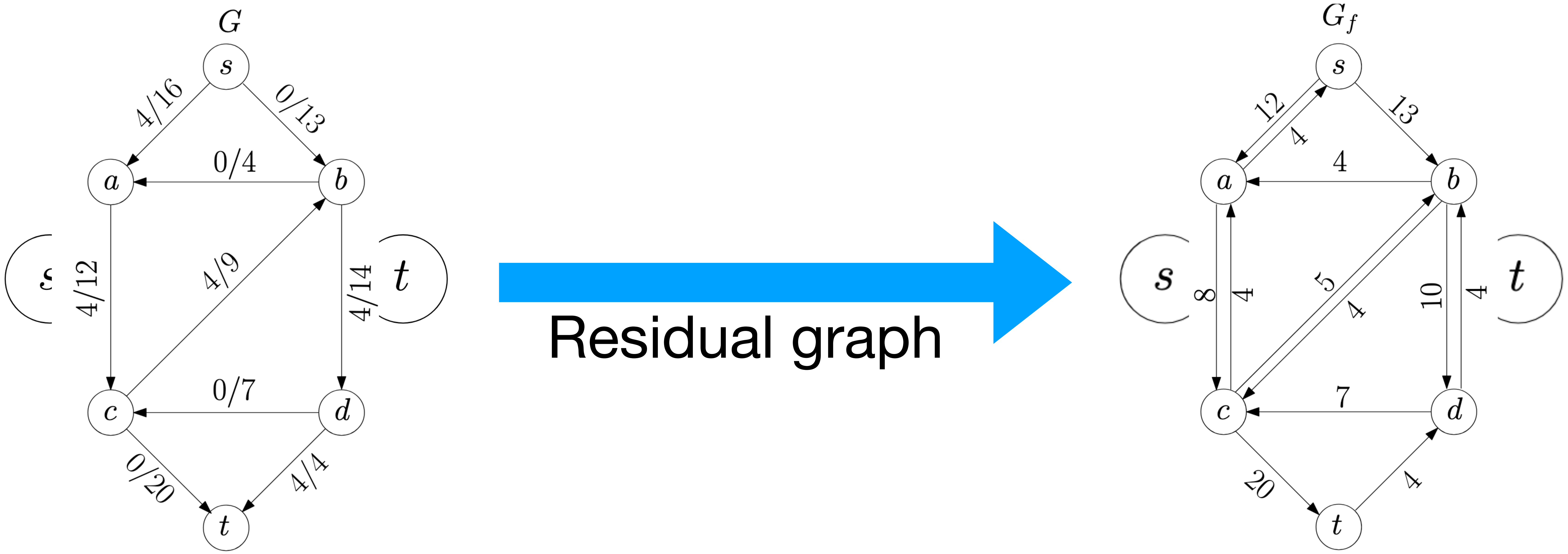


Augmenting Path

Fixing with residual graph

- Theorem: Any s-t flow can be decomposed into cycles and s-t paths.

- Residual graph $G_f = (V, E')$ of $G = (V, E)$ with capacity c and flow f:
  - for each $e = (u, v) \in E$ with $c_e > f_e$, add e to E' with $c'_e = c_e - f_e$
  - for each $e = (u, v) \in E$ with $f_e > 0$, add (v, u) to E' with $c'_{e'} = f_e$



Augmenting Path

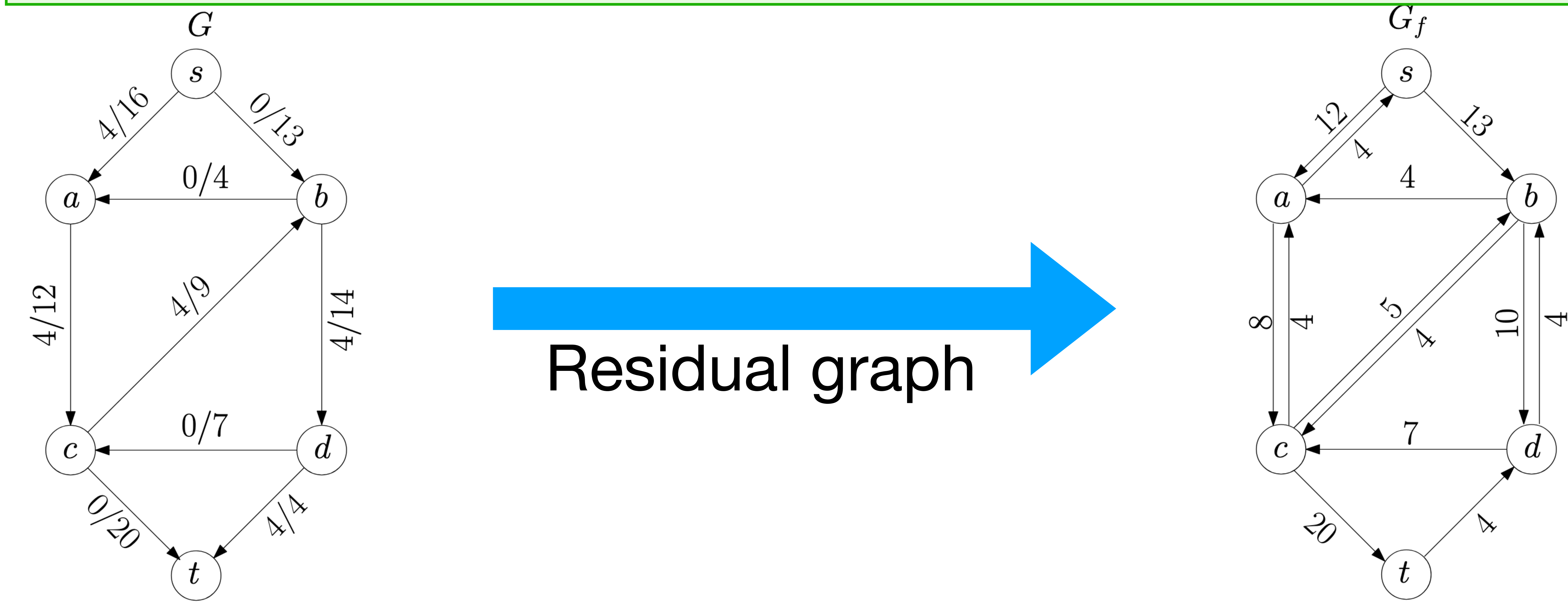
Fixing with residual graph

- Theorem: Any s-t flow can be decomposed into cycles and s-t paths.

Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

increase the flow f in P by $\delta(P)$



Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

increase the flow f in P by $\delta(P)$

The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

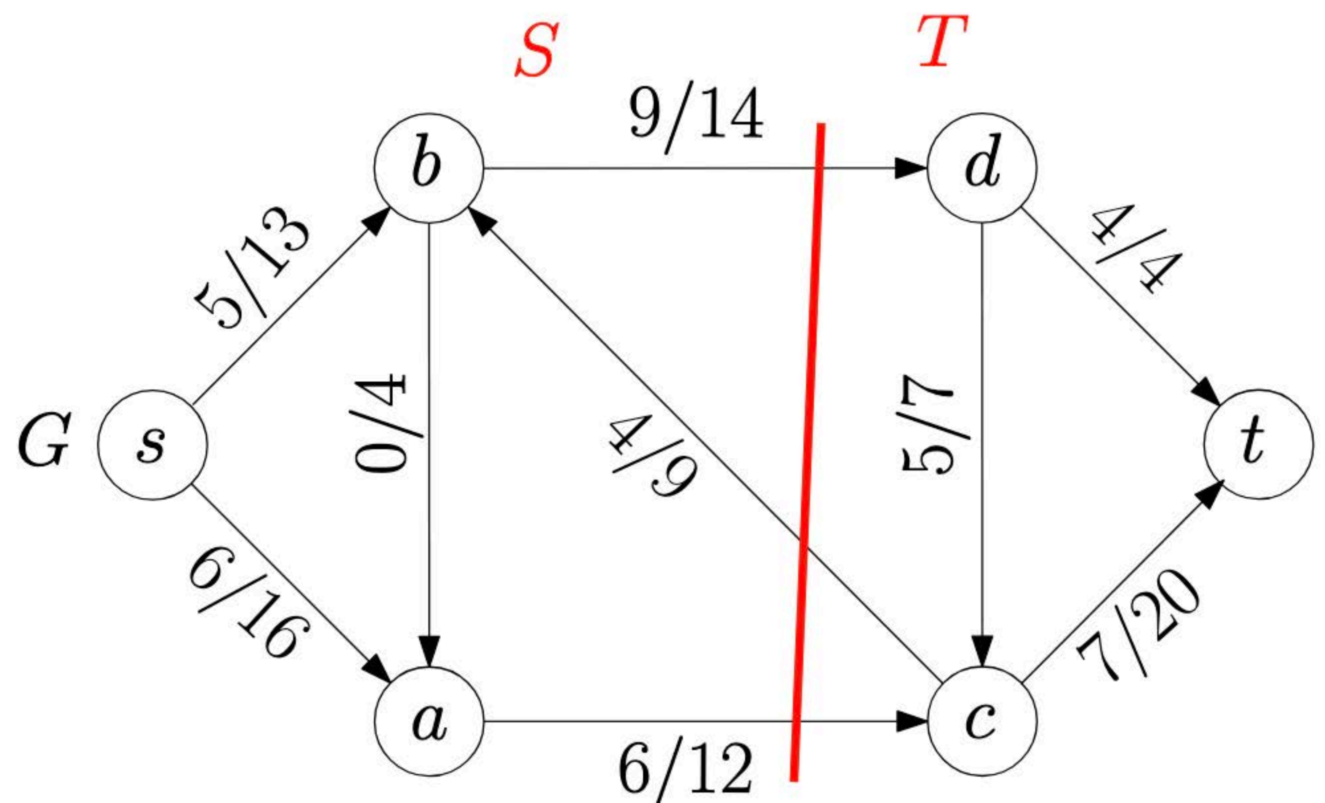
(3) There is $S, \neg S \subseteq V$ such that cut$(S) = |f|$

Proof strategy:

(1)⟹(2): proven by ¬(2)⟹¬(1)

(2)⟹(3): every edge is stuck

(3)⟹(1): every flow is at most any cut

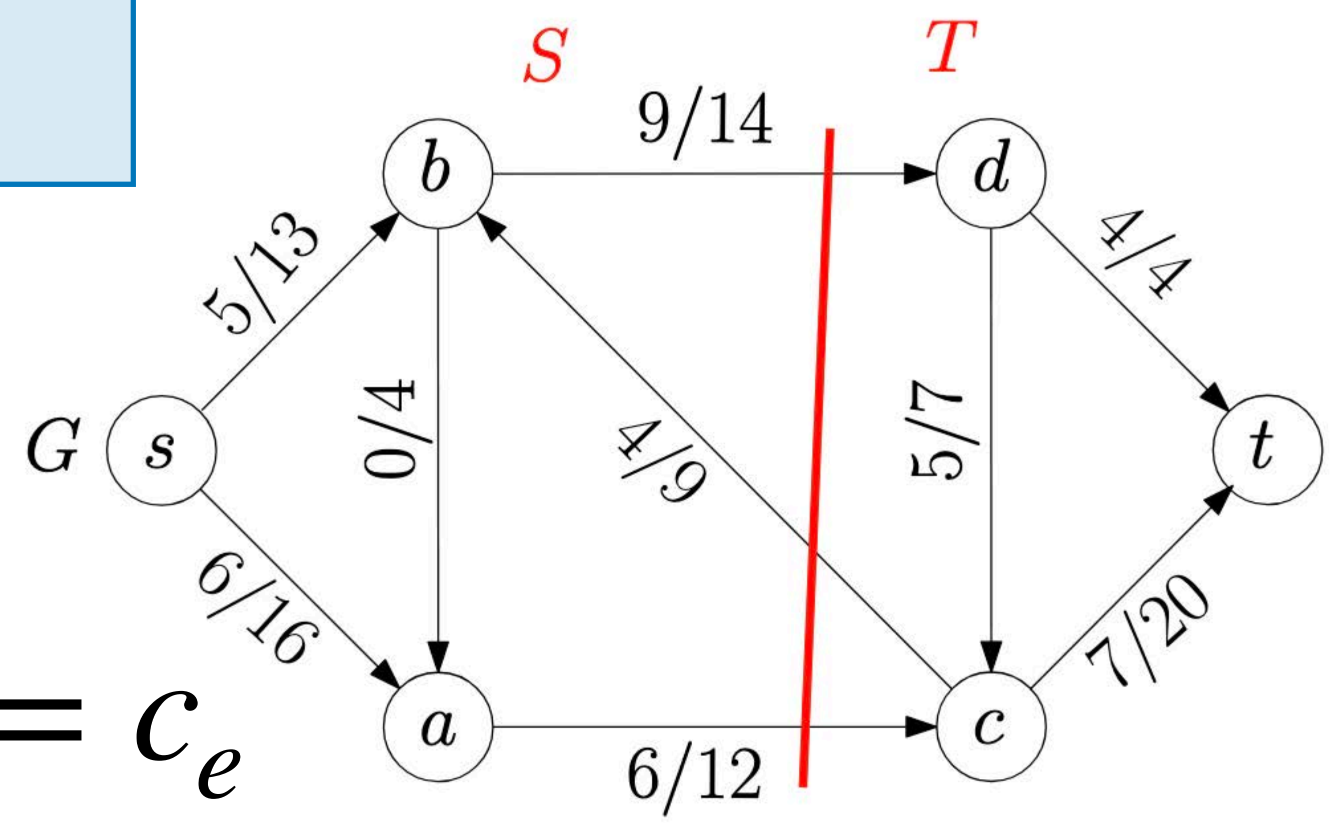


The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

(3) There is $S, \neg S \subseteq V$ such that $\mathrm{Cut}(S) = |f|$



(2)$\Longrightarrow$(3): Let S be set of vertices reachable in G^f from s

- Observation:
  - Every edge $e = (a,b) \in E$ with $a \in S, b \in \neg S$, then $f_e = c_e$
  - Every edge $e = (b,a) \in E$ with $a \in S, b \in \neg S$, then $f_e = 0$
- Otherwise b is reachable in G^f as well.
- By flow conservation:

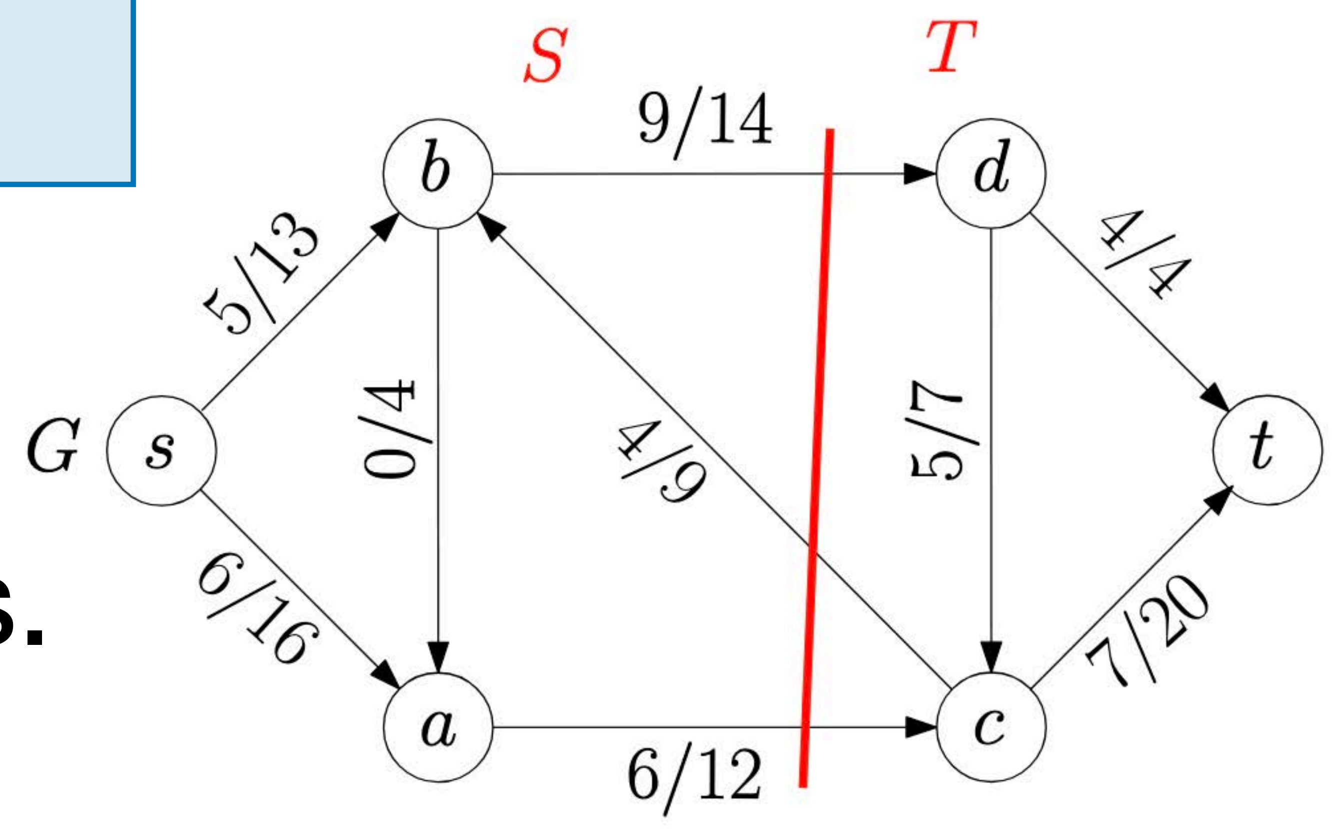
$$|f| = \sum_{a \in S, b \notin S} f_{a,b} - \sum_{a \in S, b \notin S} f_{b,a} = \sum_{\substack{(a,b) \in E \\ a \in S, b \notin S}} c_{a,b} = \mathrm{Cut}(S)$$

The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

(3) There is $S, \neg S \subseteq V$ such that Cut$(S) = |f|$



(3)$\Longrightarrow$(1): $|f| \leq$ Cut(S) for any f, S

- Proof by decomposition: decompose f, keep only s-t paths.
- Consider a path P with p flow.
  - Pass from S to $\neg S$
  - Uses $\geq p$ capacity of Cut(S) (backward is allowed)
- $|f| = \sum_P p \leq$ Cut(S)

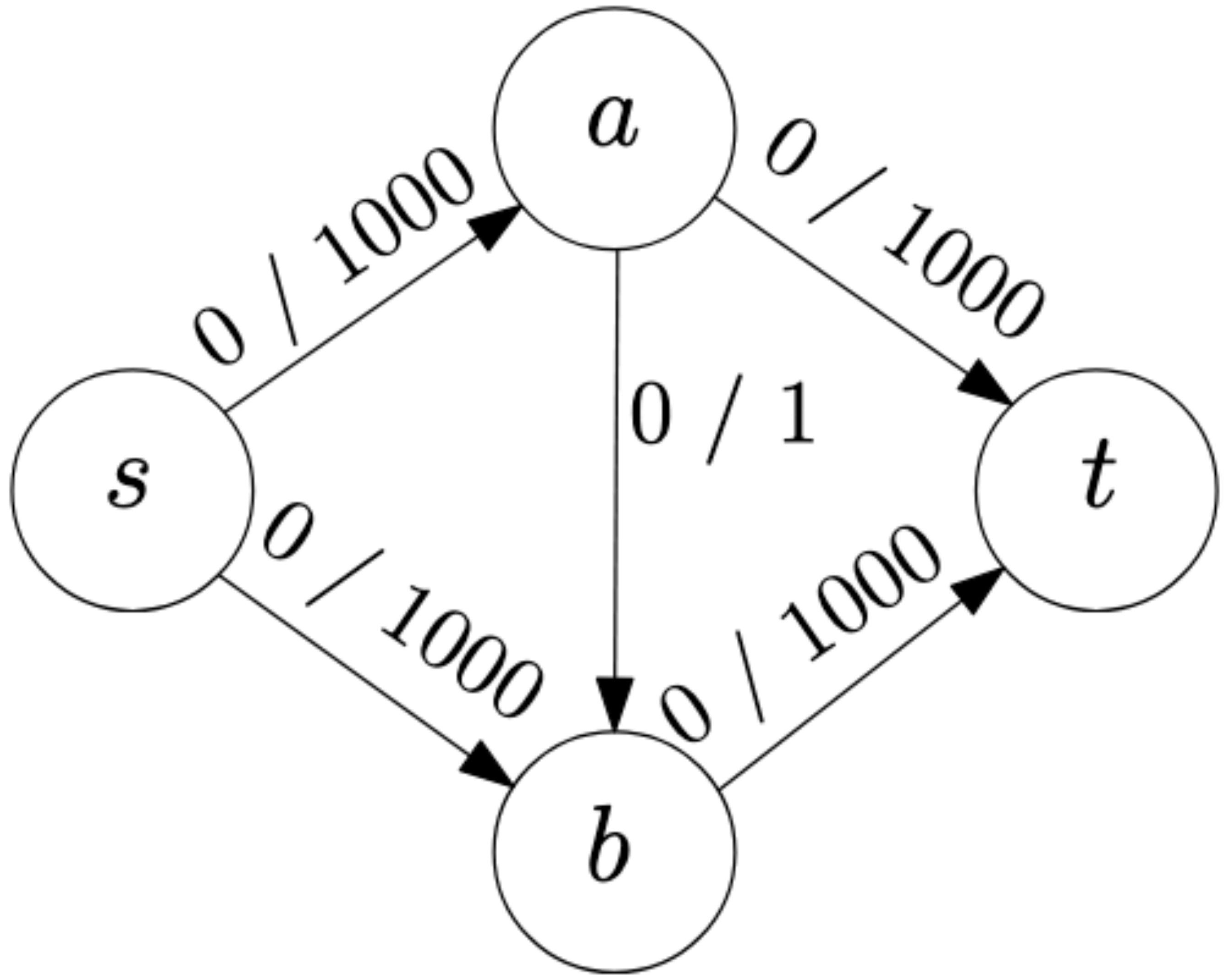
Augmenting Path

Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

increase the flow f in P by $\delta(P)$

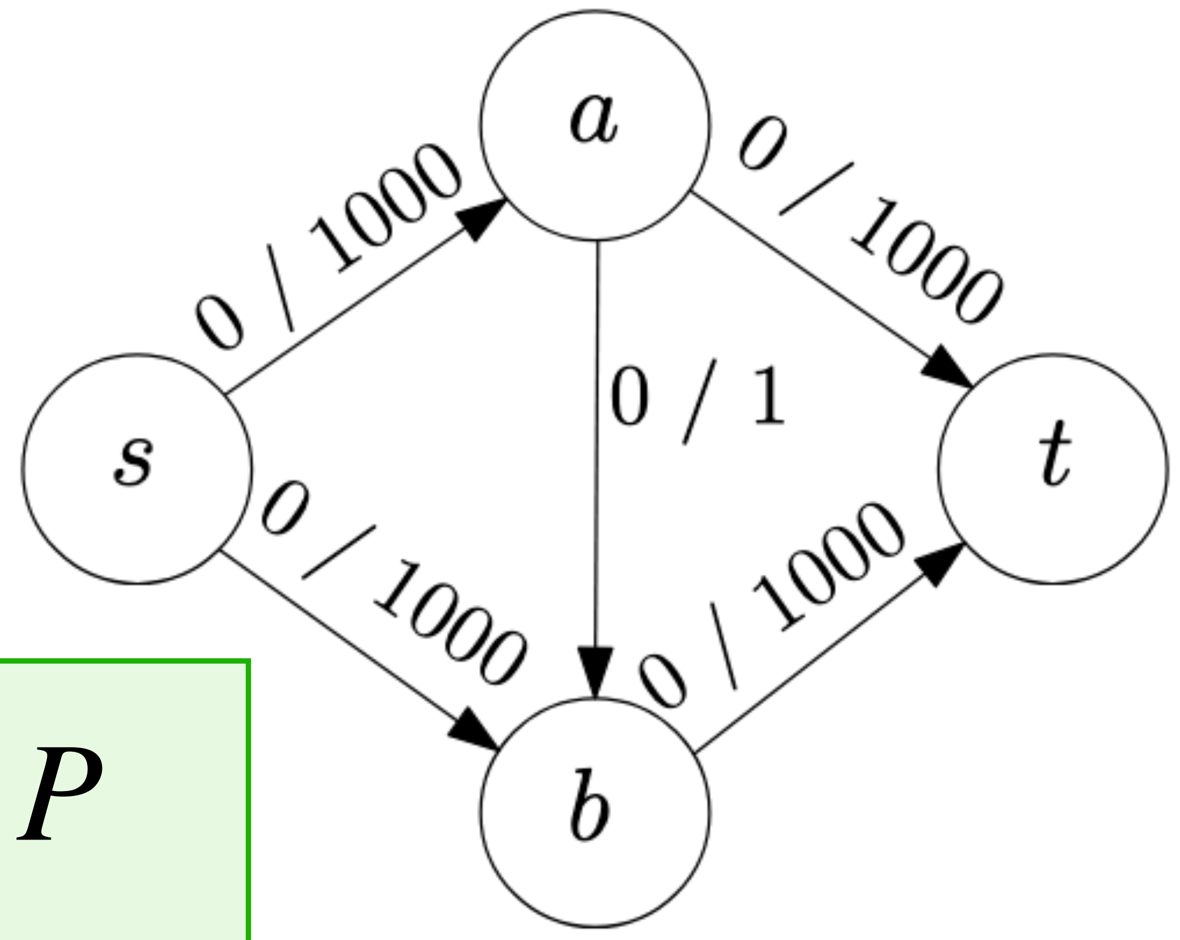
- Time cost? Worst case.



- Naive Ford-Fulkerson algorithms takes $O(|f^*| \cdot m)$ time

Augmenting Path

Greed



Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

increase the flow f in P by $\delta(P)$

- Theorem: Any s-t flow can be decomposed into cycles and s-t paths.
- Greedy idea: adopt the "fattest" augmenting path
- Max flow $|f^*| = |f| + |f'|$, where f' is the max flow in G^f
- By decomposition, exists a path P with flow $\geq |f'|/m$, since $\leq m$ paths
- Convergence: $|f'| \leq |f^*|(1 - 1/m)^t$ after t iterations.
  - #iterations $= O(m \log |f^*|)$. Might not converge for real capacity.

Augmenting Path

Greed

Observation:
augmenting fattest path
results in **thinner** path

Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

increase the flow f in P by $\delta(P)$

- Greedy idea: adopt the "fattest" augmenting path
- Max flow $|f^*| = |f| + |f'|$, where f' is the max flow in G^f
- By decomposition, exists a path P with flow $\geq |f'|/m$, since $\leq m$ paths
- Convergence: $|f'| \leq |f^*|(1 - 1/m)^t$ after t iterations.
  - #iterations $= O(m \log |f^*|)$. Might not converge for real capacity.
- Finding "fat" paths: binary search + BFS in $O(m \log |f^*|)$ time

Augmenting Path

Greed + Scaling

Observation:
augmenting shortest path
results in **longer** path

Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P
While exists s-t path $P \in G^f$ with $\delta(P) > 0$:
increase the flow f in P by $\delta(P)$

- Greedy idea: for $i = \lceil \log_2 C \rceil, \ldots, 0$: while exists flow $|f| \geq 2^i$: augment
- Imagine residual graph $G_i^f = (V, E'', f)$ with $E'' \triangleq \{e \in E' : c_e \geq 2^i\}$
  - Any edge e is of capacity $c_e \in [2^i, 2^{i-1})$
  - All augmenting paths contribute $\geq 2^i$
  - #flows in G_i^f is $\leq m$ by decomposition
- Inner loop repeats $O(m)$ times. Time cost $O(\log_2 C \cdot m \cdot m)$

Augmenting Path

Shortest augmenting [Dinitz, Edmonds-Karp]

Observation:
augmenting shortest path
results in **longer** path

Ford-Fulkerson: let $\delta(P)$ be residual capacity for path P

While exists s-t path $P \in G^f$ with $\delta(P) > 0$:

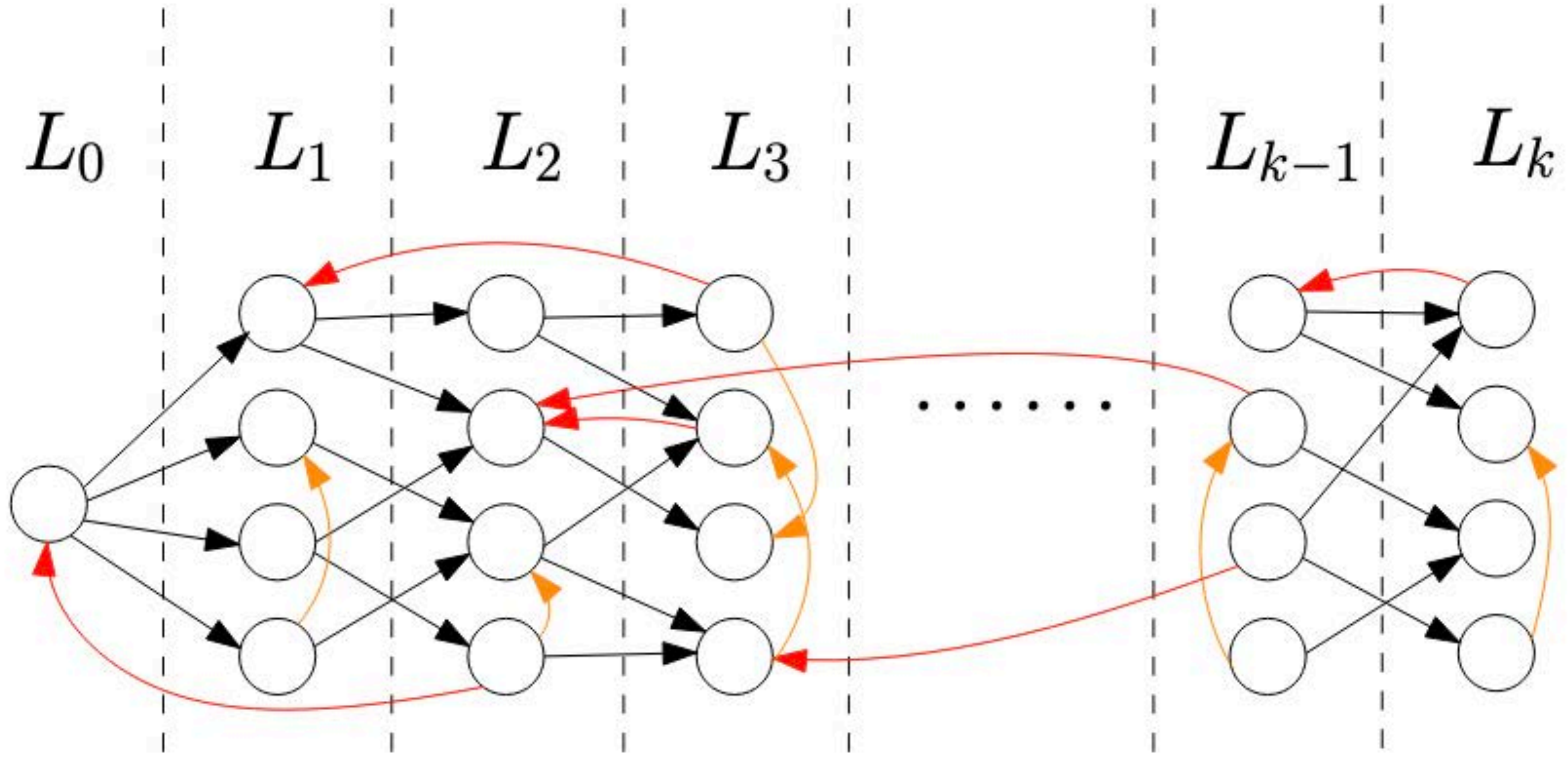
increase the flow f in P by $\delta(P)$

- Algorithm: keep finding shortest path, then augment
- Time cost?
  - Path finding: $O(m)$ time.
  - Path lengths: $1, \ldots, n$.
  - #paths for each length: $\leq m$
- $O(m^2 n)$ time in total

Augmenting Path

Shortest augmenting [Dinitz, Edmonds-Karp]

Observation:
augmenting shortest path
results in **longer** path

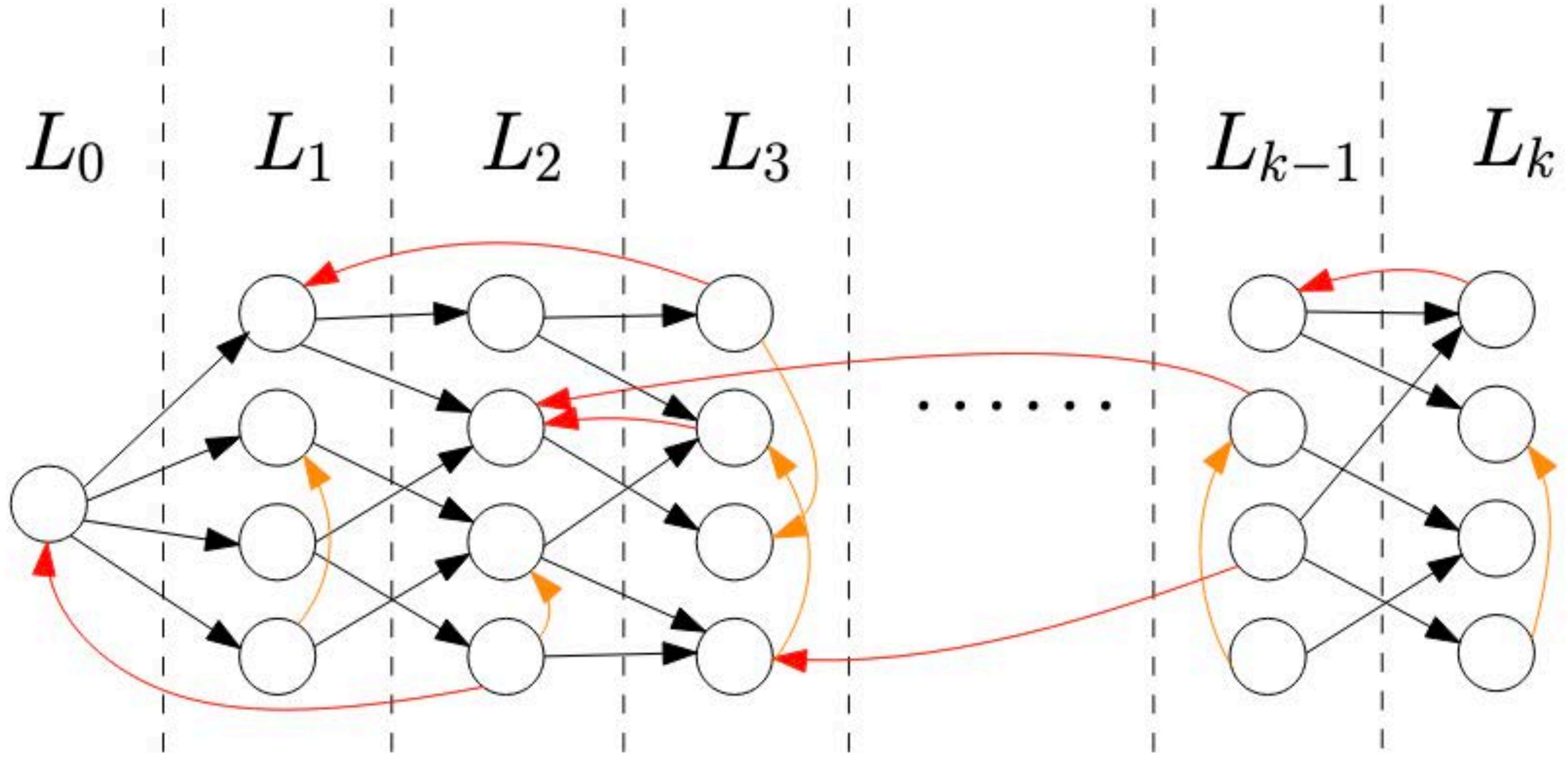


- Imagine the shortest s-t paths in G^f by layers
  - Forth edges: edges between adjacent layers
  - Side edges: edges between the same layer
  - Back edges: edges from further layer to nearer layer
  - Jumping edge: edge from nearer layer to non-adjacent further layer

Augmenting Path

Shortest augmenting [Dinitz, Edmonds-Karp]

Observation:
augmenting shortest path
results in **longer** path

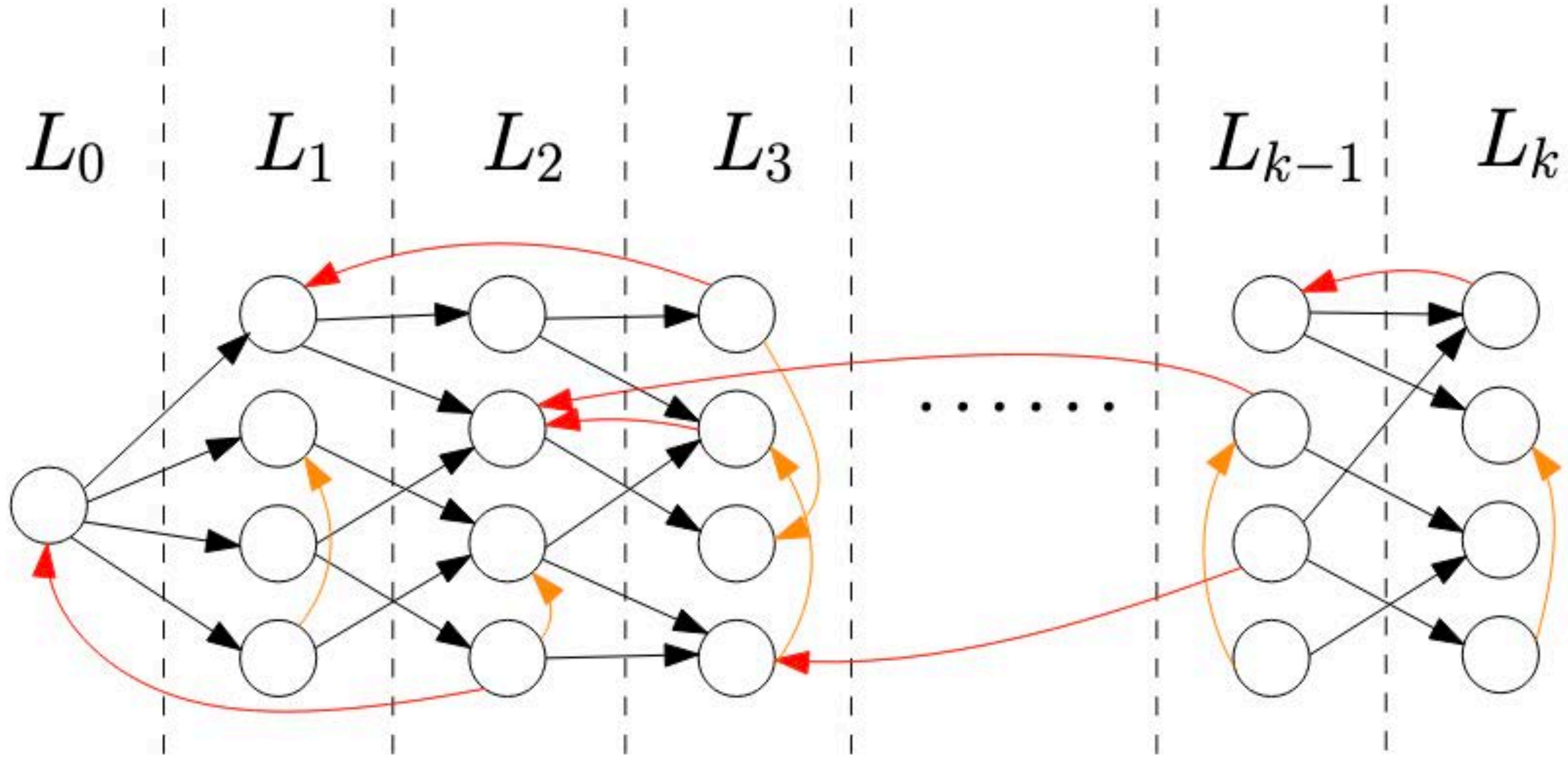


- Imagine the shortest s-t paths in G^f by layers
- Augmenting removes ≥ 1 forth edges, and adds ≥ 0 back edges
- No jumping edge is added by augmenting

Augmenting Path

Improving [Dinitz]

Observation:
augmenting shortest path
results in **longer** path

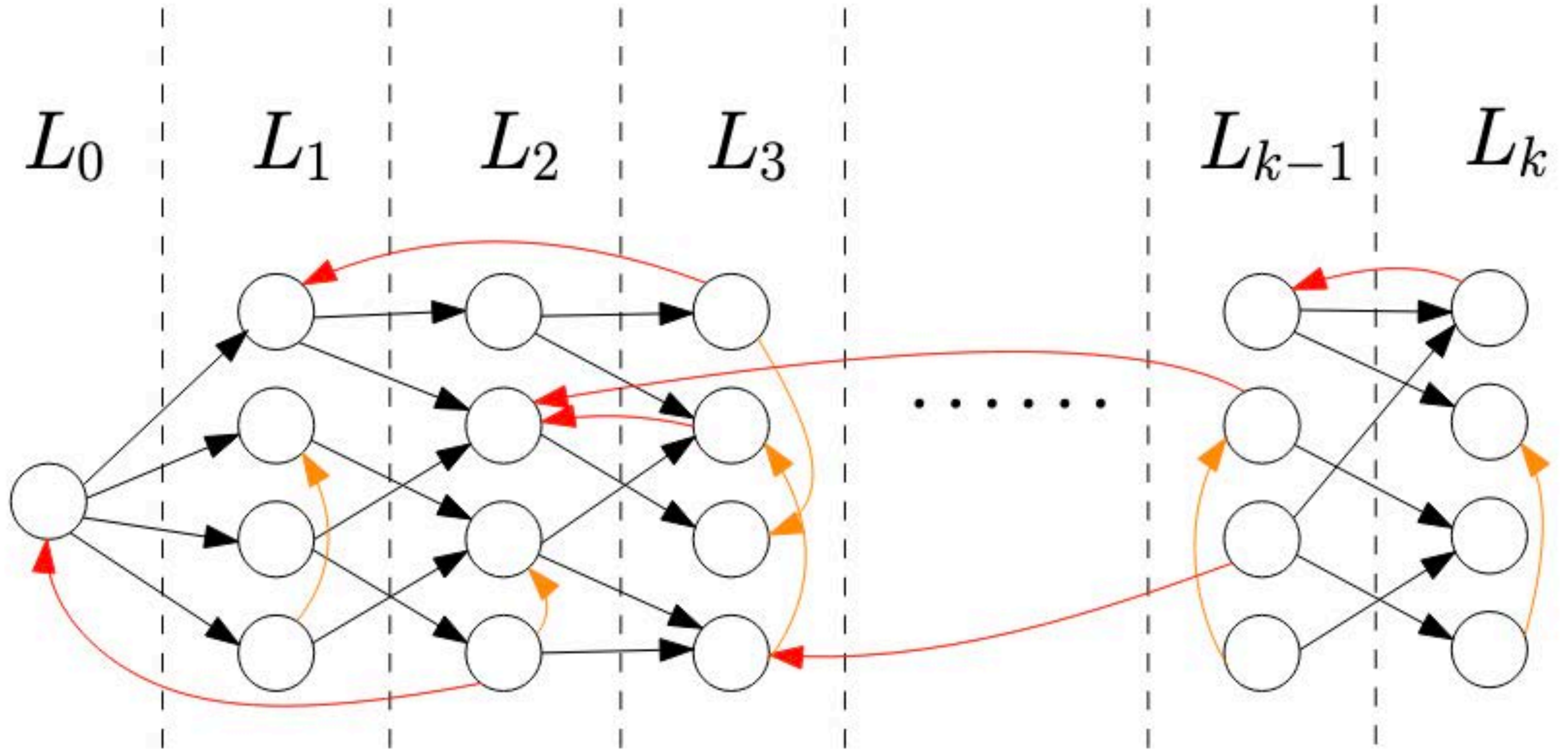


- Imagine the shortest s-t paths in G^f by layers
- Augmenting removes ≥ 1 forth edges, and adds ≥ 0 back edges
- No jumping edge is added by augmenting
- Augmenting all shortest paths at one round: DFS only on forth edges.

Augmenting Path

Improving [Dinitz]

Observation:
augmenting shortest path
results in **longer** path



- Augmenting all shortest paths at one round: DFS only on forth edges.
  - DFS: $O(nm)$ time.
  - Path lengths: $1, \ldots, n$.
- $O(n^2 m)$ time in total

Augmenting Path

Algorithms	time	type
Ford-Fulkerson	O(mf)	pseudo-polynomial
Scaling	O($m^2 \ln f$)	wealy-polynomial
Edmonds-Karp Dinitz	O($m^2 n$) O(mn^2)	strongly-polynomial

Polynomial time: in proportion to polynomial of n, where n is the **size of input**

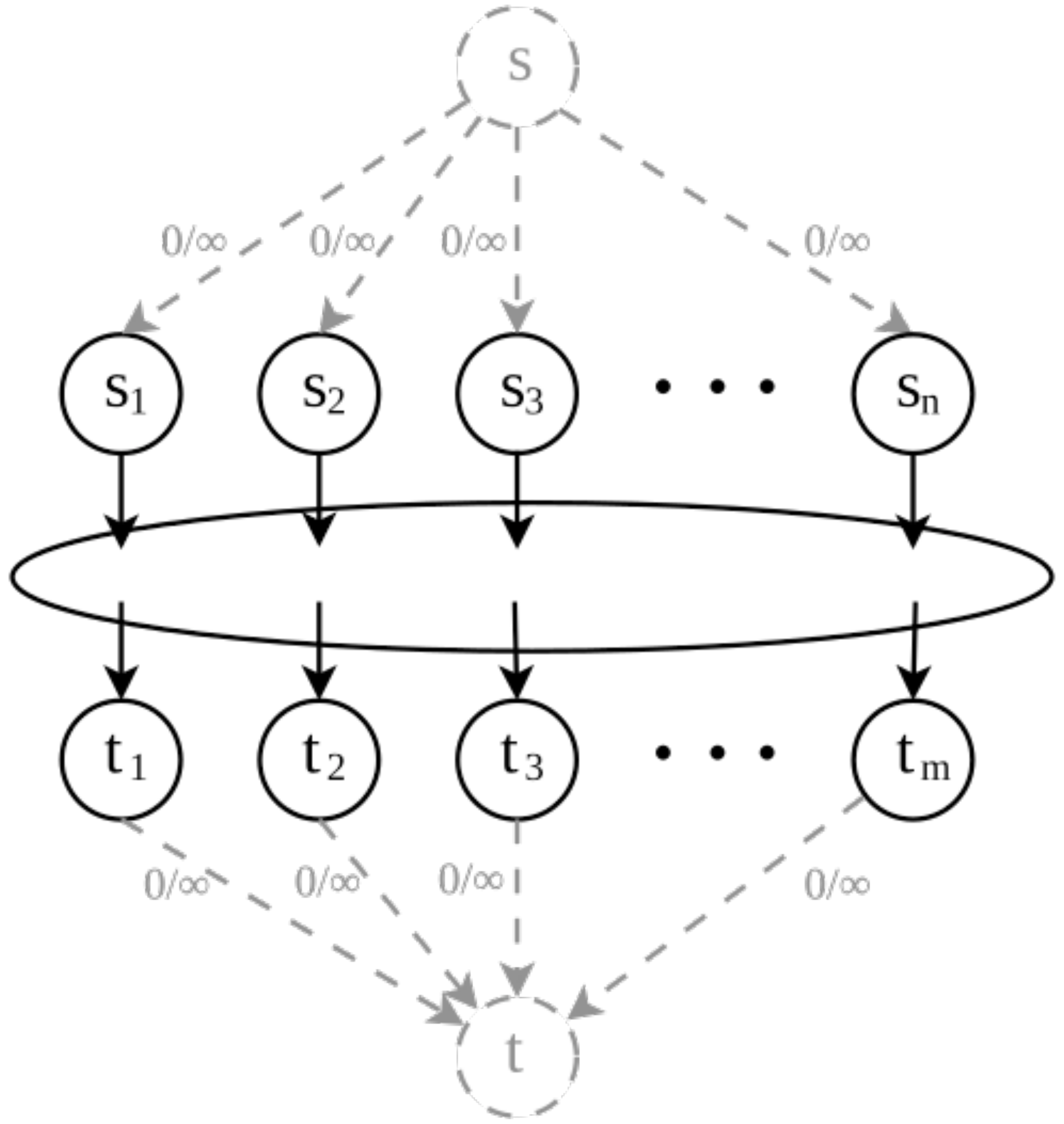
Maximum Flow Problem

Method	Year	Complexity
Linear programming		
Ford–Fulkerson algorithm	1955	$O(EU)$
Edmonds–Karp algorithm	1970	$O(VE^2)$
Dinic's algorithm	1970	$O(V^2E)$
MKM (Malhotra, Kumar, Maheshwari) algorithm[1]	1978	$O(V^3)$
Dinic's algorithm with dynamic trees	1983	$O(VE\log V)$
General push–relabel algorithm[2]	1986	$O(V^2E)$
Push–relabel algorithm with *FIFO* vertex selection rule[2]	1988	$O(V^3)$
Push–relabel algorithm with *maximum distance* vertex selection rule[3]	1988	$O(V^2\sqrt{E})$
Push-relabel algorithm with dynamic trees[2]	1988	$O\left(VE\log\frac{V^2}{E}\right)$
KRT (King, Rao, Tarjan)'s algorithm[4]	1994	$O\left(VE\log_{\frac{E}{V\log V}}V\right)$
Binary blocking flow algorithm[5]	1998	$O\left(E\cdot\min\{V^{2/3},E^{1/2}\}\cdot\log\frac{V^2}{E}\cdot\log U\right)$
James B Orlin's + KRT (King, Rao, Tarjan)'s algorithm[6]	2013	$O(VE)$
Kathuria-Liu-Sidford algorithm[7]	2020	$E^{4/3+o(1)}U^{1/3}$
BLNPSSSW / BLLSSSW algorithm[8][9]	2020	$\tilde{O}((E+V^{3/2})\log U)$
Gao-Liu-Peng algorithm[10]	2021	$\tilde{O}(E^{\frac{3}{2}-\frac{1}{328}}\log U)$
Chen, Kyng, Liu, Peng, Gutenberg and Sachdeva's algorithm[11]	2022	$O(E^{1+o(1)}\log U)$
Bernstein, Blikstad, Saranurak, Tu[12]	2024	$O(n^{2+o(1)}\log U)$

Multi-Source Multi-Sink

Input: A network $G = (V, E)$ and capacity $c : E \to \mathbb{R}^+$, with sources $S = \{s_1, \ldots, s_n\}$ and sinks $T = \{t_1, \ldots, t_m\}$.

Output: A maximum flow from S to T across G.



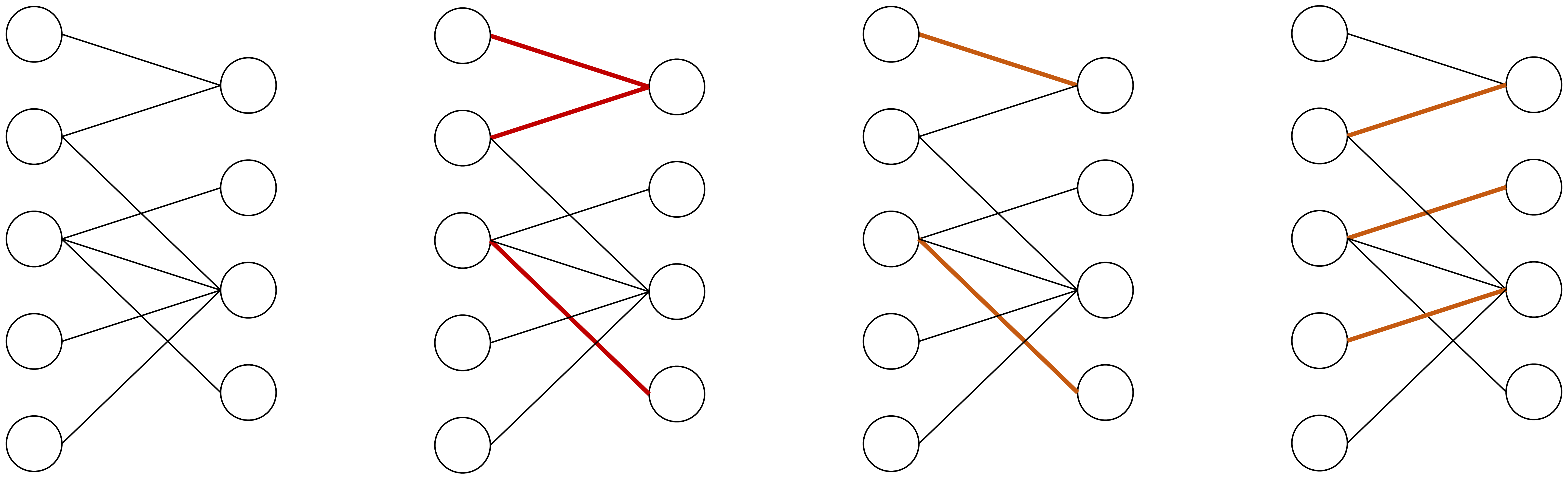
Maximum Bipartite Matching

- A graph $G = (V, E)$ is **bipartite** if V can be partitioned into L and R, such that each edge connects a vertex in L and a vertex in R.

- A **matching** of a graph $G = (V, E)$ is a subset $M \subseteq E$, such that for each $v \in V$, at most one edge in M is incident on v.

- A **maximum matching** is a matching of maximum cardinality.

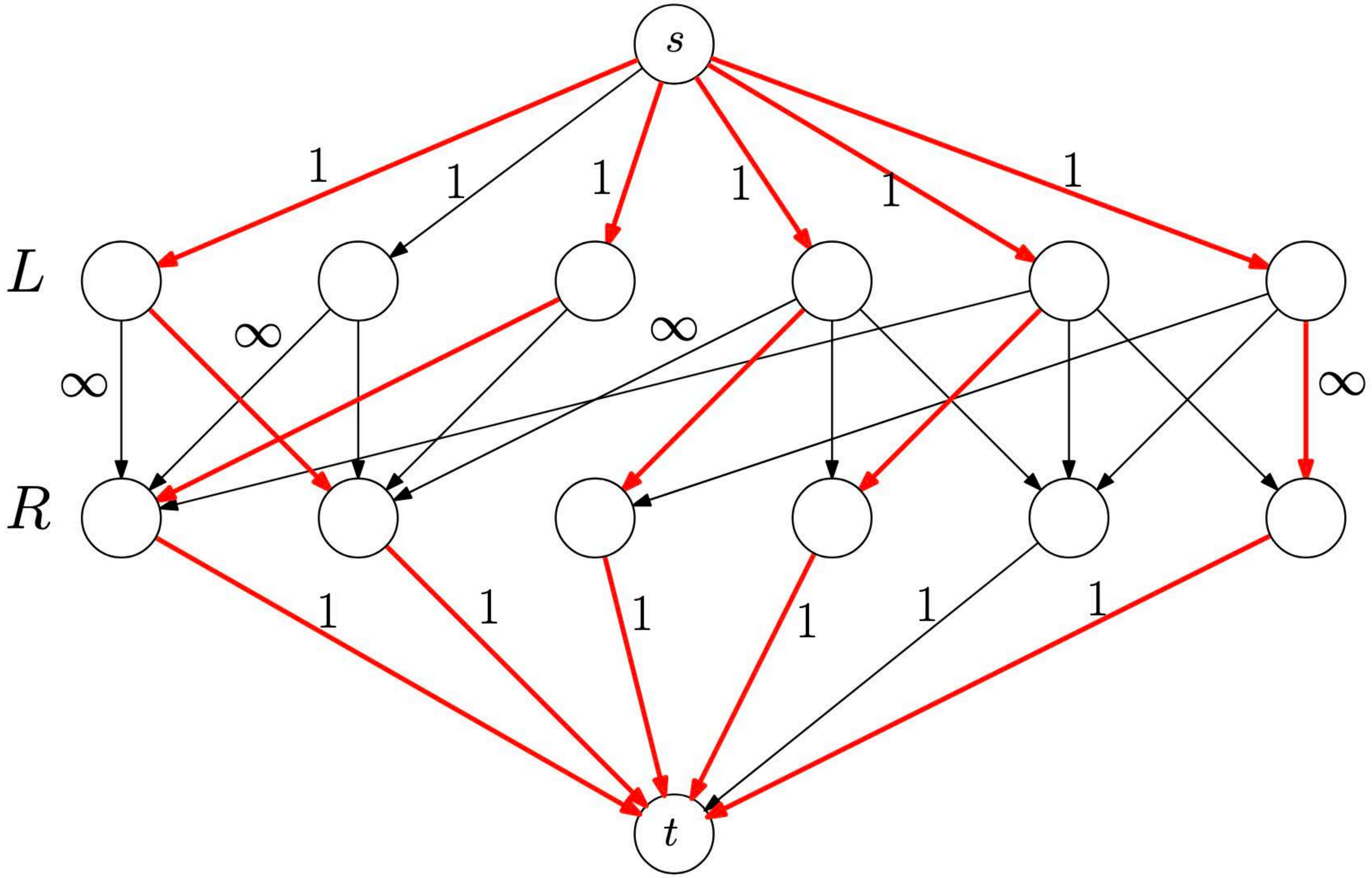
- **Problem:** Find a maximum matching of a bipartite graph.



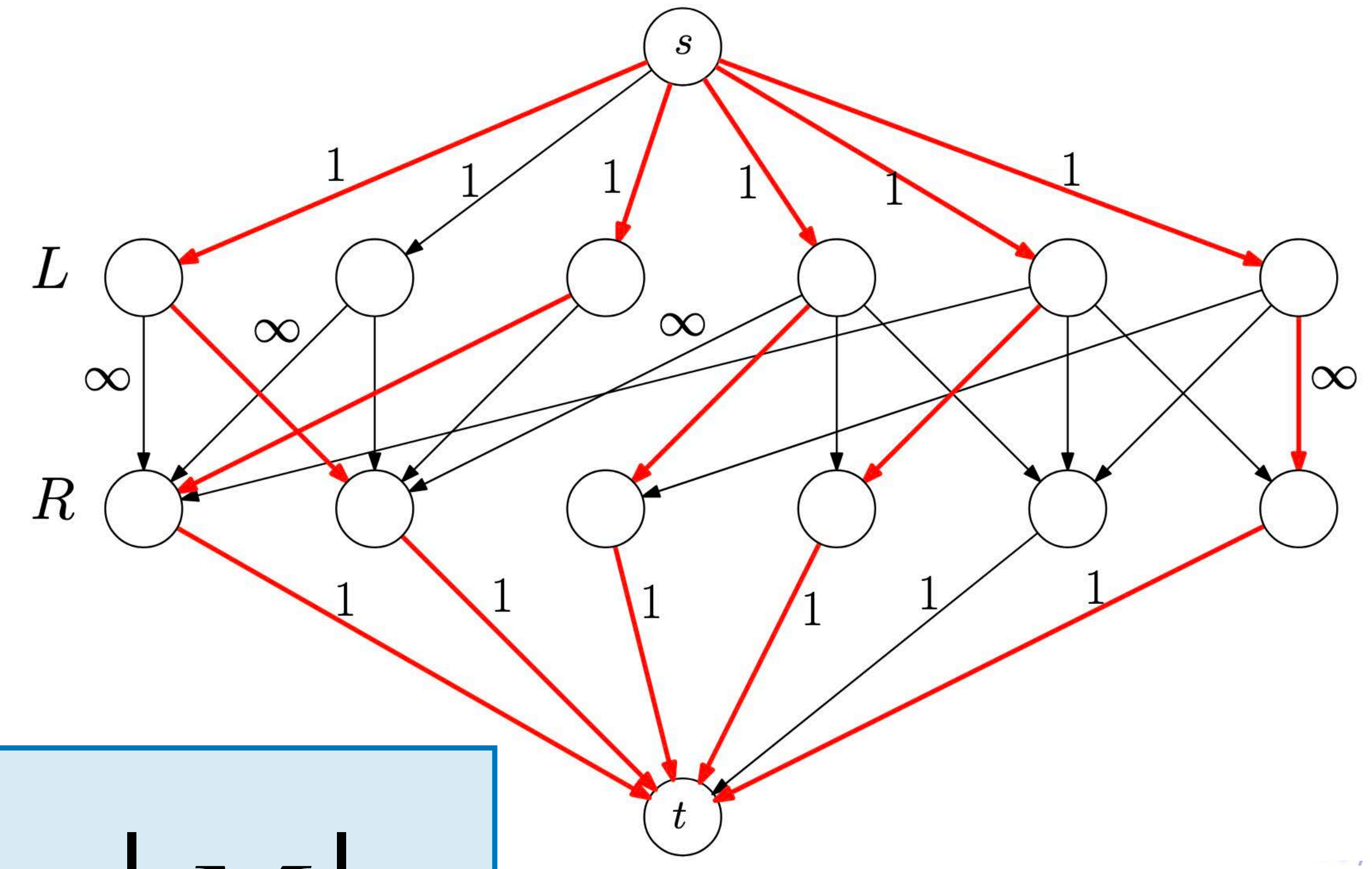
Maximum Bipartite Matching

Input: A bipartite graph $G = (L \cup R, E)$.

Output: A maximum matching of a bipartite graph.



Maximum Bipartite Matching



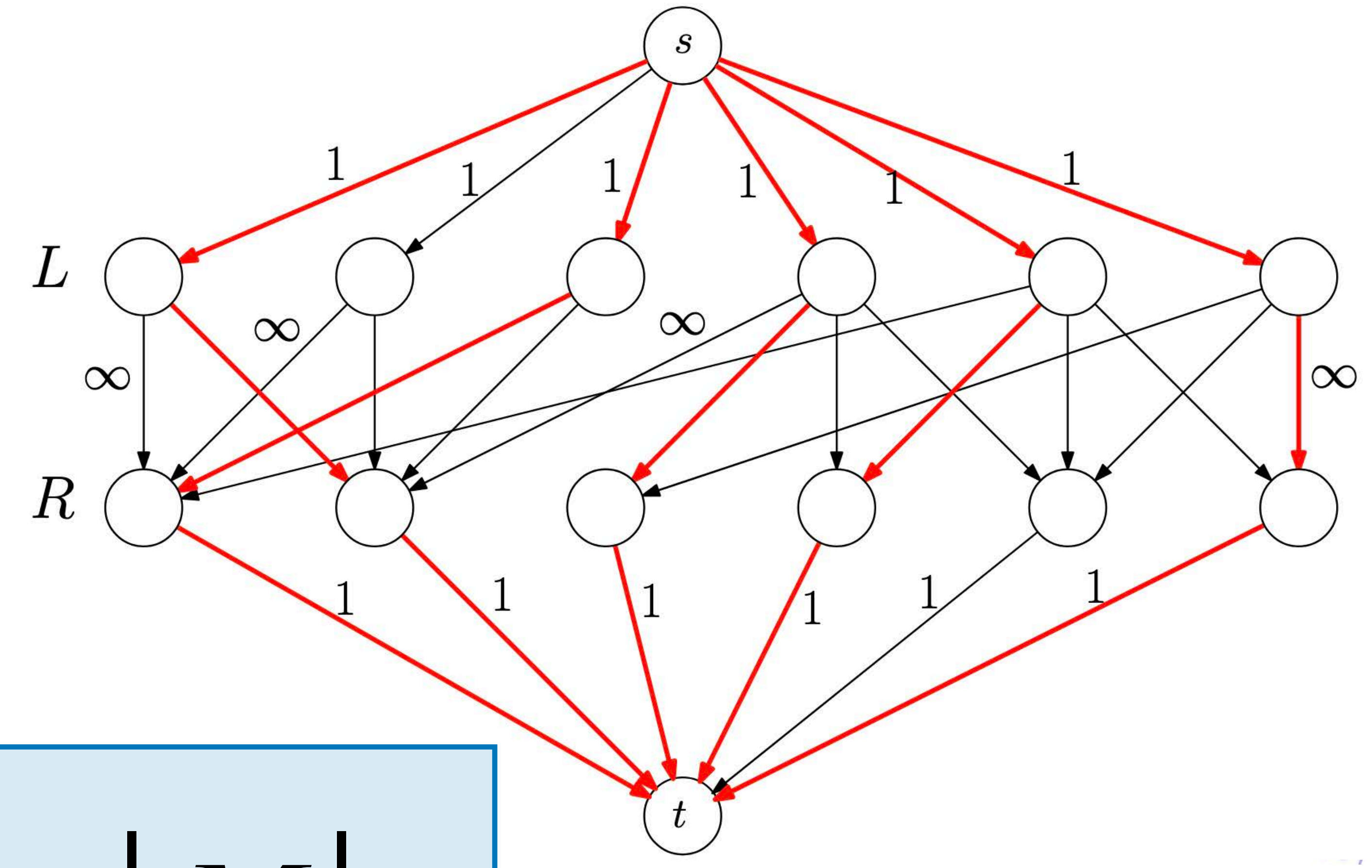
(1) $\exists$ matching M of $G \Rightarrow \exists$ flow f of G' with $|f| = |M|$

(2) $\exists$ flow f of $G' \Rightarrow \exists$ matching M of G with $|M| = |f|$.

(1):

- For each $e = (u, v) \in M$, let $f_{su} = f_e = f_{vt} = 1$
- For each $e = (u, v) \notin M$, let $f_e = 0$
- f is valid, and $|f| = |M|$

Maximum Bipartite Matching



(1) $\exists$ matching M of $G \Rightarrow \exists$ flow f of G' with $\left|f\right| = \left|M\right|$

(2) $\exists$ flow f of $G' \Rightarrow \exists$ matching M of G with $\left|M\right| = \left|f\right|$.

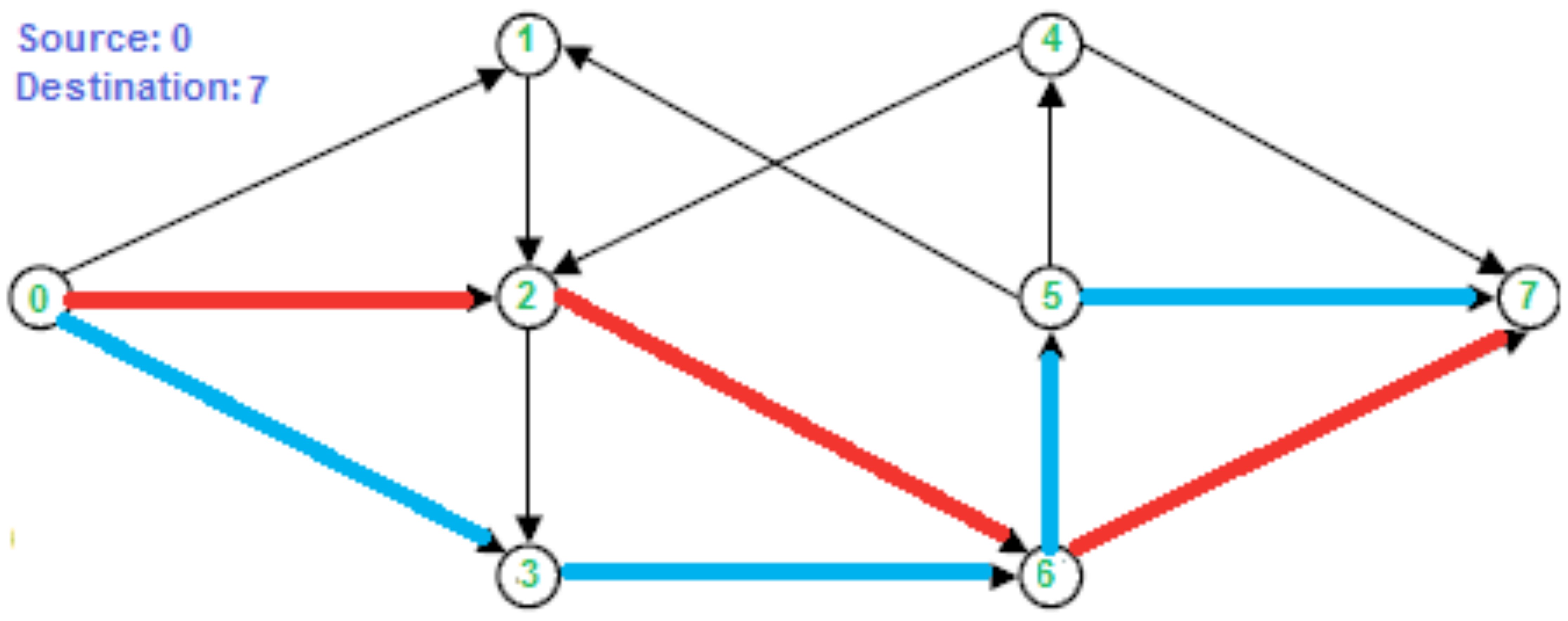
(2): Let $M \triangleq \{(u, v) \in E : u \in L, v \in R, f_e > 0\}$

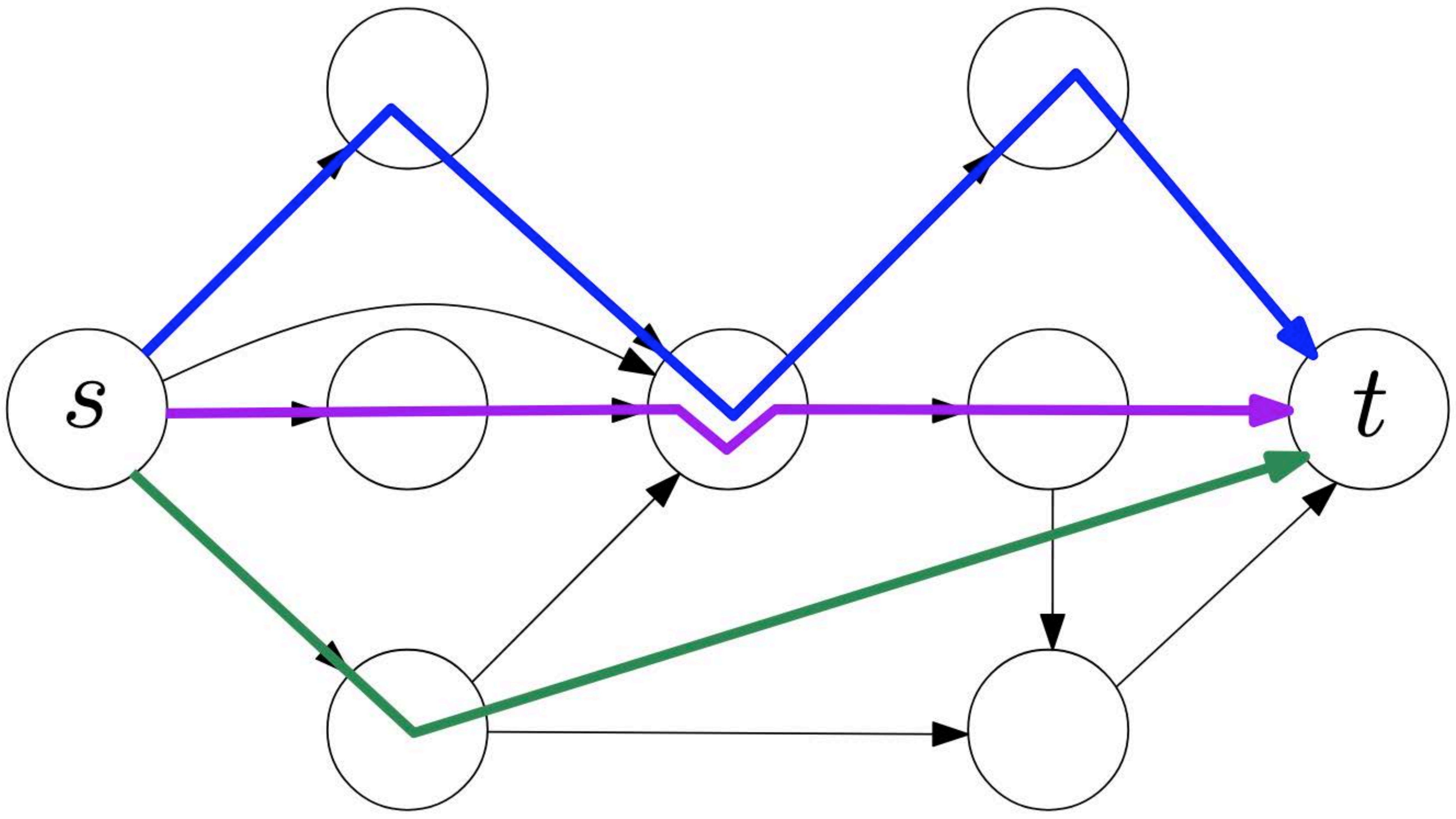
- M is a matching:
  - for any $u \in L$, $\sum_{v \in \delta(u) \setminus s} f_{uv} \leq 1$; for any $v \in R$, $\sum_{u \in \delta(v) \setminus t} f_{uv} \leq 1$.
  - Any $u \in L$ matches with at most one $v \in R$, so does any $v \in R$.
- $|M| = |f|$

#Disjont Paths

Input: A digraph $G = (V, E)$ and source & sink $s, t \in V$.

Output: The maximum #**edge-disjoint** paths from s to t





s-t Min-Cut

Input: A digraph $G = (V, E)$ and weights $c : E \to \mathbb{R}^+$, and source & sink $s, t \in V$.

Output: The minimum cut whose removal disconnects s, t.

The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

(3) There is $S, \neg S \subseteq V$ such that cut$(S) = |f|$

Global Min-Cut

Input: A digraph $G = (V, E)$ and weights $c : E \to \mathbb{R}^+$.

Output: The minimum cut whose removal disconnects G

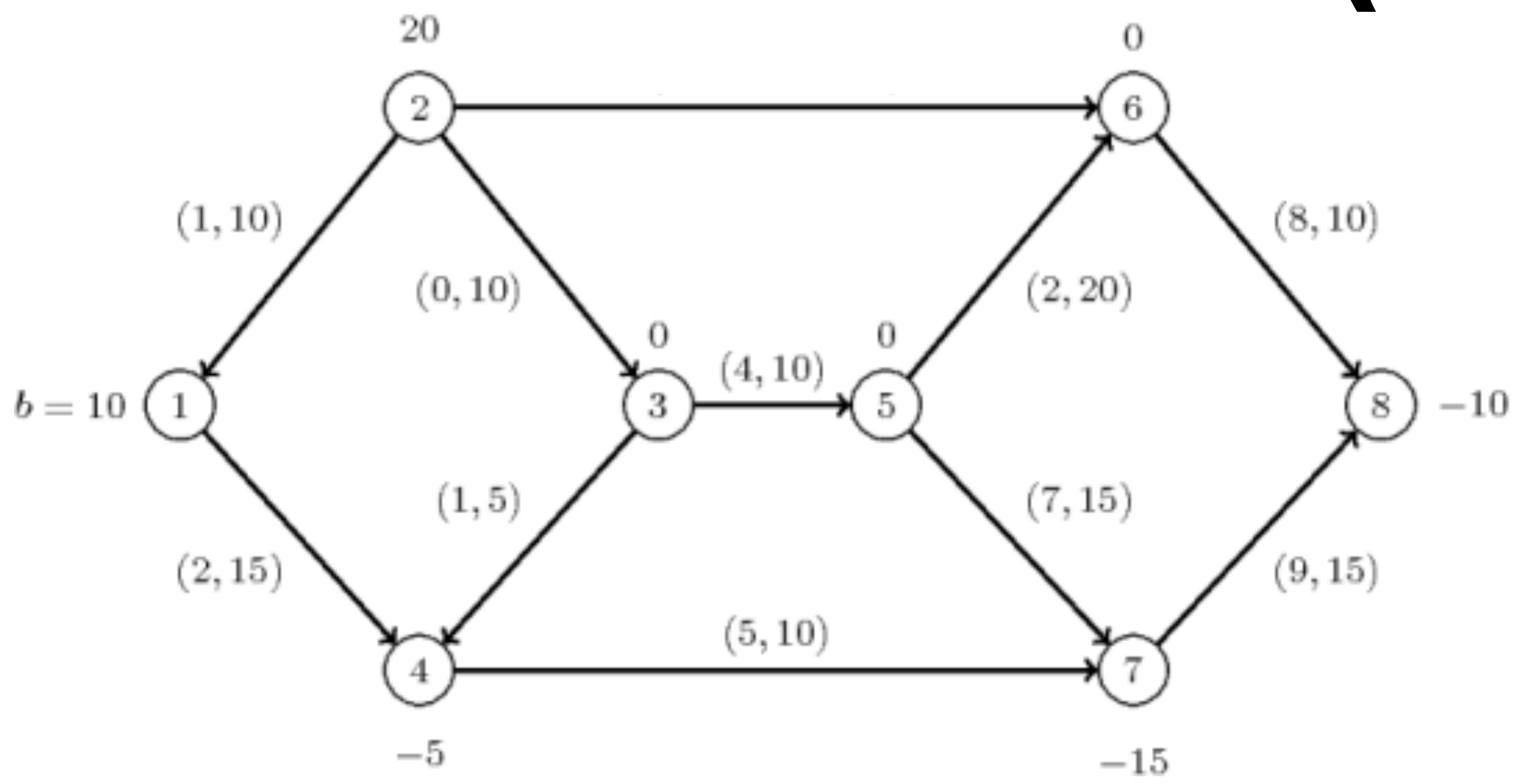
The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

(3) There is $S, \neg S \subseteq V$ such that $\text{cut}(S) = |f|$

Minimum-Cost Flow Problem (MCFP)



- Edge capacity $c_e \in \mathbb{R}^+$. **Flow cost** $a_e \in \mathbb{R}^+$.
- Flow function $f : E \to \mathbb{R}^+$
  - Valid flow: $\forall e \in E, 0 \leq f(e) \leq c_e$ & $\forall v \in V$, $\sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e)$
- MCFP: find a flow f minimizes $\sum_{e \in \delta_{out}(s)} a_e \cdot f(e)$ while maximizing $\sum_{e \in \delta_{out}(s)} f(e)$

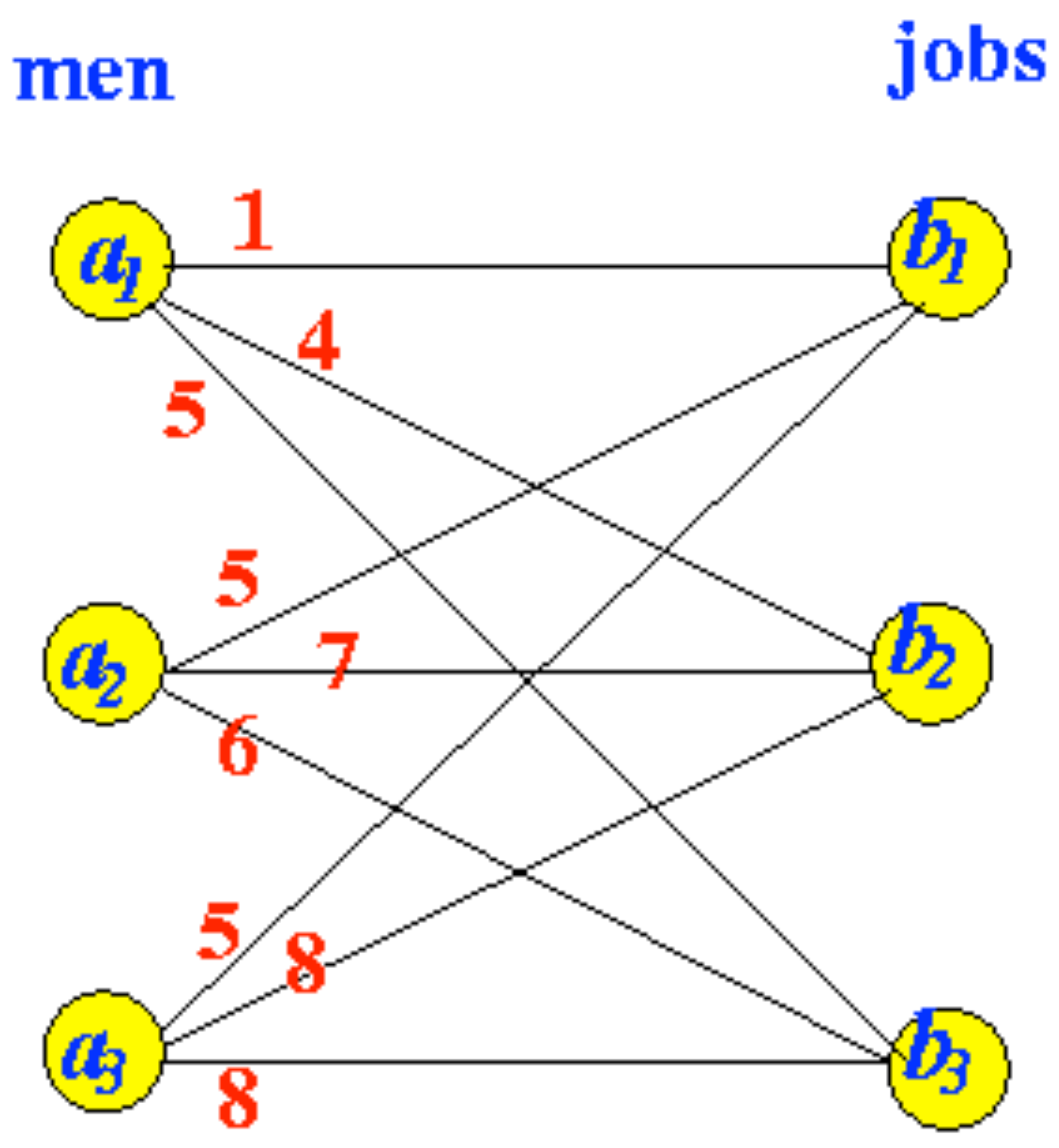
Assignment Problem

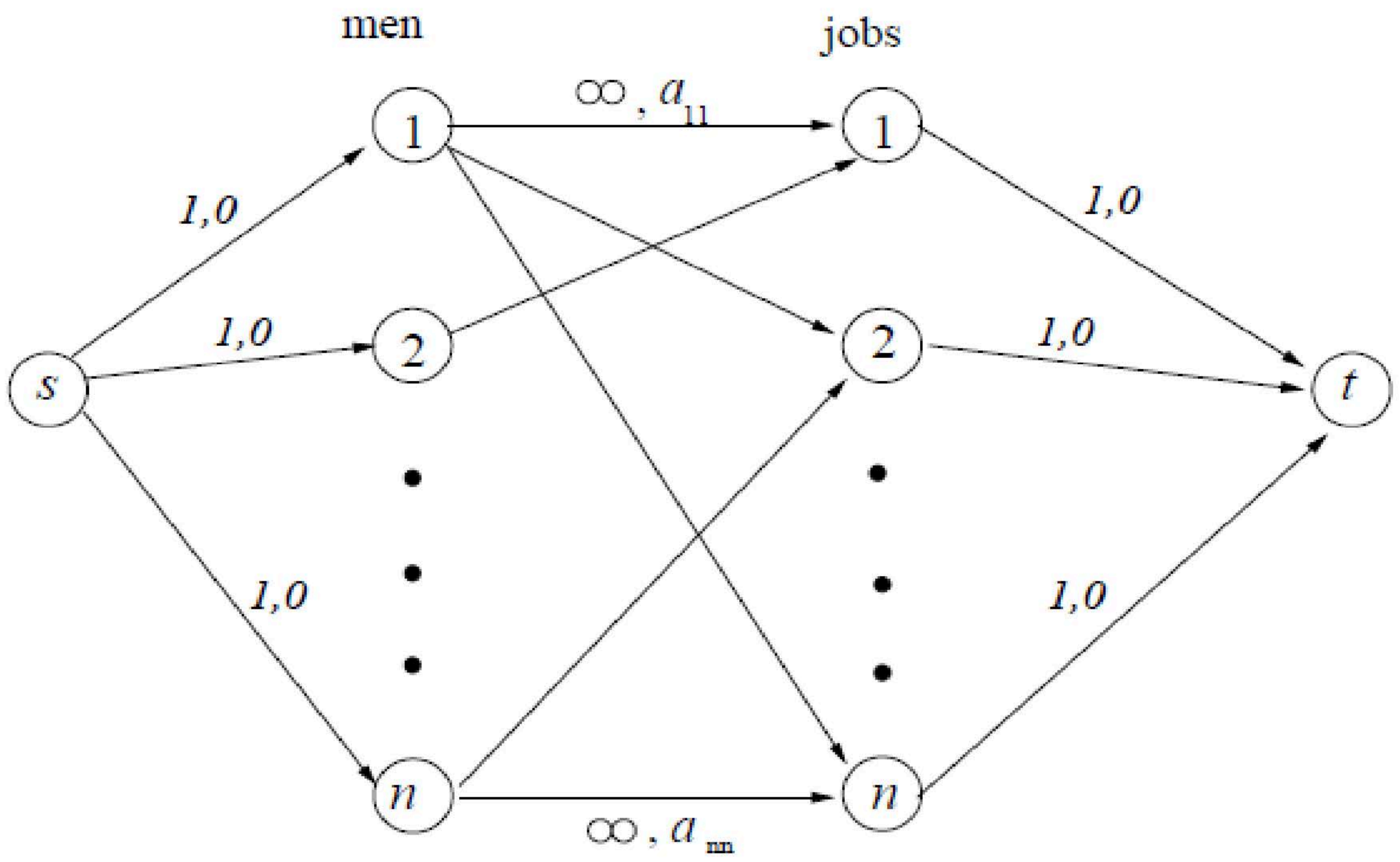
Minimum-cost perfect matching

Given matrix of costs

Worker	Task I	Task II	Task III
Ali	8	4	7
Baba	5	2	3
Curi	9	6	7
Durian	9	4	8

Make square with dummy column.
Subtract minimum for each column:





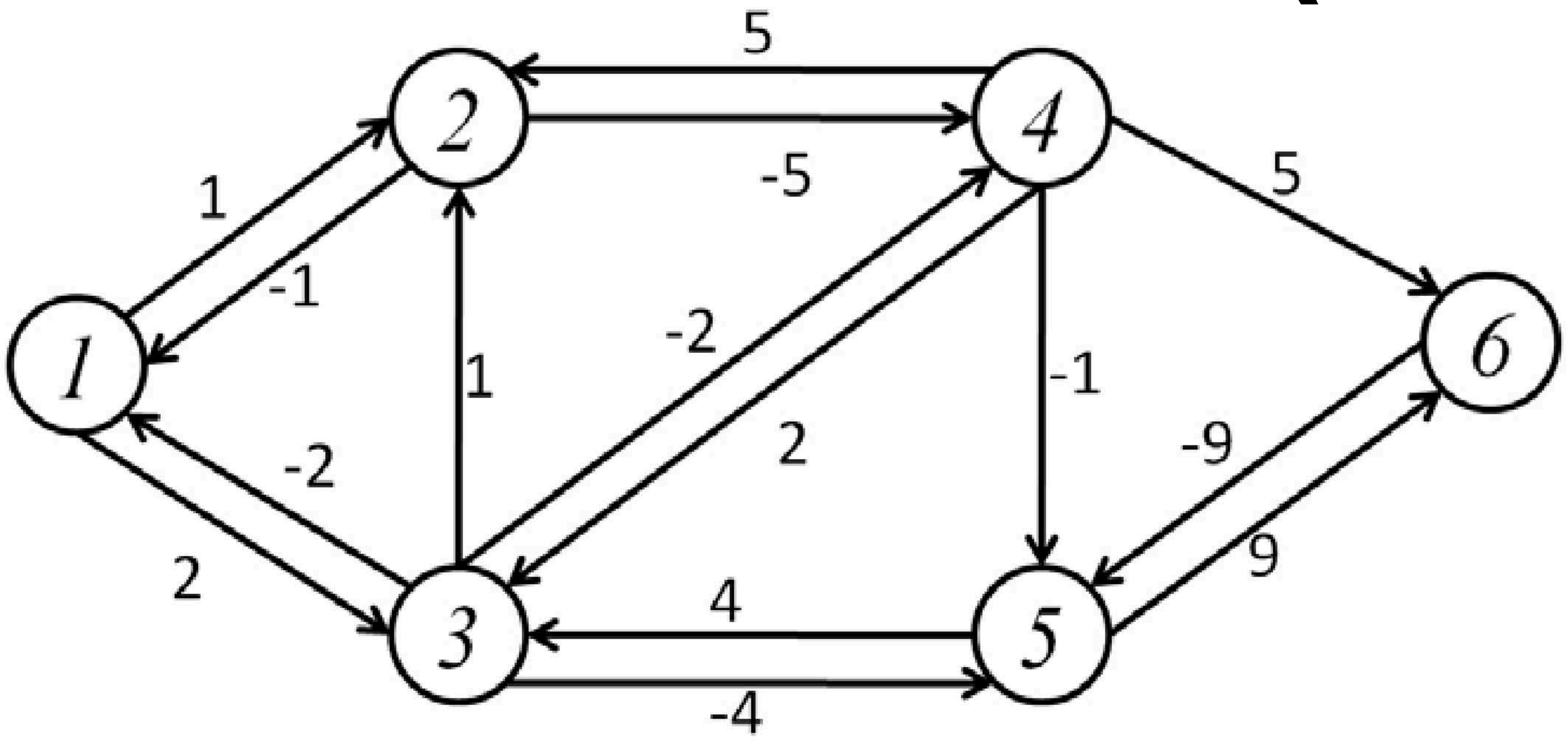
Minimum-Cost Flow Problem (MCFP)

Input: A digraph $G = (V, E)$ and capacity & costs $c, a : E \to \mathbb{R}^+$,
and source s, sink t.

Output: A flow f minimizes $\sum_{e \in \delta_{out}(s)} a_e \cdot f(e)$ while maximizing $\sum_{e \in \delta_{out}(s)} f(e)$

- An optimal solution be like: a maximum flow which cannot be improved
- Am improvement be like: redirecting the flow with lower cost
- A redirection be like:
  - pushing backward a flow from t to s, pushing forward a flow from s to t

Minimum-Cost Flow Problem (MCFP)



- An optimal solution be like: a maximum flow which cannot be improved
- Am improvement be like: redirecting the flow with lower cost
- A redirection be like:
  - pushing backward a flow from t to s, pushing forward a flow from s to t
  - a cycle in the residual graph
- An improving redirection be like:
  - a negative cycle in the residual graph

Minimum-Cost Flow Problem (MCFP)

Input: A digraph $G = (V, E)$ and capacity & costs $c, a : E \to \mathbb{R}^+$,
and source s, sink t.

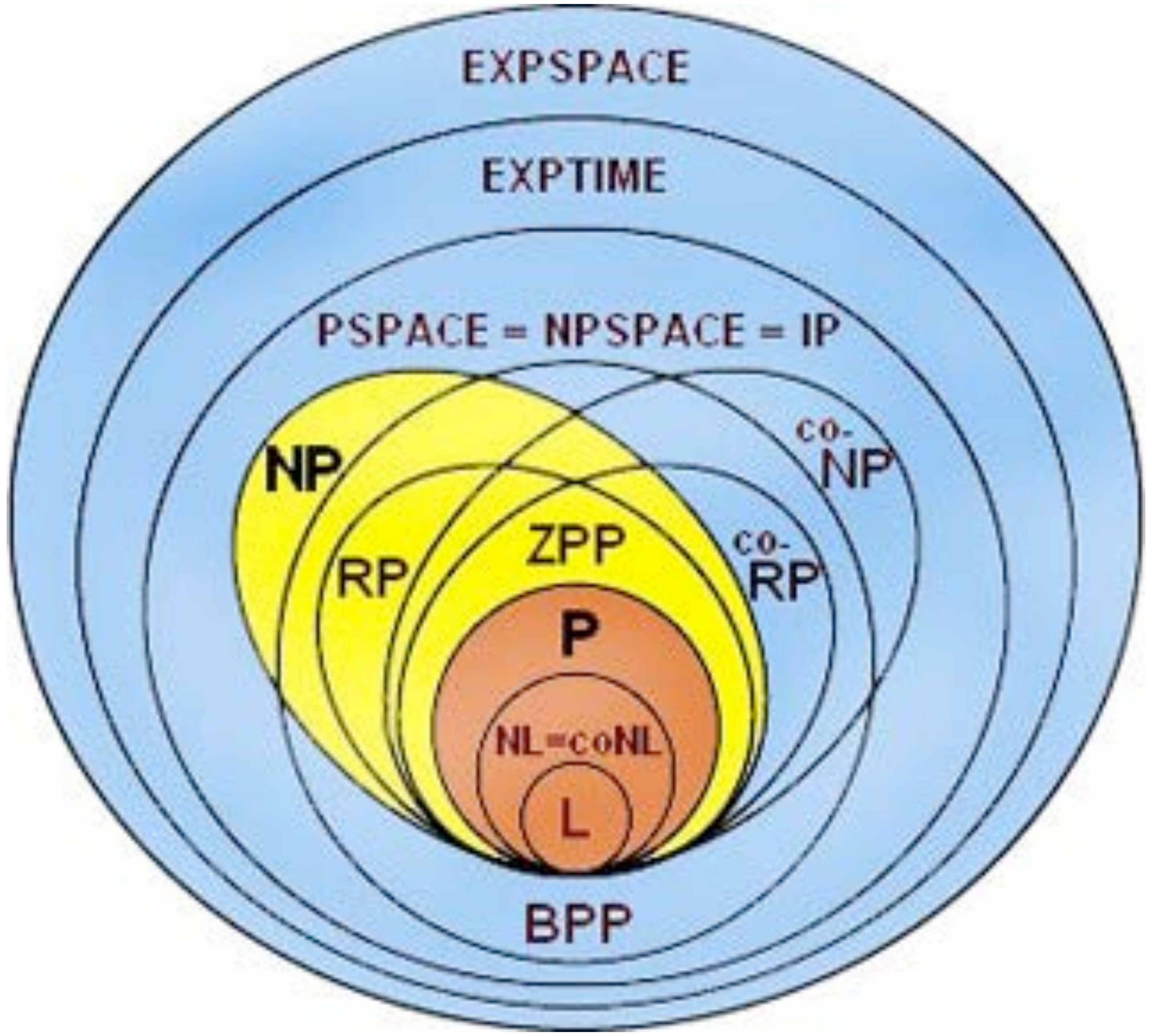
Output: A flow f minimizes $\sum_{e \in \delta_{out}(s)} a_e \cdot f(e)$ while maximizing $\sum_{e \in \delta_{out}(s)} f(e)$

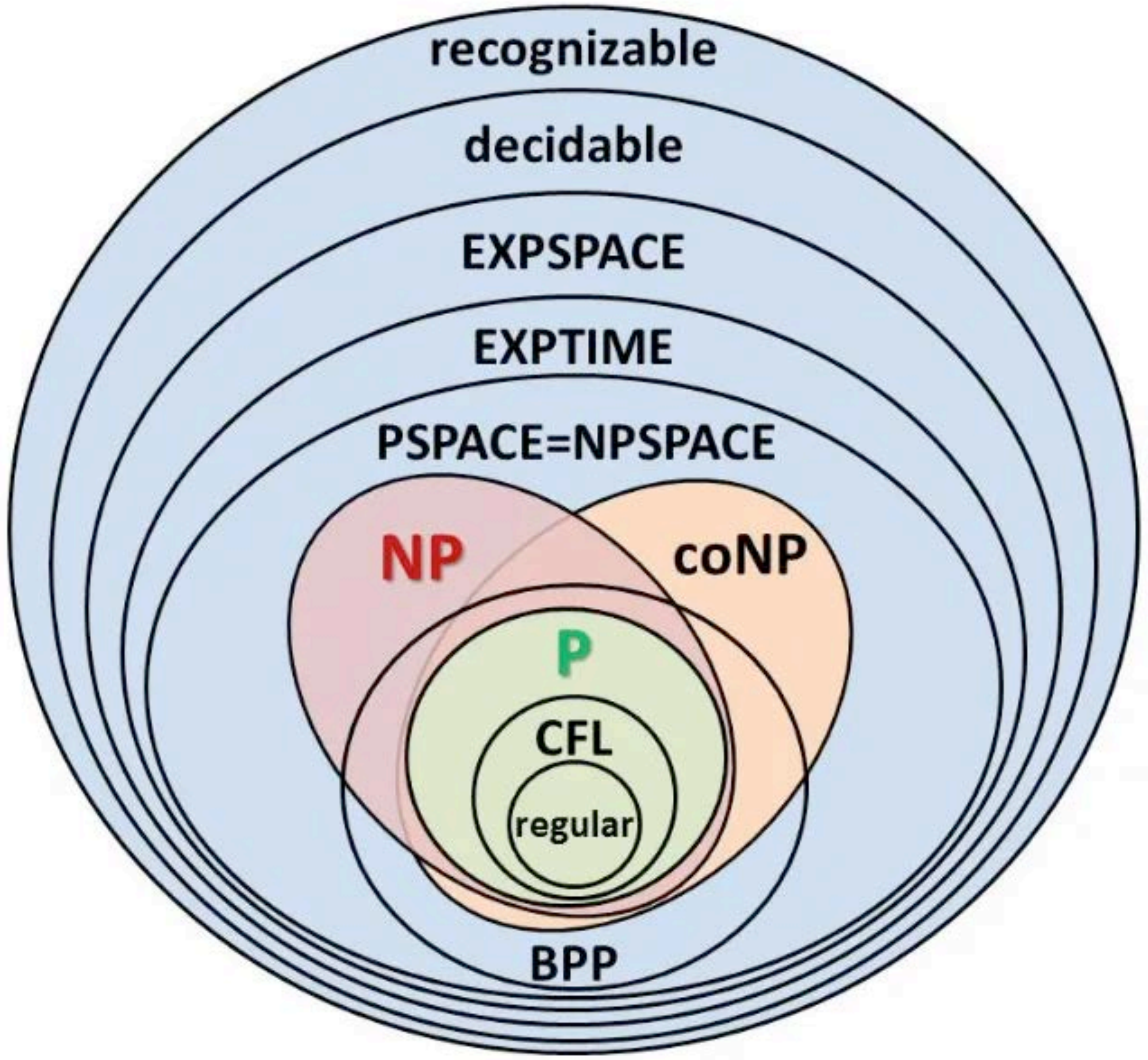
- An improving redirection be like:
  - a negative cycle in the residual graph

Cycle Canceling:

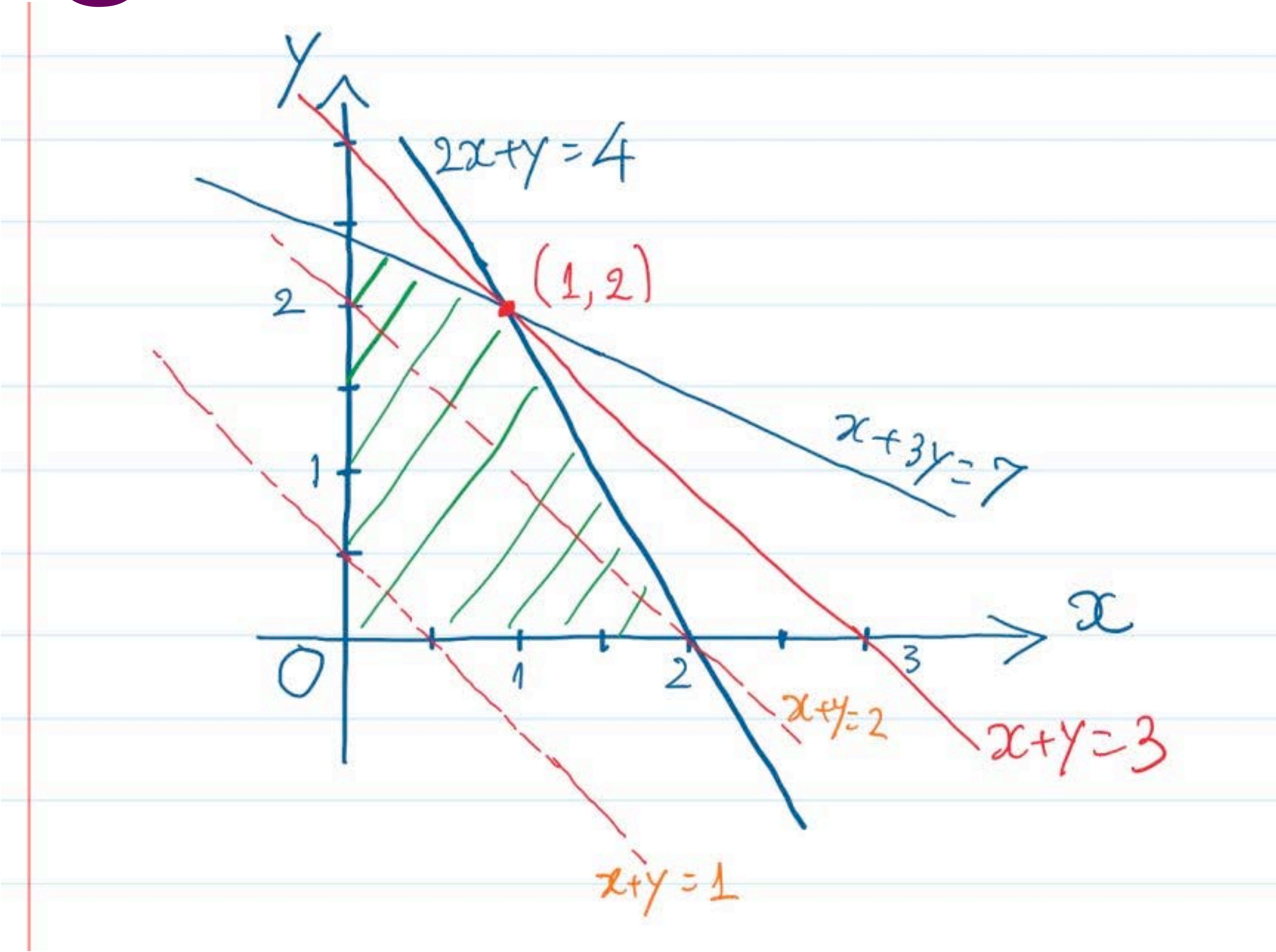
While exists negative cycle in residual graph
cancel the negative cycle by redirecting

Maximum Flow Is P-Complete
Under log space reduction





Linear Programming



A Familiar Problem

某厂生产甲乙两种产品，每生产 1 吨产品的电耗、煤耗、所需劳动力及产值如表 3 所示：

表 3

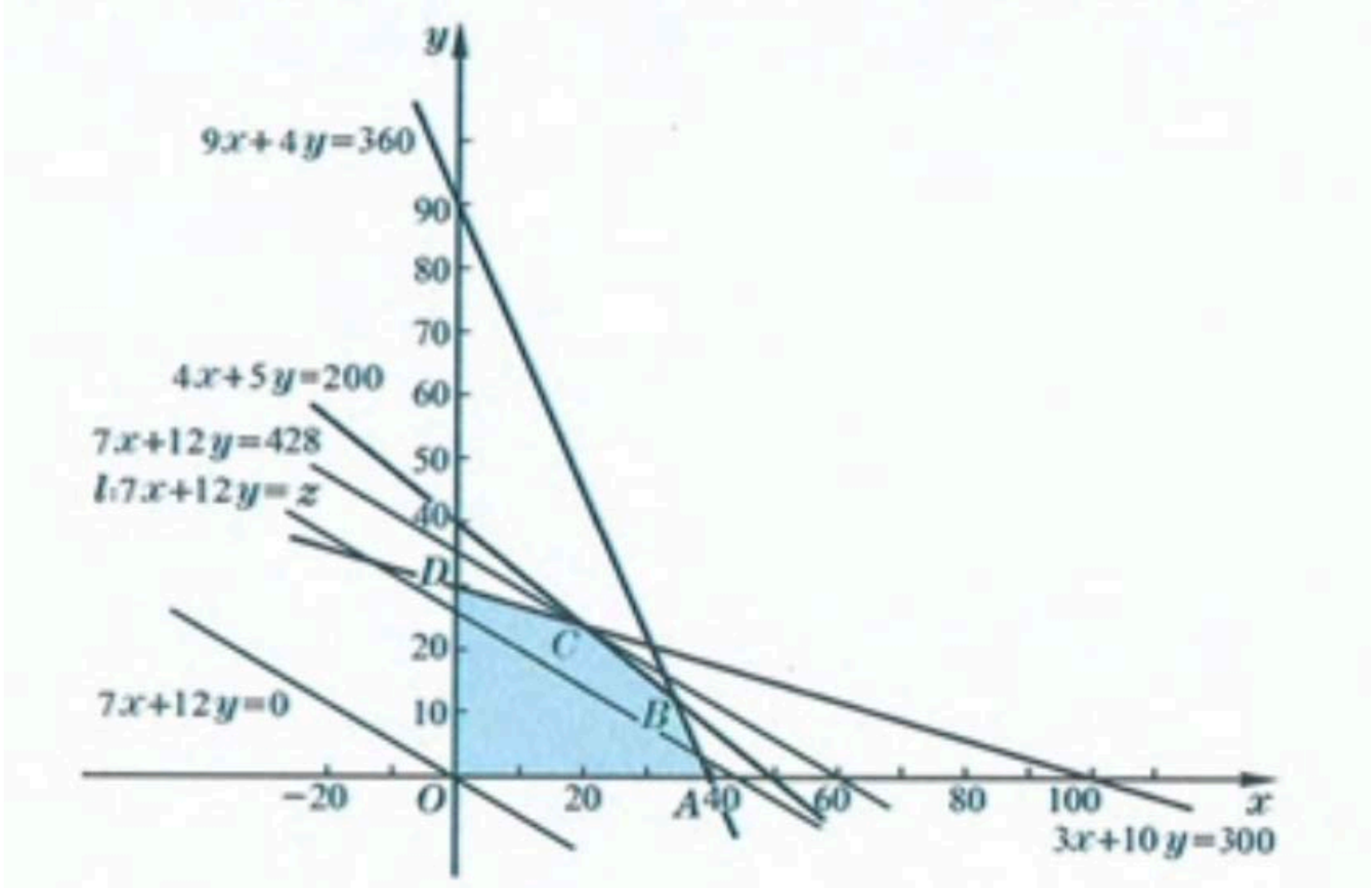
产品	电耗（千瓦时）	煤耗（吨）	劳动力（人）	产值（万元）
甲	4	9	3	7
乙	5	4	10	12

已知该厂有劳动力 300 人，按计划煤耗每天不超过 360 吨，电耗每天不超过 200 千瓦时．每天应如何安排生产，可使产值最大？

如果设该厂每天生产甲产品 x 吨，乙产品 y 吨，那么上述问题可转化为在满足以下线性约束条件：

$$\text{(B)}\quad \begin{cases} 9x+4y\leqslant 360, \\ 4x+5y\leqslant 200, \\ 3x+10y\leqslant 300, \\ x\geqslant 0, \\ y\geqslant 0, \end{cases}$$

求线性目标函数 $z=7x+12y$ 的最大值．

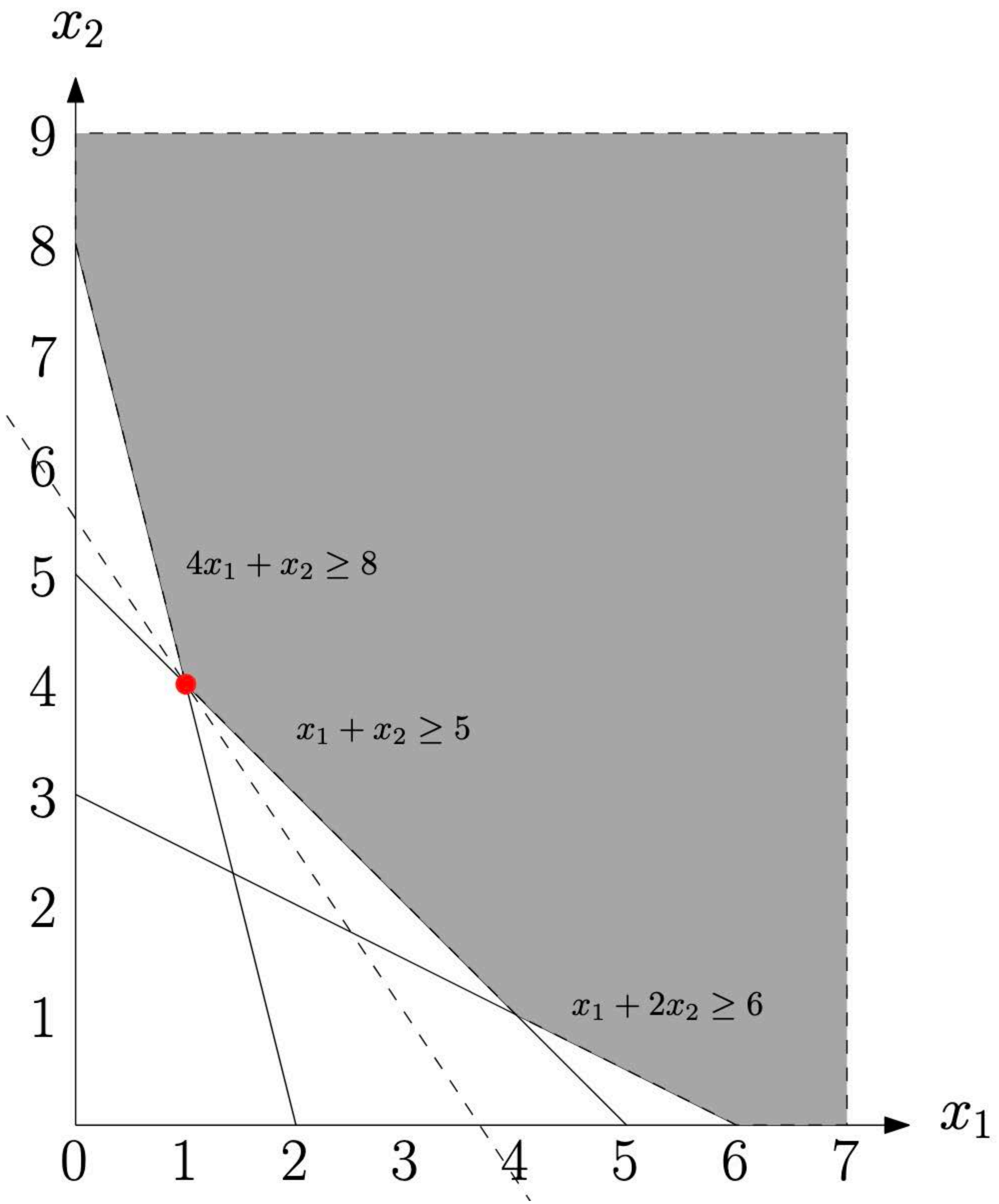


minimize $\quad 7x_1 + 4x_2$

subject to
$$x_1 + x_2 \geq 5$$
$$x_1 + 2x_2 \geq 6$$
$$4x_1 + x_2 \geq 8$$
$$x_1, x_2 \geq 0$$

optimal point $x_1 = 1, x_2 = 4$

value $= 7 \times 1 + 4 \times 4 = 23$



Linear Programming (LP)

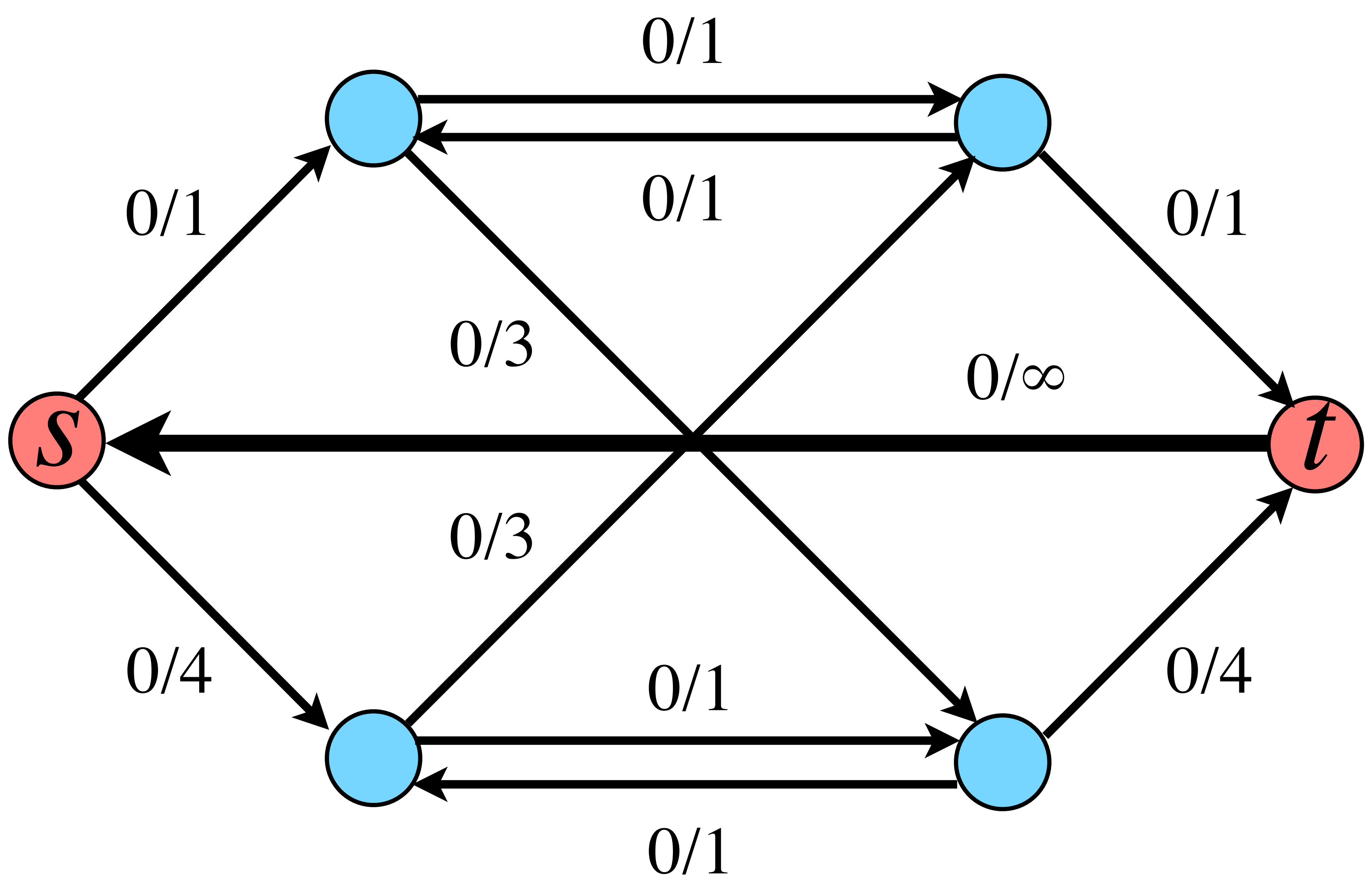
- General form:
  - matrix $A = \{a_{ij}\}_{[m]\times[n]}$, sets $M \subseteq [m]$ and $N \subseteq [n]$

$$
\begin{array}{lll}
\text{minimize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & \\
\text{subject to} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i & i \in M \\
 & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i & i \in \overline{M} \\
 & x_j \geq 0 & j \in N \\
 & x_j \text{ unconstrained} & j \in \overline{N}
\end{array}
$$

Max-Flow

digraph: $G = (V, E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



max $\sum_{u:(s,u)\in E} f_{su}$ $\quad f_{ts}$

s.t. $0 \le f_{uv} \le c_{uv}$ $\quad \forall (u,v) \in E$

$$\sum_{w:(w,u)\in E} f_{wu} - \sum_{v:(u,v)\in E} f_{uv} \geq 0 \quad \forall u \in V \setminus \{s,t\}$$

Linear Programming (LP)

General form:

$$\begin{array}{lll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & \\ \text{s.t.} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i & i \in M \\ & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i & i \in \overline{M} \\ & x_j \geq 0 & j \in N \\ & x_j \text{ unconstrained} & j \in \overline{N} \end{array}$$

$\Longrightarrow$

Canonical form:

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i \quad \Longrightarrow \quad \begin{cases} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i \\ -\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq -b_i \end{cases}$$

$$x_j \text{ unconstrained} \quad \Longrightarrow \quad x_j = x_j^+ - x_j^- \text{ where} \begin{cases} x_j^+ \geq 0 \\ x_j^- \geq 0 \end{cases}$$

Solvable in Polynomial Time

Canonical Form of Linear programming

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Algorithm	Theory	Practice
Simplex Method	Exponential Time	Works Well
Ellipsoid Method	Polynomial Time	Slow
Internal Point Methods	Polynomial Time	Works Well

Convex Polytopes

- **hyperplane**:

  subspace of dimension $n - 1$

  $$\sum_{j=1}^{n} a_j x_j = b$$

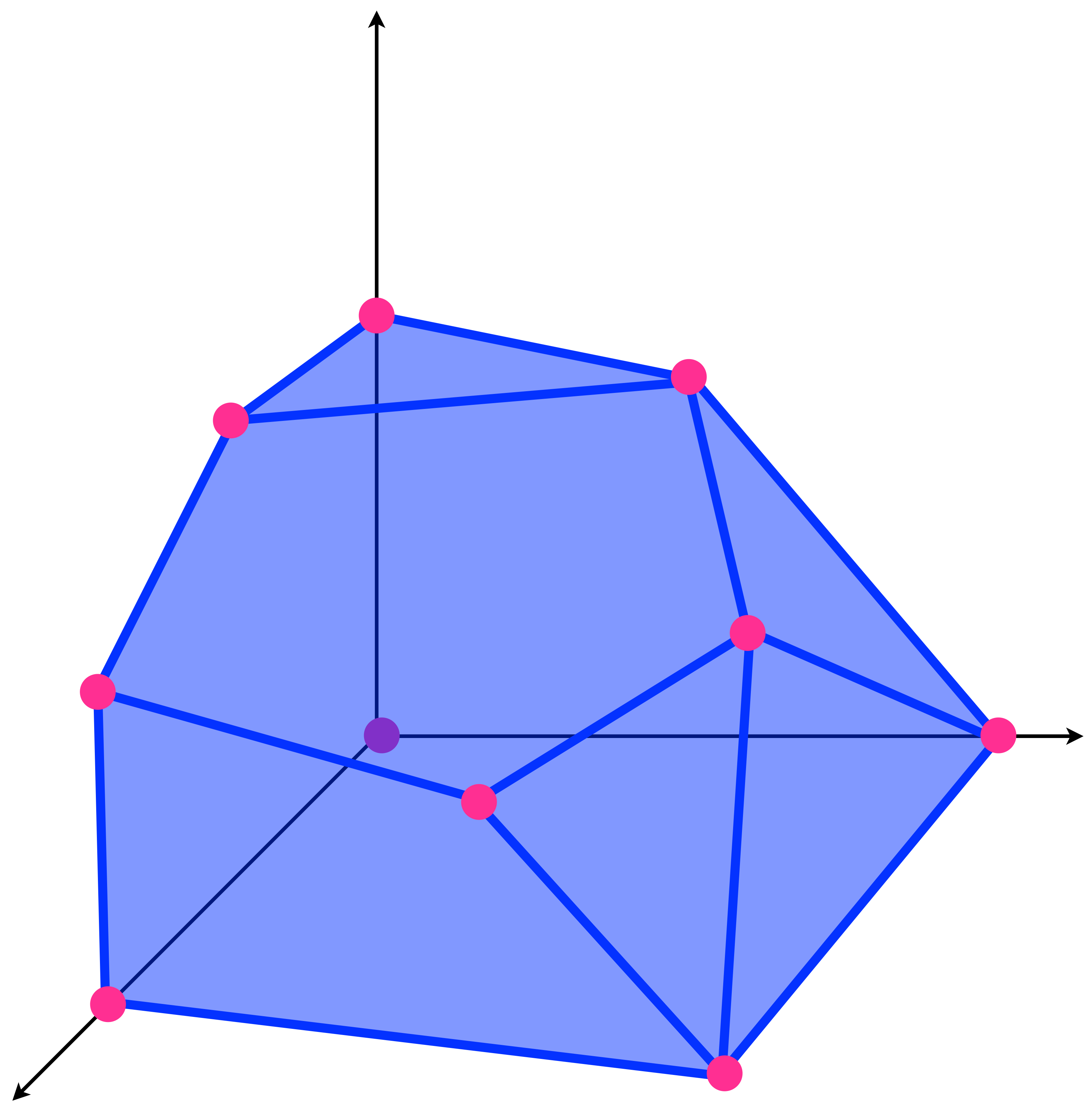
- (*closed, affine*) **halfspace**:

  $$\sum_{j=1}^{n} a_j x_j \geq b$$

- **convex polyhedron**:

  intersection of finitely many halfspaces

- **convex polytope**: bounded convex polyhedron



Convex Polyhedron

- A set $S \subseteq \mathbb{R}^n$ is **convex** if $\lambda x + (1 - \lambda)y \in S$ for all $x, y \in S$ and $\lambda \in [0,1]$. A **convex body** is a compact convex set.

- The **convex hull** of a set $S \subseteq \mathbb{R}^n$ is the smallest convex set $\supseteq S$.

- **Affine subspace:** $\{x \in \mathbb{R}^n \mid c^{\mathrm{T}}x \geq b\}$ for $c \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ and $b \in \mathbb{R}$.

- **Convex polyhedron:** $\{x \in \mathbb{R}^n \mid Ax \geq b\}$ for $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

- **Polytope:** convex hull of a finite $V \subseteq \mathbb{R}^n \iff$ bounded polyhedron

- **Vertex:** point x in a convex polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ such that $\exists c \in \mathbb{R}^n$ s.t. $c^{\mathrm{T}}x < c^{\mathrm{T}}y$ for all $y \in P$ with $y \neq x$

- **Extreme point:** a point x in a convex polyhedron P that cannot be expressed as a convex combination of any other $y, z \in P$

Convex Polyhedron

Proposition (existence of vertex):

A convex polyhedron P has a vertex iff it does not contain a line, i.e. there're no $x, y \in \mathbb{R}^n$ s.t. $x + \lambda y \in P$ for all $\lambda \in \mathbb{R}$.

Proposition (vertex = extreme point):

For nonempty convex polyhedron P, $\forall x \in P$:
x is a vertex $\Longleftrightarrow$ x is an extreme point

Proposition (vertex as optimal solution):

If the convex polyhedron $\{x \in \mathbb{R}^n \mid Ax \geq b\}$ has a vertex and the LP $\min\{c^{\mathrm{T}}x \mid Ax \geq b\}$ has an optimal solution, then there is an optimal solution that is a vertex.

Linear Programming (LP)

Canonical form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

$$\Longrightarrow$$

Standard form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i \quad \Longrightarrow \quad \begin{cases} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} + s_i = b_i \\ \qquad\quad s_i \geq 0 \end{cases}$$

slack variable

$$A \quad \Longrightarrow \quad A' = \begin{bmatrix} A & I \end{bmatrix}$$

Basic Feasible Solutions (*bfs*)

Standard form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

WLOG:

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

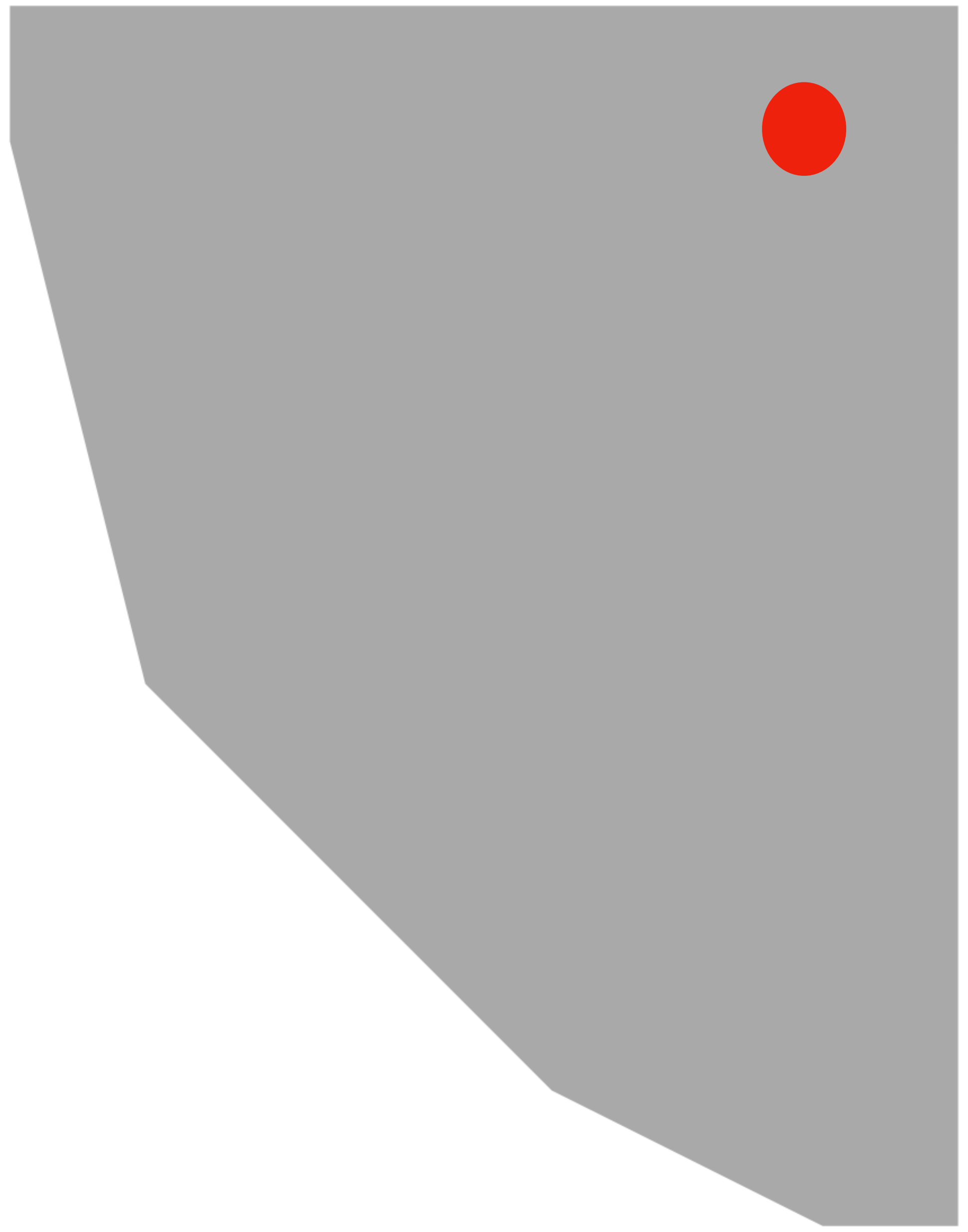
$$\boldsymbol{b} \in \mathbb{R}^m \qquad \boldsymbol{c} \in \mathbb{R}^n$$

$$m = \mathrm{rank}(A) \leq n$$

- Basis $B \subseteq [n]$: a set of m linearly independent columns of A
- Basic solution $\boldsymbol{x} \in \mathbb{R}^n$: $A_{[m]\times B}\boldsymbol{x}_B = \boldsymbol{b}$ and $\boldsymbol{x}_{[n]\setminus B} = \boldsymbol{0}$
  - $\implies A\boldsymbol{x} = \boldsymbol{b}$ but not necessarily $\boldsymbol{x} \geq \boldsymbol{0}$
- Basic feasible solution (*bfs*): a basic solution satisfying $\boldsymbol{x} \geq \boldsymbol{0}$

$\boldsymbol{x}$ is a *bfs* of the LP $\min\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \mid A\boldsymbol{x} = \boldsymbol{b} \wedge \boldsymbol{x} \geq \boldsymbol{0}\} \iff$
$\boldsymbol{x}$ is a vertex of the polyhedron $\{\boldsymbol{x} \mid A\boldsymbol{x} = \boldsymbol{b} \wedge \boldsymbol{x} \geq \boldsymbol{0}\}$

The Simplex Algorithm



Standard form:

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

WLOG:

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

$$\boldsymbol{b} \in \mathbb{R}^{m} \quad \boldsymbol{c} \in \mathbb{R}^{n}$$

$$m = \mathrm{rank}(A) \leq n$$

- Two *bfs*'s are **neighbors** if their bases share $m-1$ columns of A

Simplex Algorithm (Dantzig 1947):

start at a *bfs* $\boldsymbol{x}$;
while $\exists$ a **neighboring** *bfs* $\boldsymbol{x}'$ with $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}' < \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$:
    move to one of such $\boldsymbol{x}'$;

- Stops at a *local* optima $\Longrightarrow$ a ***global*** optima (by convexity)

Linear Programming (LP) Solvers

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

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

$\boldsymbol{b} \in \mathbb{R}^{m} \quad \boldsymbol{c} \in \mathbb{R}^{n}$

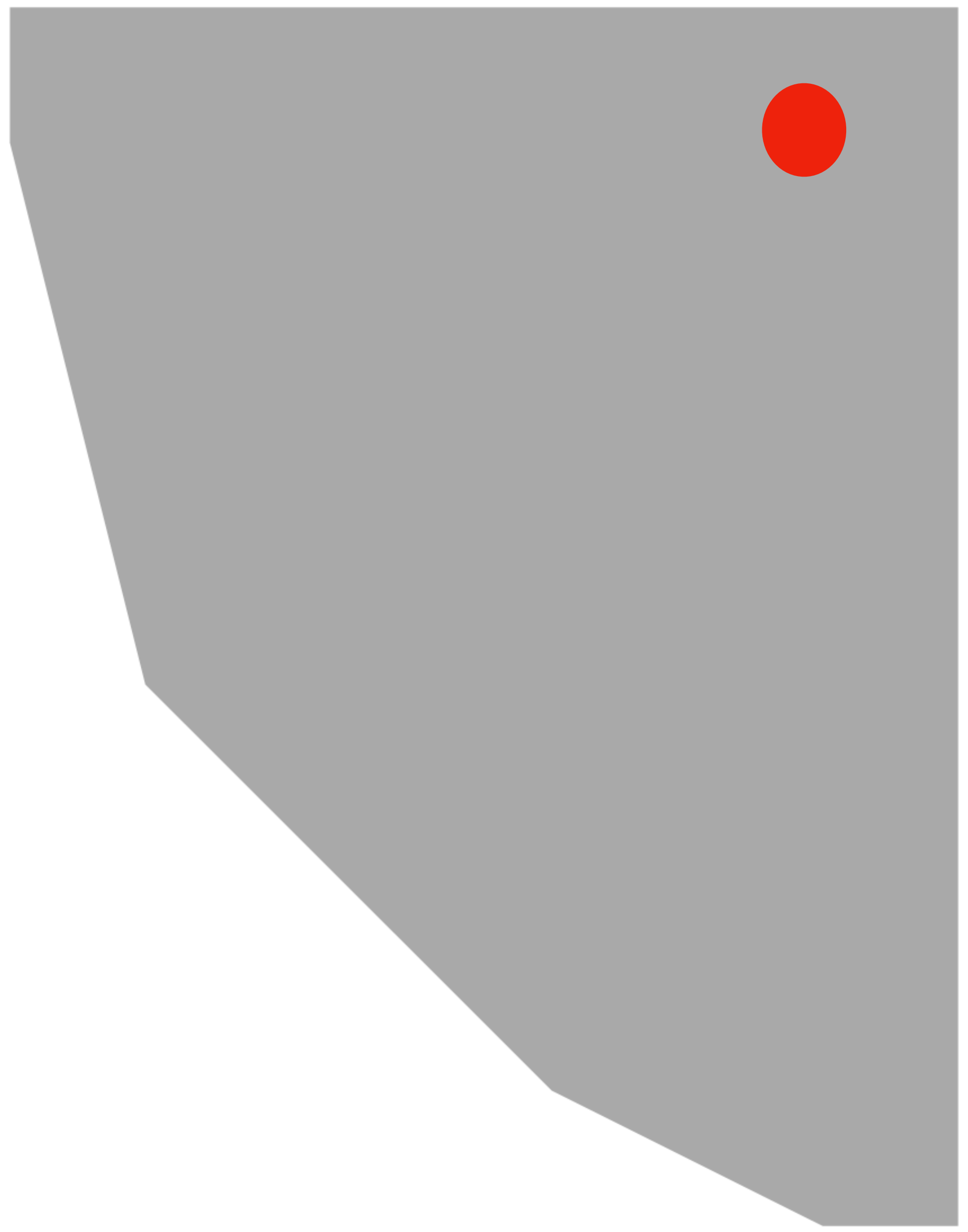
$m = \operatorname{rank}(A) \leq n$

- Dantzig's **simplex method** [Dantzig '47]:
  - walks over polytope vertices along polytope edges
  - exponential time in the worst case (Klee–Minty cube, 1972)
  - poly-time in *smoothed* complexity [Spielman-Teng'01]
- Solvable in (*weakly*) polynomial time:
  - **ellipsoid method** [Khachiyan '80] in $O(n^6)$ time
  - **interior-point methods** [Karmarkar '84] in $O(n^{2.5})$ time [Vaidya '89] and recently, in current matrix multiplication time [Cohen, Lee, Song '19] [Jiang, Song, Weinstein, Zhang '21]

Interior Point Method

Interior Point Method (Karmarkar 1984)**:**

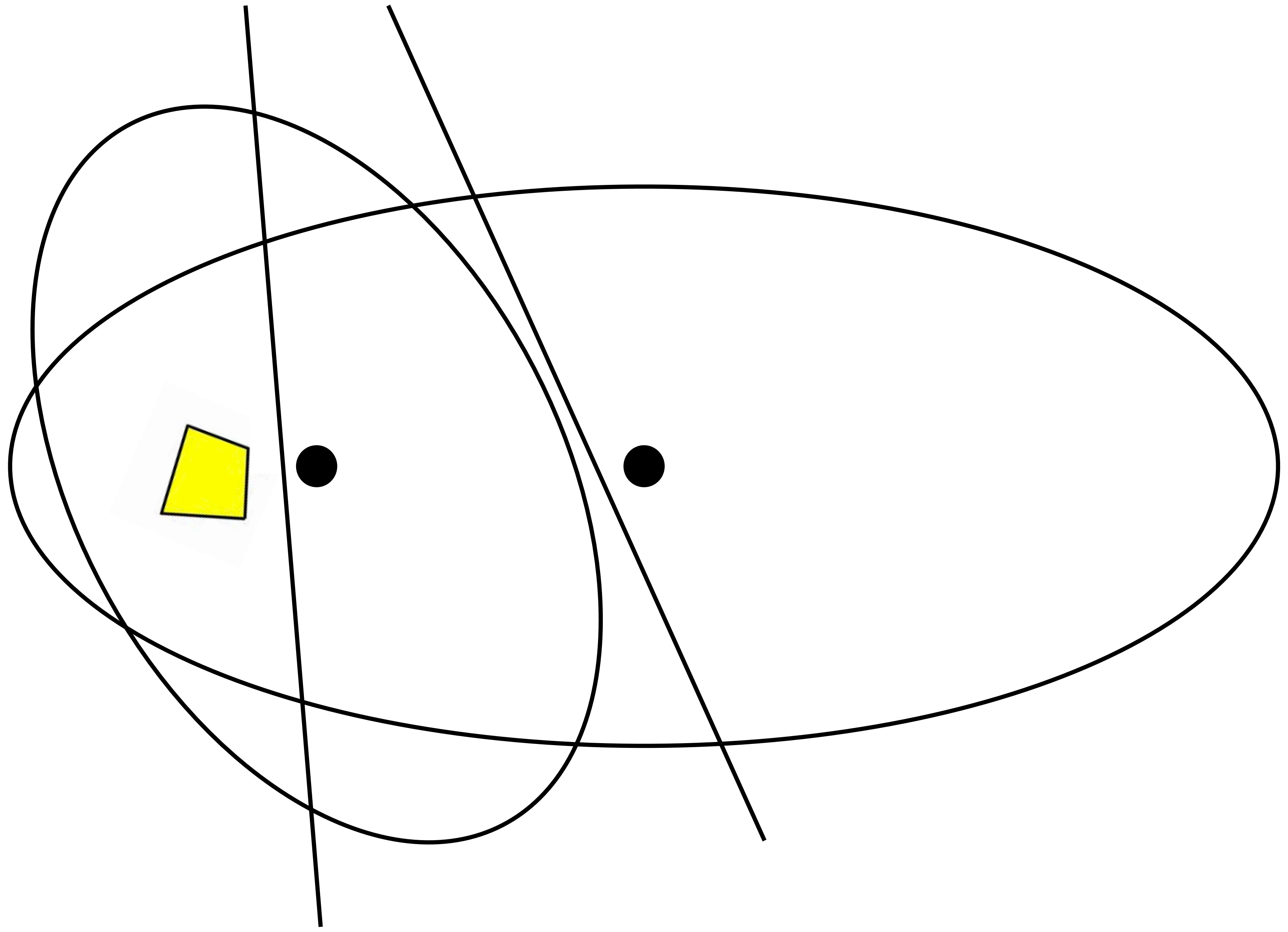
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Ellipsoid Method

Ellipsoid Method (Khachiyan 1979)**:**

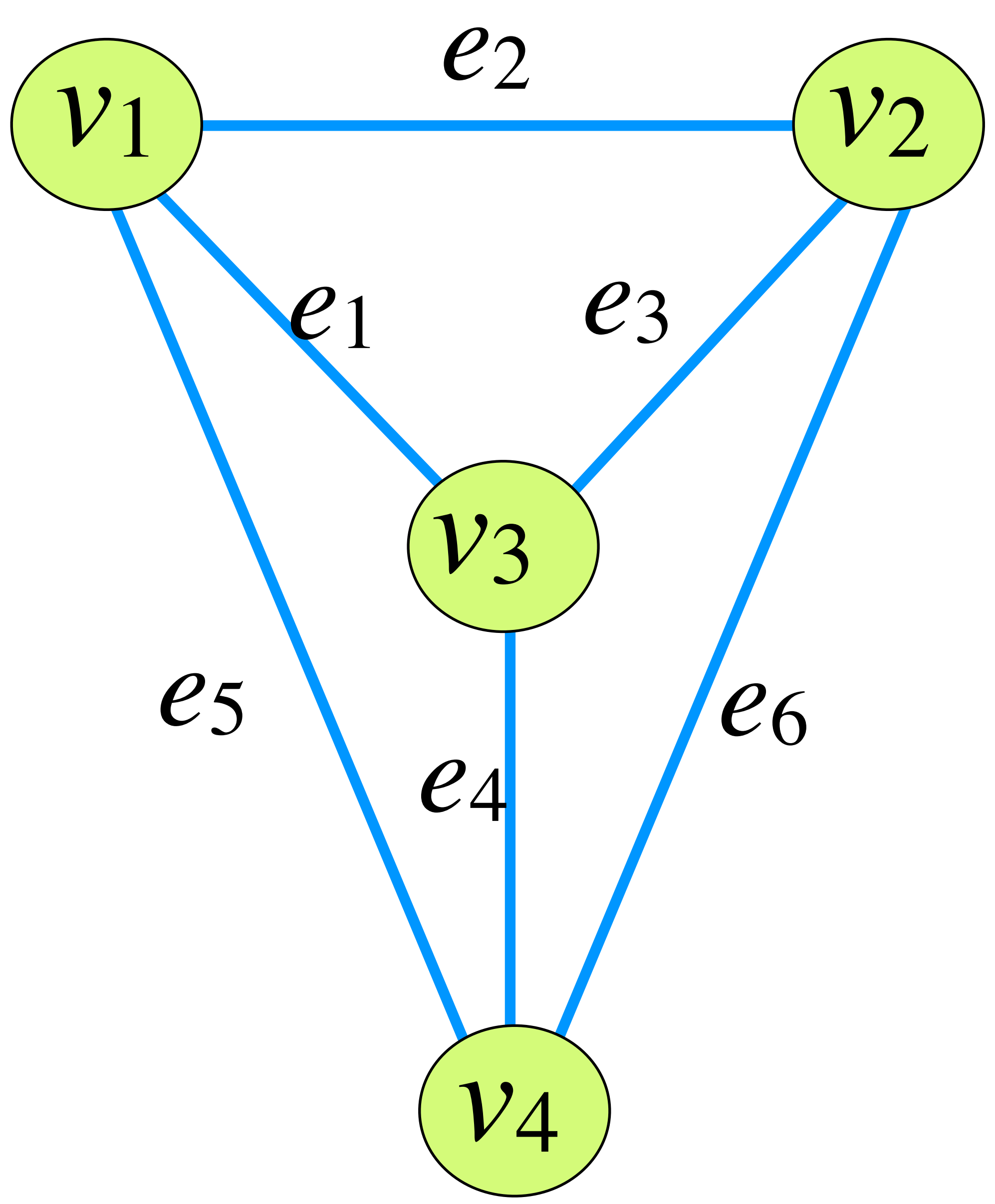
- maintain an ellipsoid that contains the feasible region
- cut the ellipsoid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Applications of Linear Programming

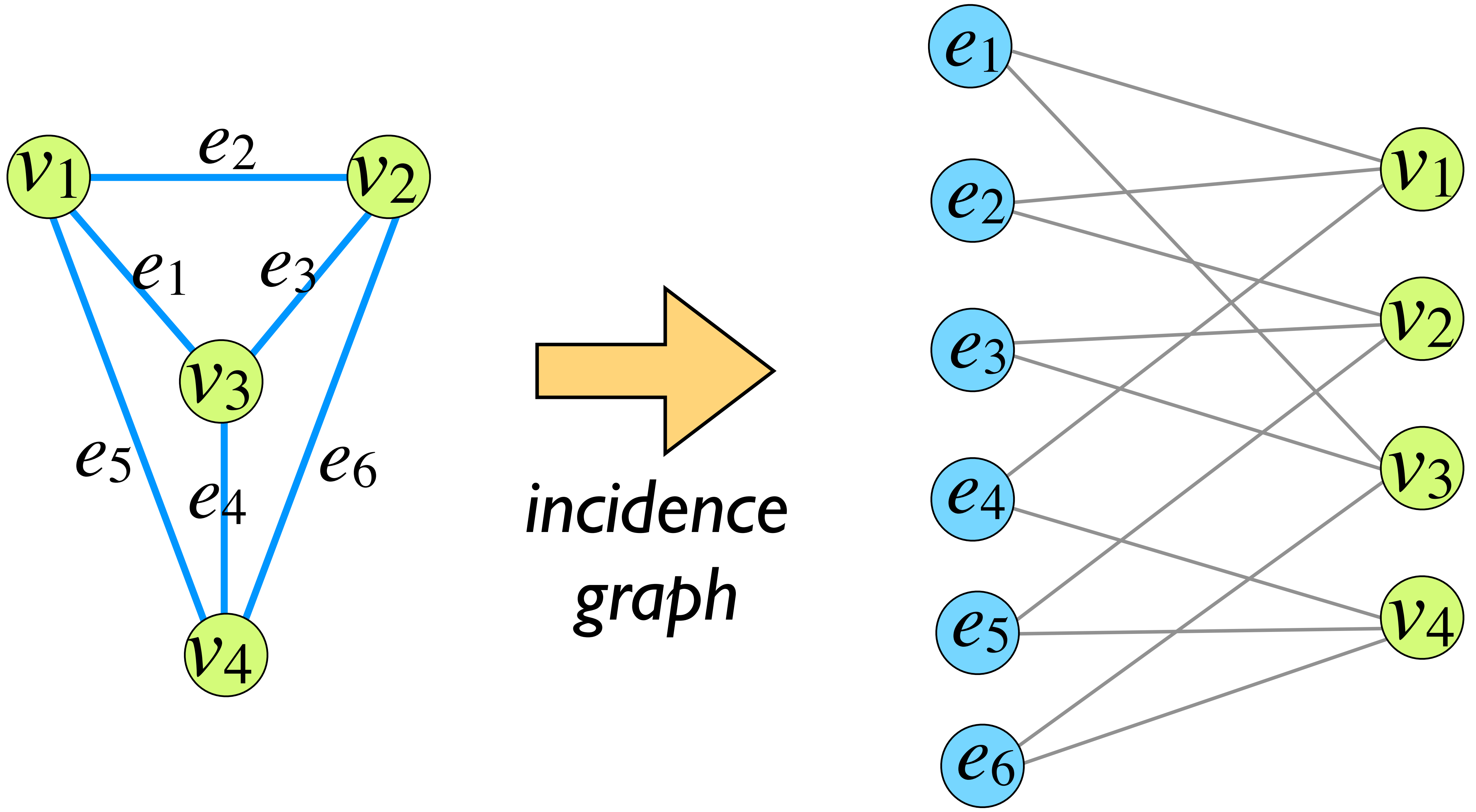
- Domain: computer science, mathematics, operations research, economics
- Types of problems: transportation, scheduling, clustering, network routing, resource allocation, facility location
- Research directions:
    - polynomial time exact algorithm
    - polynomial time approximation algorithm
    - sub-routines for the branch-and-bound method for integer programming
    - other algorithmic models: online algorithm, distributed algorithms, dynamic algorithms, fast algorithms

Vertex Cover



Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.



set cover instance
with *frequency* =2

Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- $\mathbf{NP}$-hard
- $\ln(n)$-approximation by greedy set cover
- 2-approximation algorithm:

Find a ***maximal matching***;
return the ***matched*** vertices;

- [Khot, Regev 2008] Assuming the unique games conjecture, there is no poly-time $(2 - \epsilon)$-approximation algorithm.

Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- **Integer Linear Program (ILP)** for vertex cover:

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{v \in V} x_v & \text{linear objective function} \\
\text{subject to} & \displaystyle\sum_{v \in e} x_v \geq 1, \quad e \in E & \text{linear constraints} \\
& x_v \in \{0,1\}, \quad v \in V & \text{integer domains}
\end{array}
$$

- Solving integer linear program is **NP**-hard.

Vertex Cover

Instance: An undirected graph $G(V,E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- Linear Program (LP) *relaxation*:

$$\begin{array}{lll} \text{minimize} & \sum_{v \in V} x_v & \\ \text{subject to} & \sum_{v \in e} x_v \geq 1, & e \in E \\ & x_v \in [0,1], & v \in V \end{array}$$

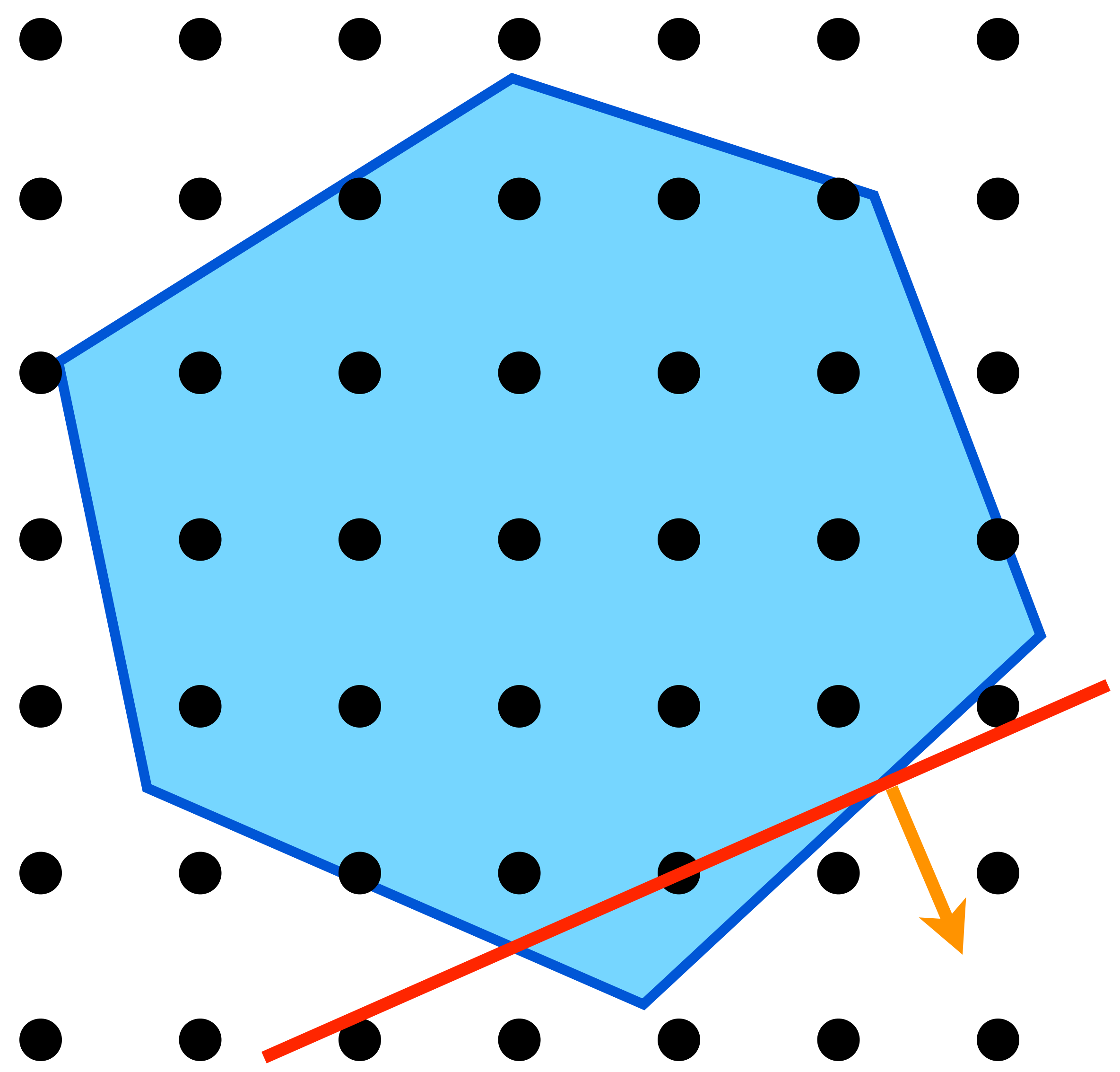
fractional domains

- linear programs are solvable in polynomial time!

Integrality

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \cancel{\boldsymbol{x} \in \mathbb{Z}^n} \end{aligned}$$

LP-relaxation



Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- **Integer Linear Program (ILP)** for vertex cover:

$$
\begin{array}{lll}
\text{minimize} & \sum_{v \in V} x_v & \\
\text{subject to} & \sum_{v \in e} x_v \geq 1, & e \in E \\
& x_v \in \{0,1\}, & v \in V
\end{array}
$$

- LP relaxation for Minimum vertex cover of $G(V,E)$

$$\begin{aligned} &\text{minimize} && \sum_{v\in V} x_v && \\ &\text{subject to} && \sum_{v\in e} x_v \geq 1, && e \in E \\ & && x_v \in [0,1], && v \in V \end{aligned}$$

LP Relaxation & Rounding:

find optimal solution $\boldsymbol{x}^* \in [0,1]^V$ of **LP relaxation**;
round $\boldsymbol{x}^*$ to an feasible *integral* solution $\hat{\boldsymbol{x}} \in \{0,1\}^V$:

$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

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

$$\text{s.t.} \quad \sum_{v\in e} x_v \geq 1, \qquad e \in E$$

$$x_v \in [0,1], \qquad v \in V$$

LP Relax & Round:

find **OPT** $\boldsymbol{x}^* \in [0,1]^V$;

round $\boldsymbol{x}^*$ to **feasible *integral*** $\hat{\boldsymbol{x}}$:

$$\hat{x}_v = \underbrace{\begin{cases} 1 & \text{if } x_v^* \geq 0.5 \\ 0 & \text{otherwise} \end{cases}}_{\leq 2x_v^*}$$

- **Soundness** of rounded solution $\hat{\boldsymbol{x}}$ (as a vertex cover):

$$\sum_{v\in e} x_v^* \geq 1 \implies \sum_{v\in e} \hat{x}_v \geq 1$$

- **Approximation ratio:**

$$OPT = OPT_{Int} \geq OPT_{LP} = \sum_{v\in V} x_v^*$$

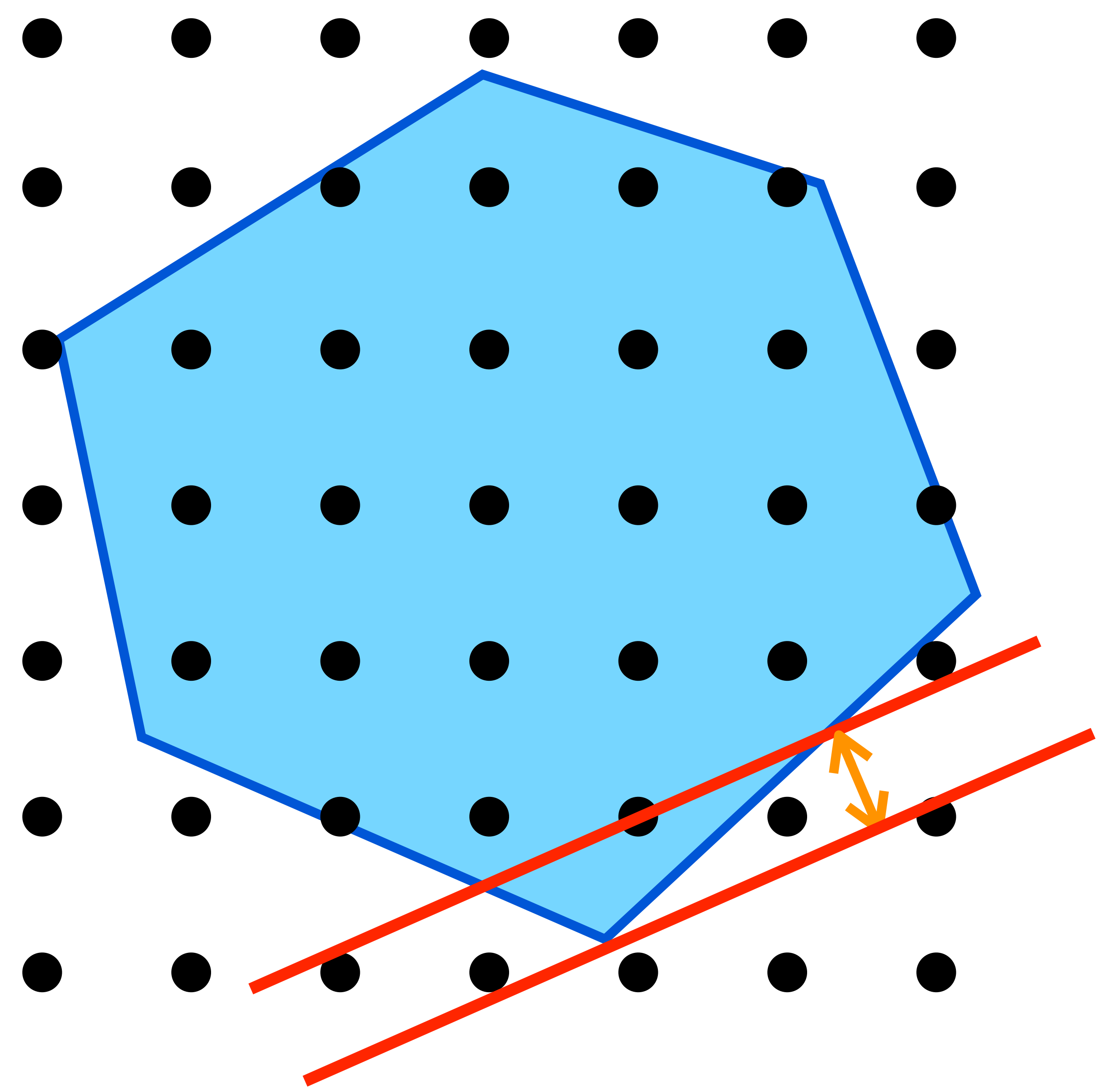
$$SOL = \sum_{v\in V} \hat{x}_v \leq 2\sum_{v\in V} x_v^* \leq 2OPT$$

LP Relaxation & Rounding

- **Modeling**: Express the optimization problem as an **Integer Linear Program (ILP)**.
- **Relaxation**: Relax the **ILP** to a **Linear Program** (**LP**).
- **Solving**: Find the ***optimal*** solution by an efficient LP solver.
- **Rounding**: Round the optimal solution to a ***feasible integral*** solution.
- **Analysis**: Prove that the rounded solution is not too far away from the ***optimal integral*** solution (usually by **comparing with the optimal solution**).

Integrality Gap

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \cancel{\boldsymbol{x} \in \mathbb{Z}^n} \end{aligned}$$



$$\text{integrality gap} = \sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_{\mathrm{LP}}(I)}$$

Integrality Gap

- minimum vertex cover of $G(V,E)$:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{v\in V} x_v & \\ \text{subject to} & \displaystyle\sum_{v\in e} x_v \geq 1, & e \in E \\ & x_v \in \{0,1\}, & v \in V \end{array}$$

$$\text{integrality gap} = \sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_{\mathrm{LP}}(I)}$$

- The 2-approx. LP-rounding algorithm shows integrality gap ≤ 2

Because the analysis compares the relaxed OPT with an integral feasible solution (output of the algorithm)

Integrality Gap

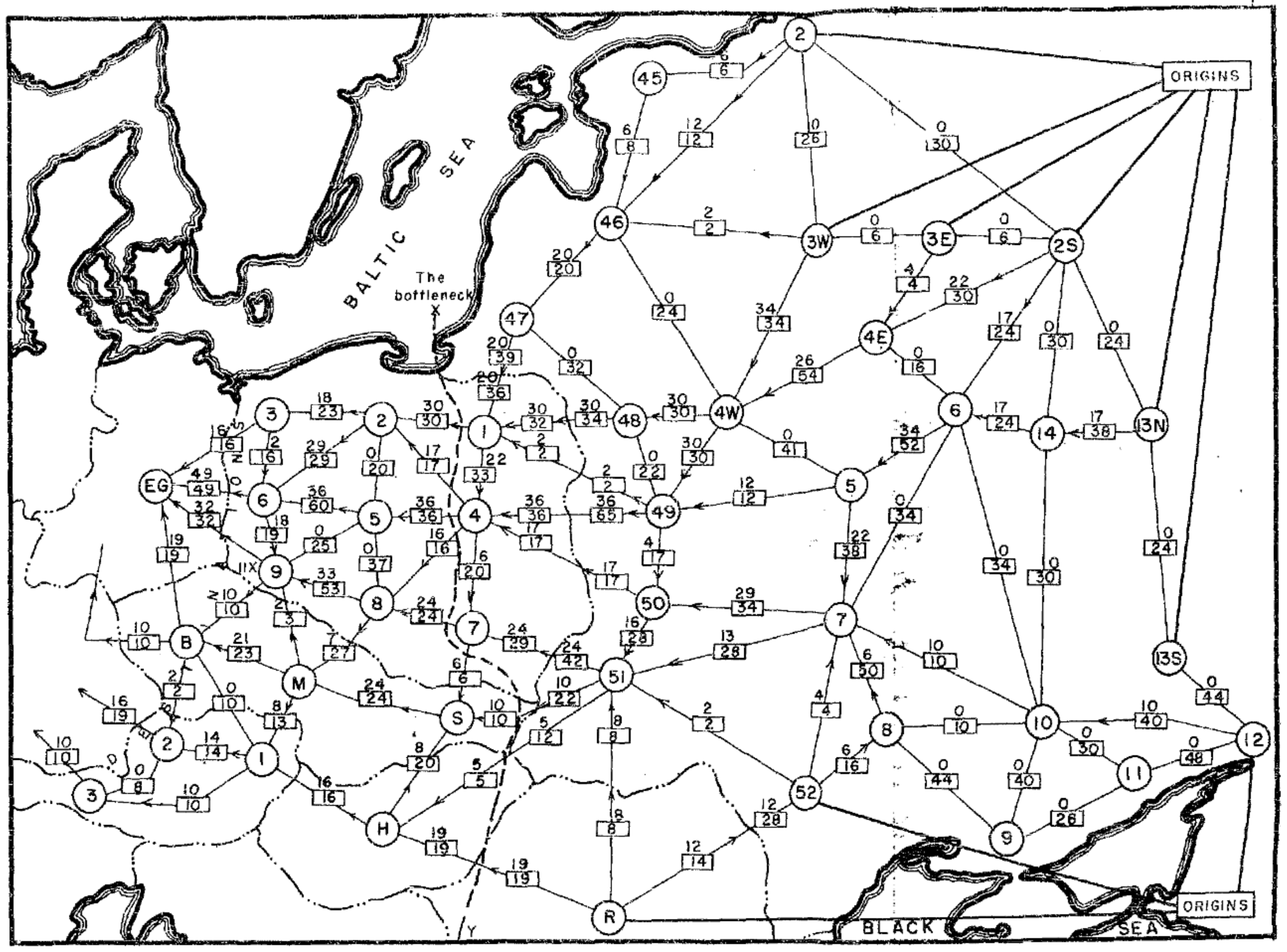
- minimum vertex cover of $G(V,E)$:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{v\in V} x_v & \\ \text{subject to} & \displaystyle\sum_{v\in e} x_v \geq 1, & e \in E \\ & x_v \in \{0,1\}, & v \in V \end{array}$$

$$\text{integrality gap} = \sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_{\mathrm{LP}}(I)}$$

- For LP relaxation of **vertex cover**: integrality gap $= 2$
- [Singh '19] int. gap on $G = \left(2 - \dfrac{2}{\chi^f(G)}\right)$ fractional chromatic number

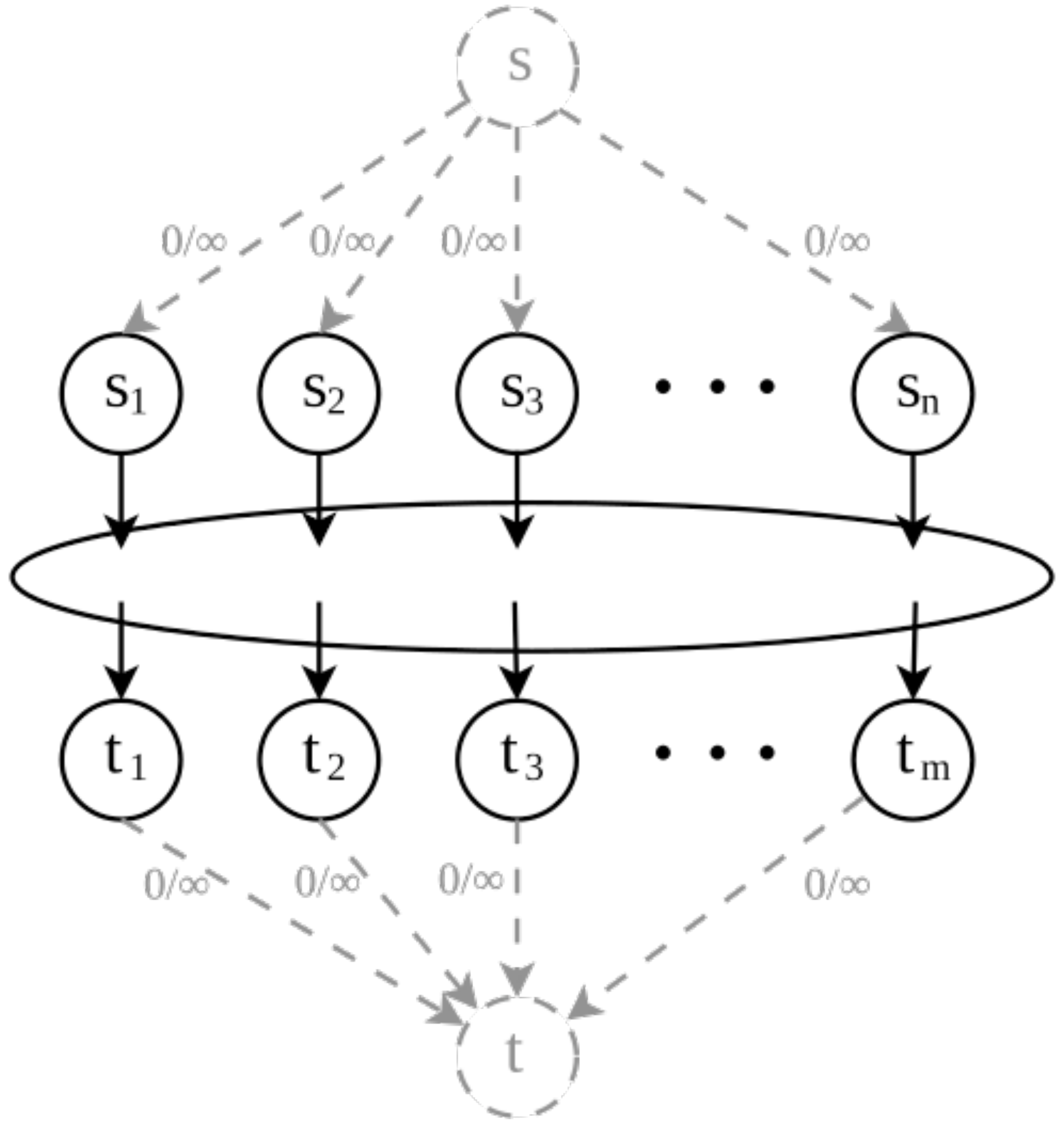
Network Flow (Recap)



Multi-Source Multi-Sink

Input: A network $G = (V, E)$ and capacity $c : E \to \mathbb{R}^+$, with sources $S = \{s_1, \ldots, s_n\}$ and sinks $T = \{t_1, \ldots, t_m\}$.

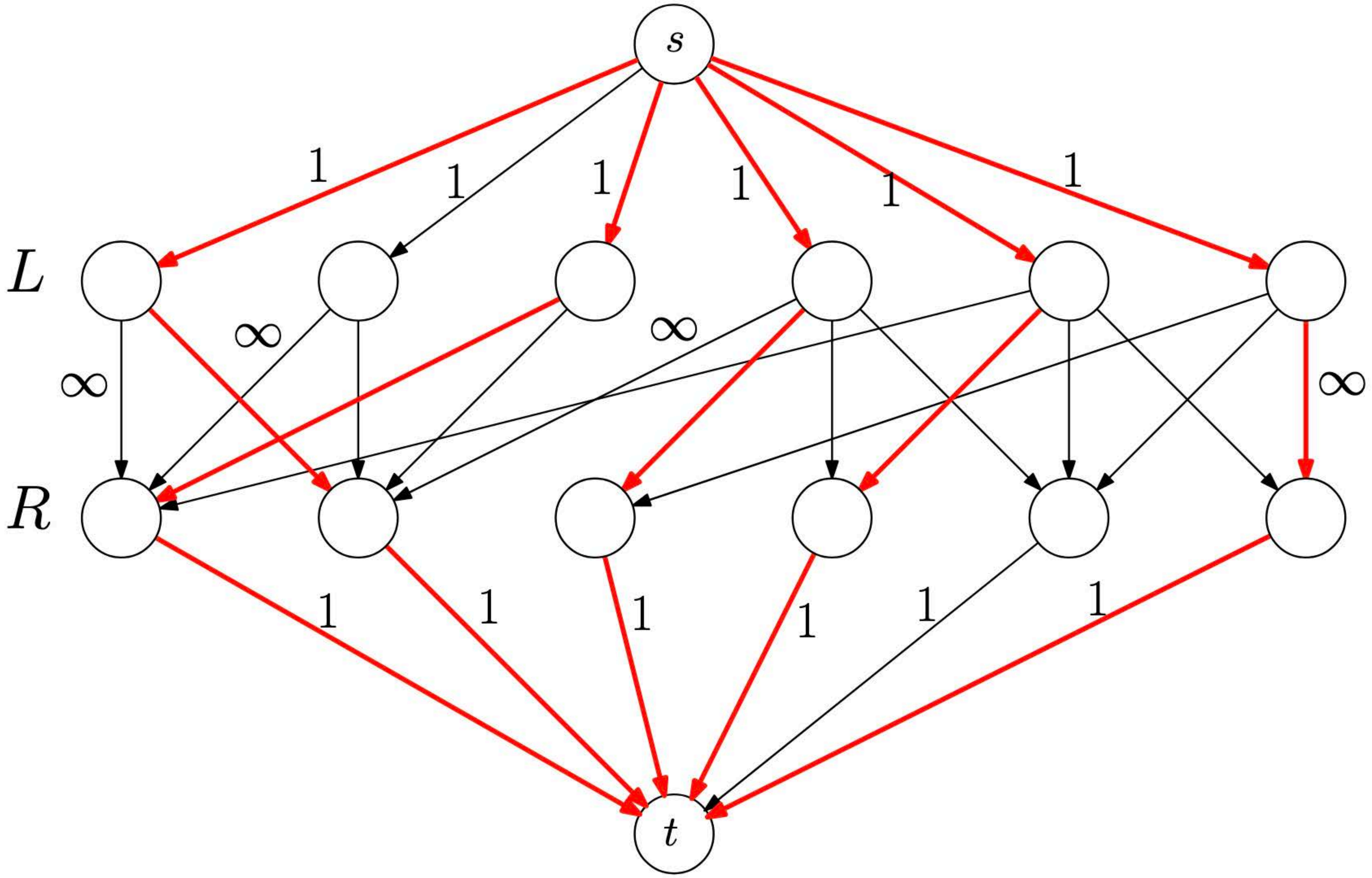
Output: A maximum flow from S to T across G.



Maximum Bipartite Matching

Input: A bipartite graph $G = (L \cup R, E)$.

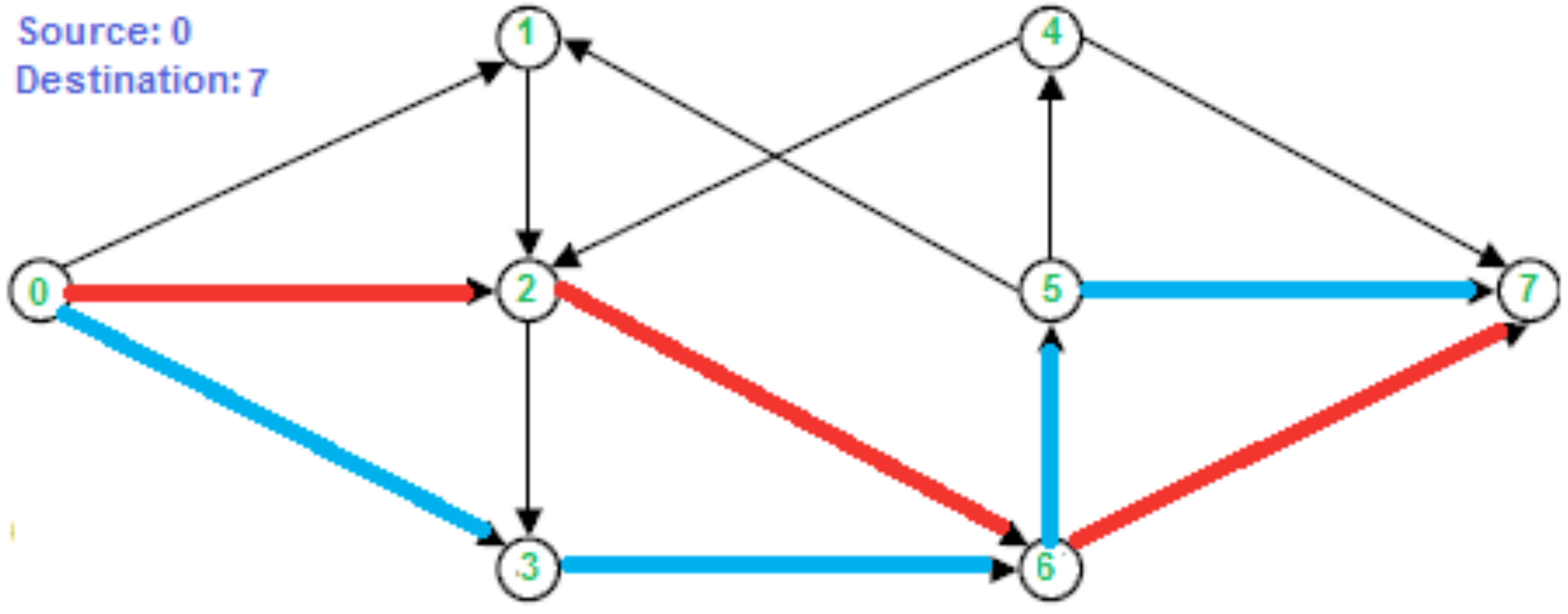
Output: A maximum matching of a bipartite graph.

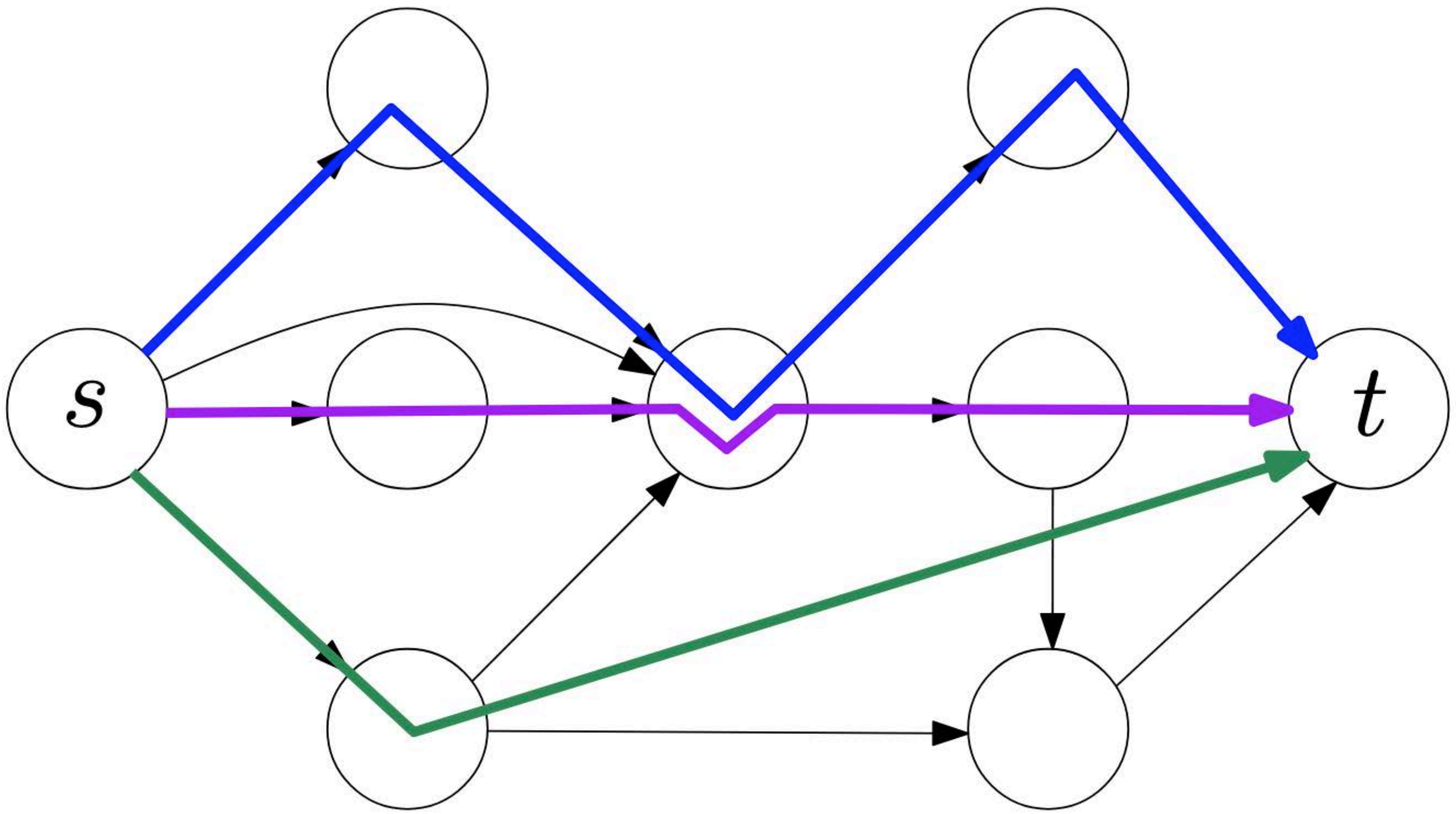


#Disjont Paths

Input: A digraph $G = (V, E)$ and source & sink $s, t \in V$.

Output: The maximum #**edge-disjoint** paths from s to t





Global Min-Cut

Input: A digraph $G = (V, E)$ and weights $c : E \to \mathbb{R}^+$.

Output: The minimum cut whose removal disconnects G

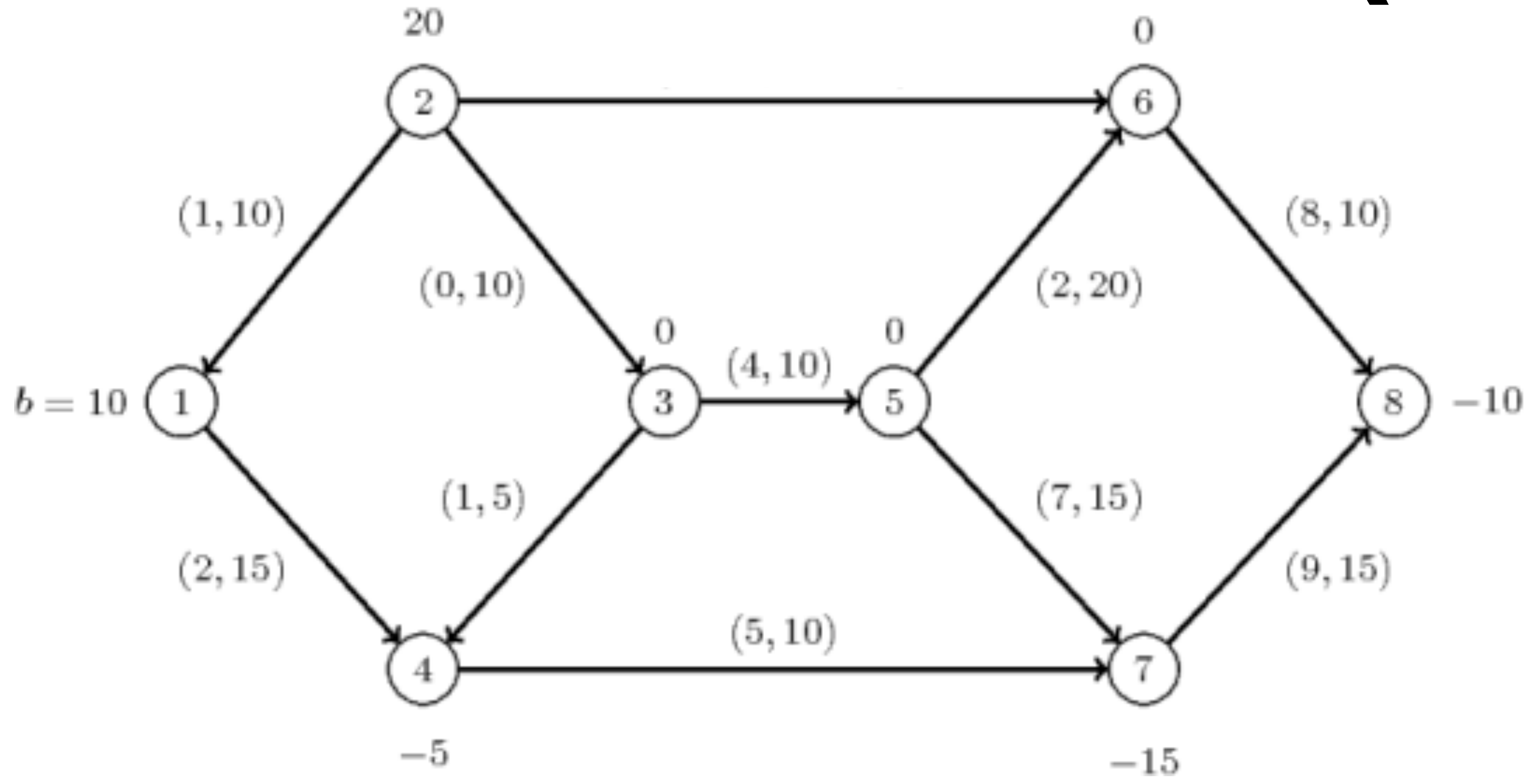
The following statements are equivalent:

(1) f is a maximum flow

(2) There is no s-t path P in G^f with $\delta(P) > 0$

(3) There is $S, \neg S \subseteq V$ such that cut$(S) = |f|$

Minimum-Cost Flow Problem (MCFP)



- Edge capacity $c_e \in \mathbb{R}^+$. **Flow cost** $a_e \in \mathbb{R}^+$.
- Flow function $f : E \to \mathbb{R}^+$
  - Valid flow: $\forall e \in E, 0 \leq f(e) \leq c_e$ & $\forall v \in V$, $\sum_{e \in \delta_{in}(v)} f(e) = \sum_{e \in \delta_{out}(v)} f(e)$
- MCFP: find a flow f minimizes $\sum_{e \in \delta_{out}(s)} a_e \cdot f(e)$ while maximizing $\sum_{e \in \delta_{out}(s)} f(e)$

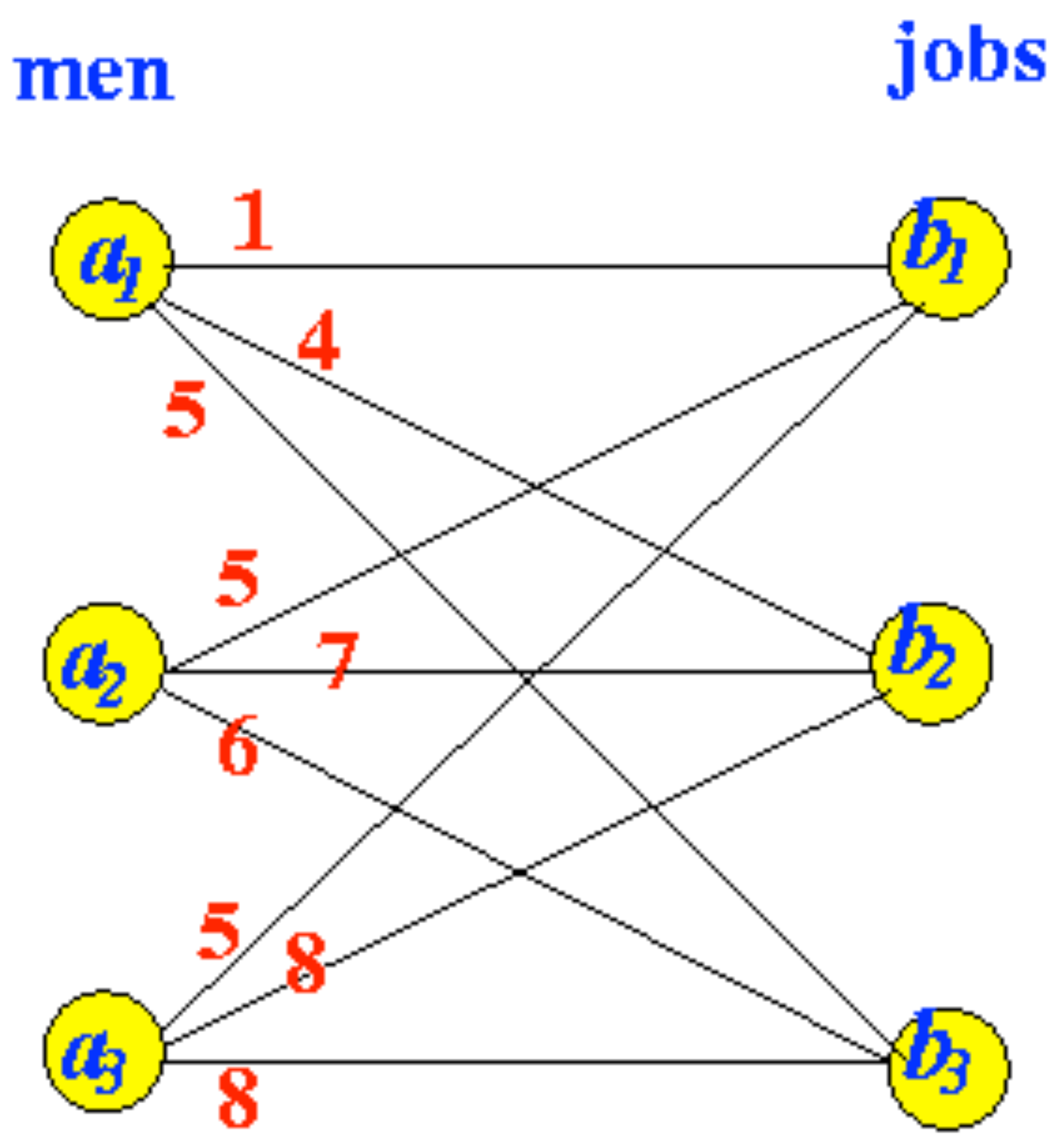
Assignment Problem

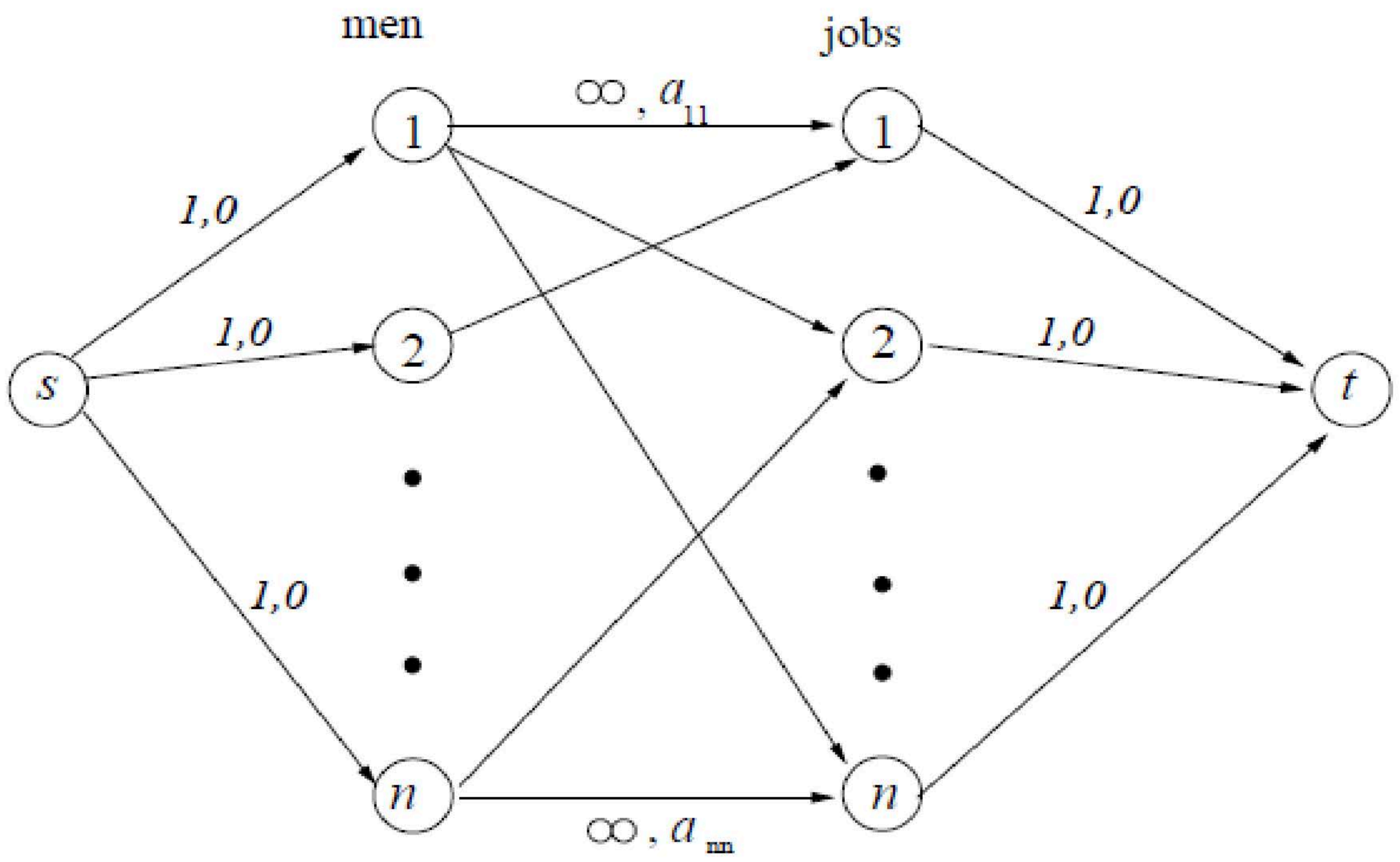
Minimum-cost perfect matching

Given matrix of costs

Worker	Task I	Task II	Task III
Ali	8	4	7
Baba	5	2	3
Curi	9	6	7
Durian	9	4	8

Make square with dummy column.
Subtract minimum for each column:





Minimum-Cost Flow Problem (MCFP)

Input: A digraph $G = (V, E)$ and capacity & costs $c, a : E \to \mathbb{R}^+$,
and source s, sink t.

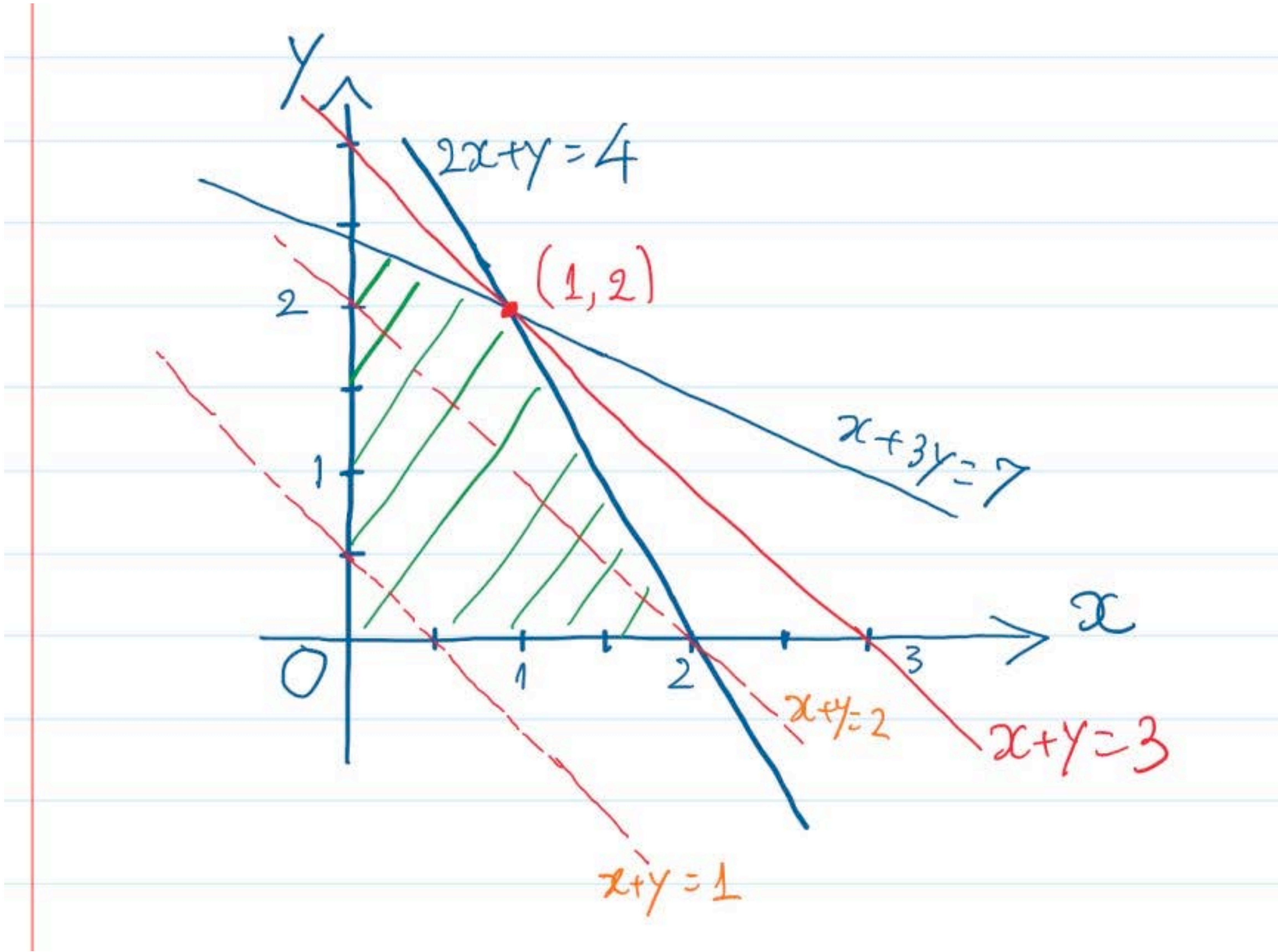
Output: A flow f minimizes $\sum_{e \in \delta_{out}(s)} a_e \cdot f(e)$ while maximizing $\sum_{e \in \delta_{out}(s)} f(e)$

- An improving redirection be like:
  - a negative cycle in the residual graph

Cycle Canceling:

While exists negative cycle in residual graph
cancel the negative cycle by redirecting

Linear Programming (Recap)



A Familiar Problem

某厂生产甲乙两种产品，每生产 1 吨产品的电耗、煤耗、所需劳动力及产值如表 3 所示：

表 3

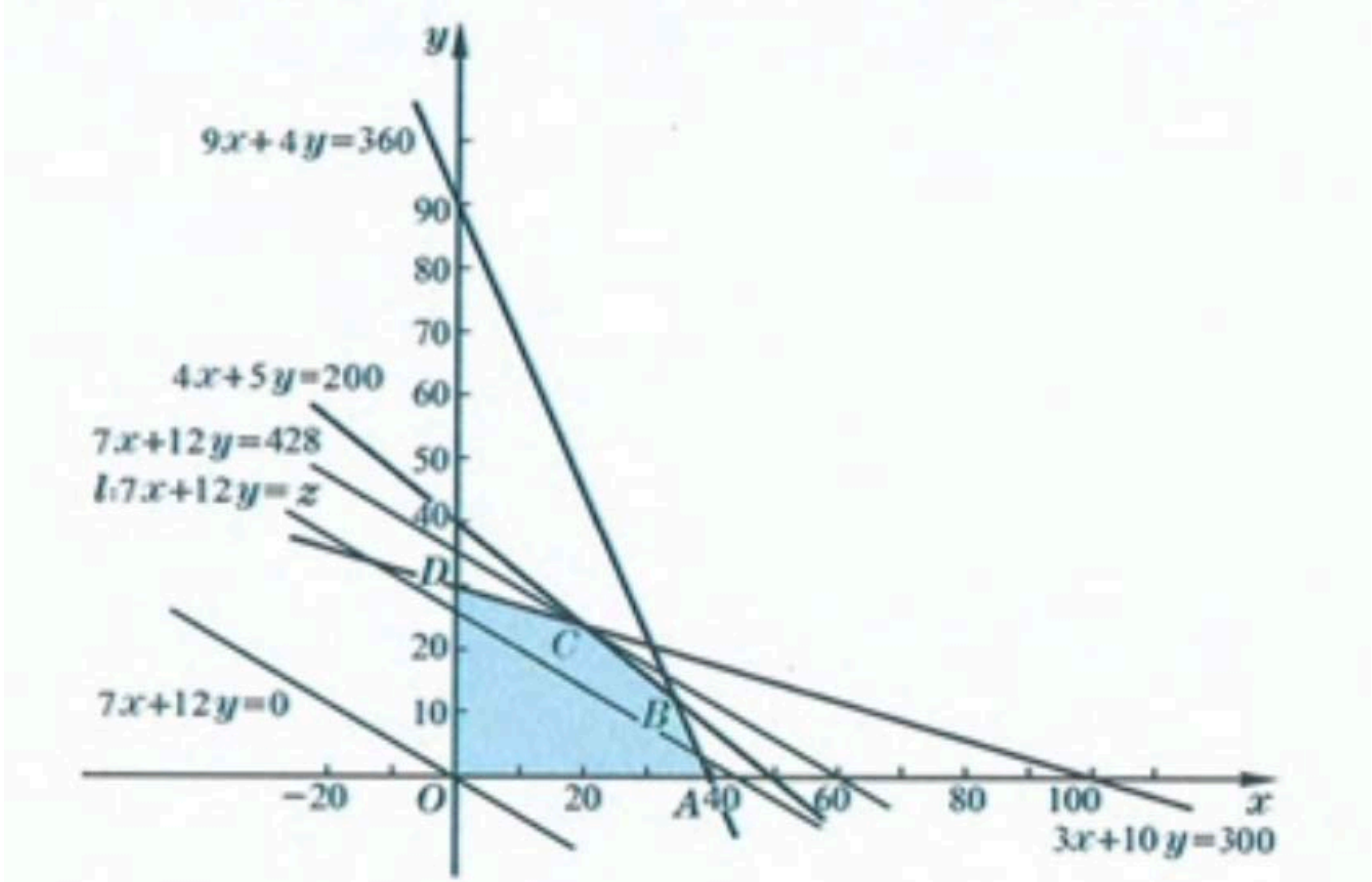
产品	电耗（千瓦时）	煤耗（吨）	劳动力（人）	产值（万元）
甲	4	9	3	7
乙	5	4	10	12

已知该厂有劳动力 300 人，按计划煤耗每天不超过 360 吨，电耗每天不超过 200 千瓦时. 每天应如何安排生产，可使产值最大？

如果设该厂每天生产甲产品 x 吨，乙产品 y 吨，那么上述问题可转化为在满足以下线性约束条件：

$$(\mathrm{B})\ \begin{cases} 9x+4y\leqslant 360, \\ 4x+5y\leqslant 200, \\ 3x+10y\leqslant 300, \\ x\geqslant 0, \\ y\geqslant 0, \end{cases}$$

求线性目标函数 $z=7x+12y$ 的最大值.

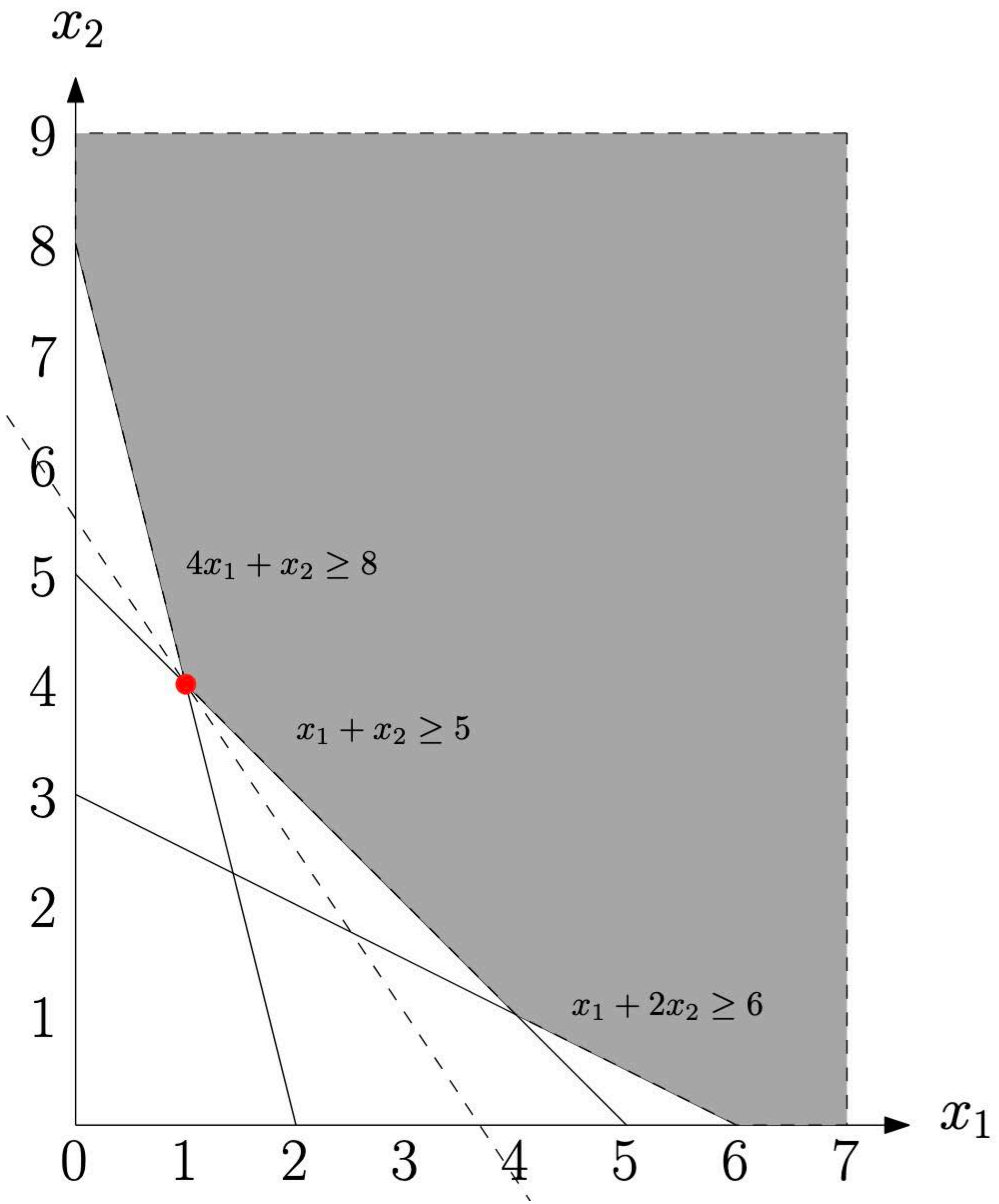


minimize $\quad 7x_1 + 4x_2$

subject to
$$x_1 + x_2 \geq 5$$
$$x_1 + 2x_2 \geq 6$$
$$4x_1 + x_2 \geq 8$$
$$x_1, x_2 \geq 0$$

optimal point $x_1 = 1, x_2 = 4$

value $= 7 \times 1 + 4 \times 4 = 23$



Linear Programming (LP)

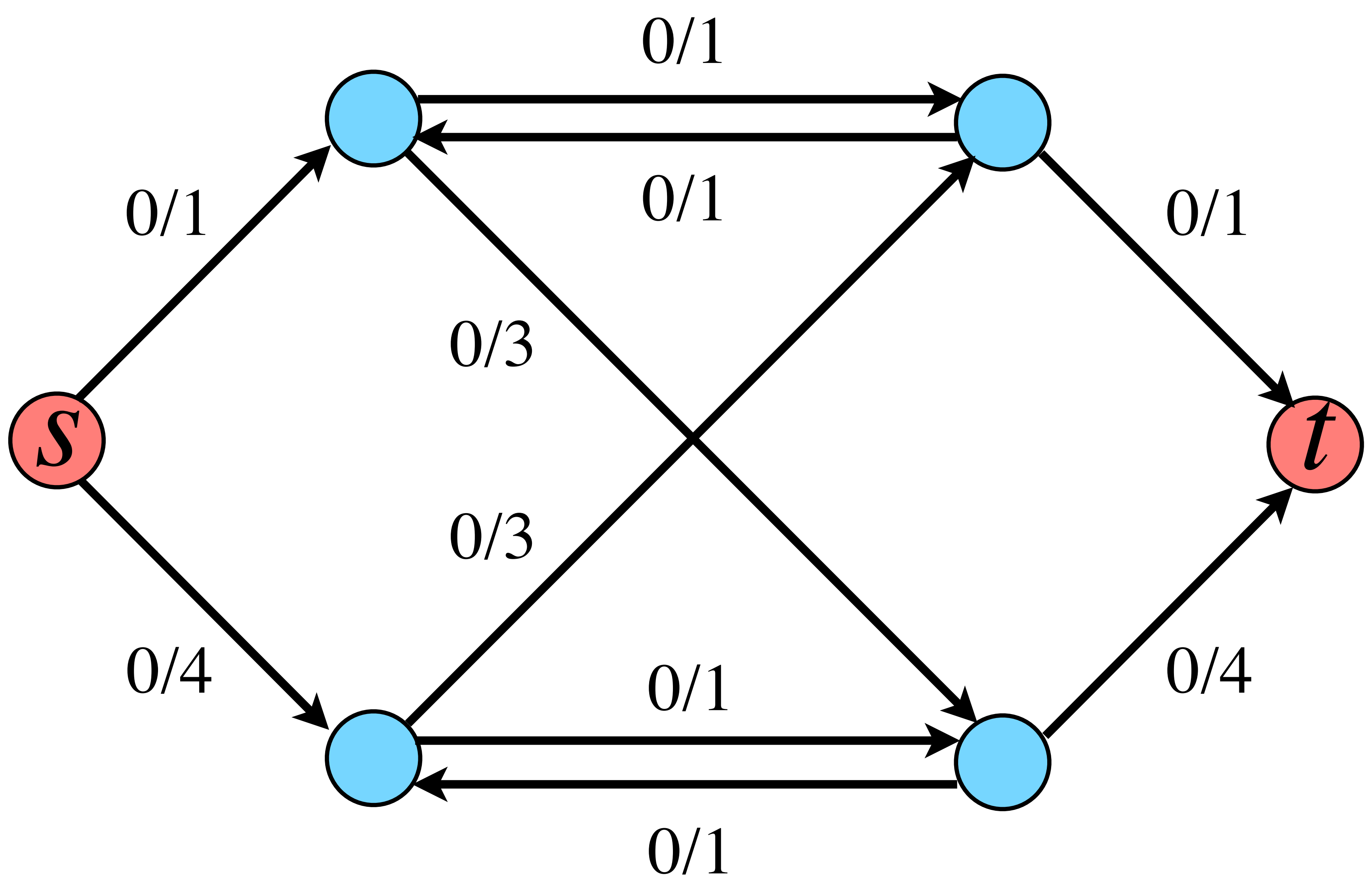
- General form:
  - matrix $A = \{a_{ij}\}_{[m]\times[n]}$, sets $M \subseteq [m]$ and $N \subseteq [n]$

$$
\begin{array}{rll}
\text{minimize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & \\
\text{subject to} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i & i \in M \\
& \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i & i \in \overline{M} \\
& x_j \geq 0 & j \in N \\
& x_j \text{ unconstrained} & j \in \overline{N}
\end{array}
$$

Max-Flow

digraph: $G = (V, E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



max $\displaystyle\sum_{u:(s,u)\in E} f_{su}$

s.t. $0 \le f_{uv} \le c_{uv}$ $\forall (u,v) \in E$

$$\sum_{w:(w,u)\in E} f_{wu} - \sum_{v:(u,v)\in E} f_{uv} = 0 \quad \forall u \in V \setminus \{s,t\}$$

Linear Programming (LP)

General form:

$$\begin{array}{lll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & \\ \text{s.t.} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i & i \in M \\ & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i & i \in \overline{M} \\ & x_j \geq 0 & j \in N \\ & x_j \text{ unconstrained} & j \in \overline{N} \end{array}$$

$\Longrightarrow$

Canonical form:

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i \implies \begin{cases} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i \\ -\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq -b_i \end{cases}$$

$$x_j \text{ unconstrained} \implies x_j = x_j^+ - x_j^- \text{ where} \begin{cases} x_j^+ \geq 0 \\ x_j^- \geq 0 \end{cases}$$

Solvable in Polynomial Time

Canonical Form of Linear programming

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Algorithm	Theory	Practice
Simplex Method	Exponential Time	Works Well
Ellipsoid Method	Polynomial Time	Slow
Internal Point Methods	Polynomial Time	Works Well

Linear Programming (LP)

Canonical form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

$$\Longrightarrow$$

Standard form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

$$\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i \quad \Longrightarrow \quad \begin{cases} \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} + s_i = b_i \\ s_i \geq 0 \end{cases}$$

slack variable

$$A \quad \Longrightarrow \quad A' = \begin{bmatrix} A & I \end{bmatrix}$$

Linear Programming (LP) Solvers

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

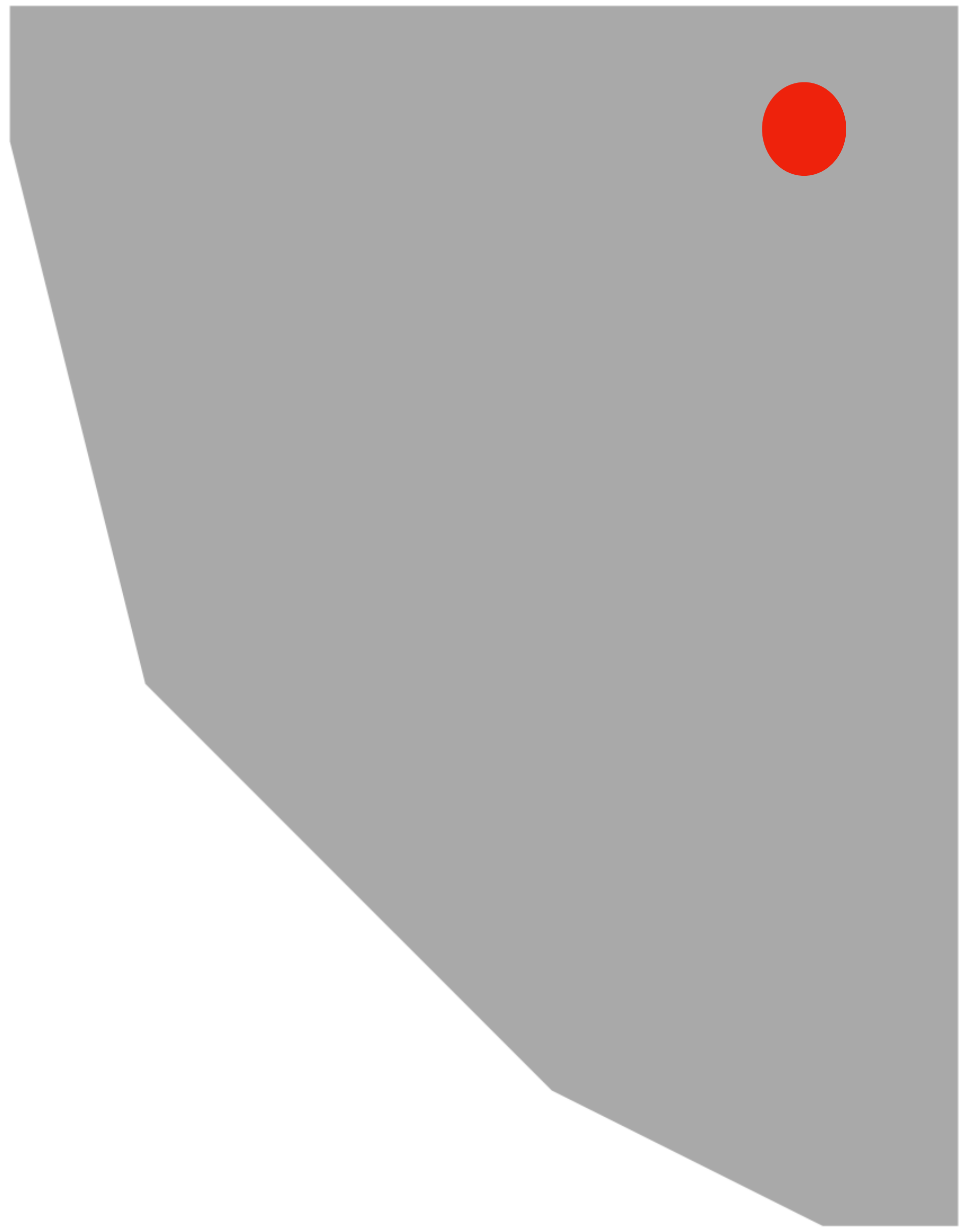
$A \in \mathbb{R}^{m \times n}$

$\boldsymbol{b} \in \mathbb{R}^m \quad \boldsymbol{c} \in \mathbb{R}^n$

$m = \operatorname{rank}(A) \leq n$

- Dantzig's **simplex method** [Dantzig '47]:
  - walks over polytope vertices along polytope edges
  - exponential time in the worst case (Klee–Minty cube, 1972)
  - poly-time in *smoothed* complexity [Spielman-Teng'01]
- Solvable in (*weakly*) polynomial time:
  - **ellipsoid method** [Khachiyan '80] in $O(n^6)$ time
  - **interior-point methods** [Karmarkar '84] in $O(n^{2.5})$ time [Vaidya '89] and recently, in current matrix multiplication time [Cohen, Lee, Song '19] [Jiang, Song, Weinstein, Zhang '21]

The Simplex Algorithm



Standard form:

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

WLOG:

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

$$\boldsymbol{b} \in \mathbb{R}^m \qquad \boldsymbol{c} \in \mathbb{R}^n$$

$$m = \mathrm{rank}(A) \leq n$$

- Two *bfs*'s are **neighbors** if their bases share $m-1$ columns of A

Simplex Algorithm (Dantzig 1947):

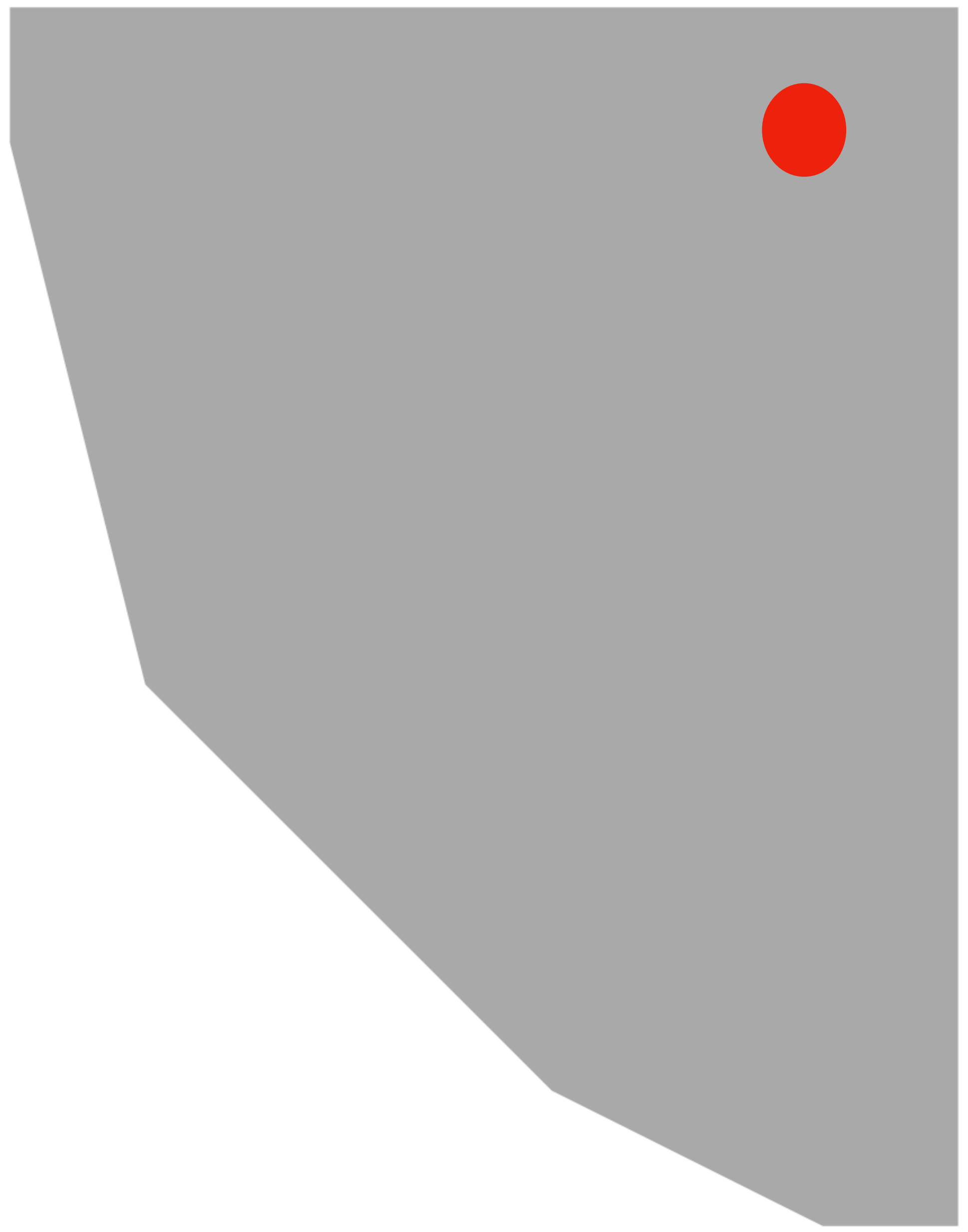
start at a *bfs* $\boldsymbol{x}$;
while $\exists$ a **neighboring** *bfs* $\boldsymbol{x}'$ with $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}' < \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$:
    move to one of such $\boldsymbol{x}'$;

- Stops at a *local* optima $\Longrightarrow$ a ***global*** optima (by convexity)

Interior Point Method

Interior Point Method (Karmarkar 1984):

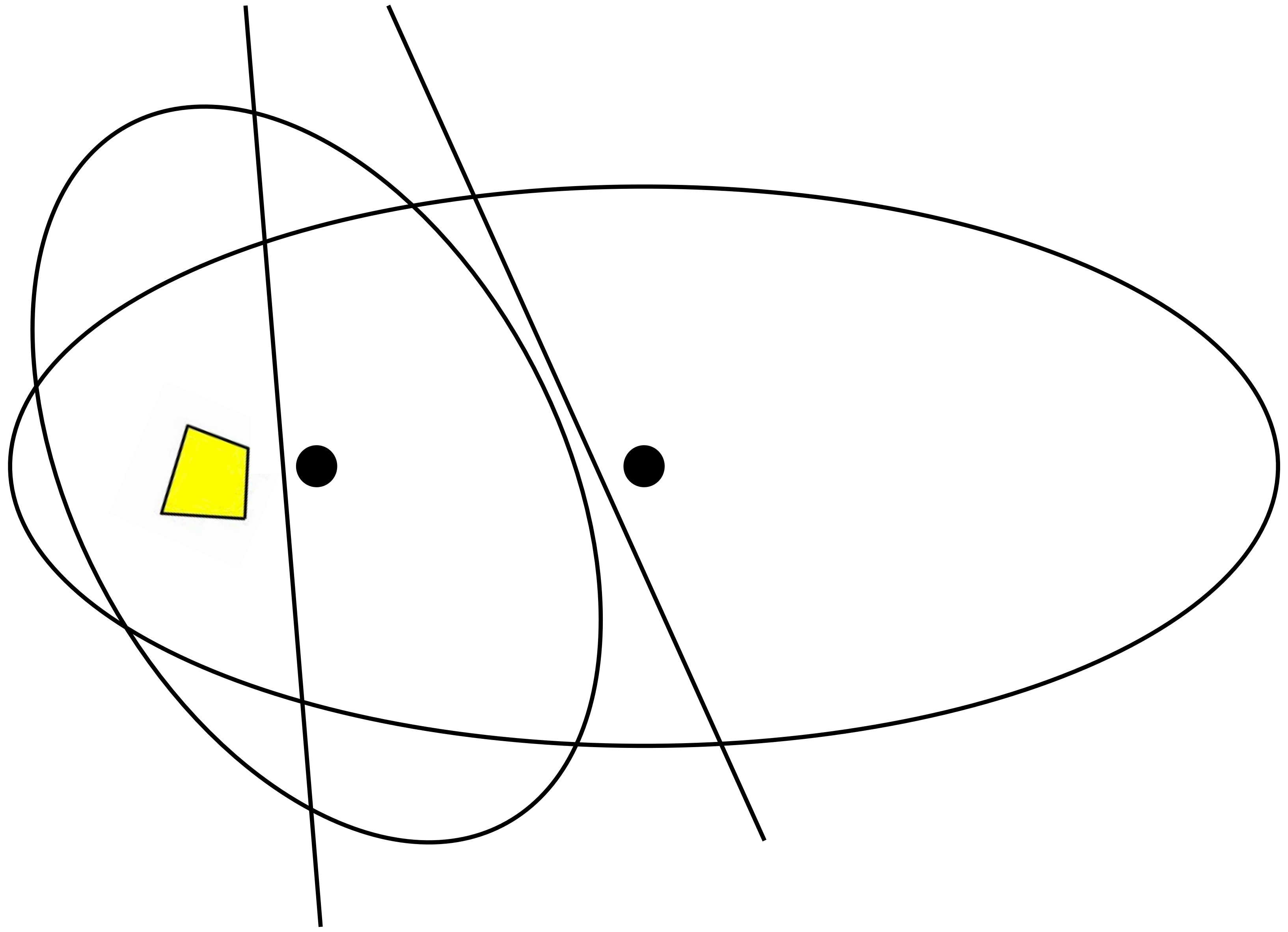
- keep the solution inside the polytope
- design penalty function so that the solution is not too close to the boundary
- the final solution will be arbitrarily close to the optimum solution



Ellipsoid Method

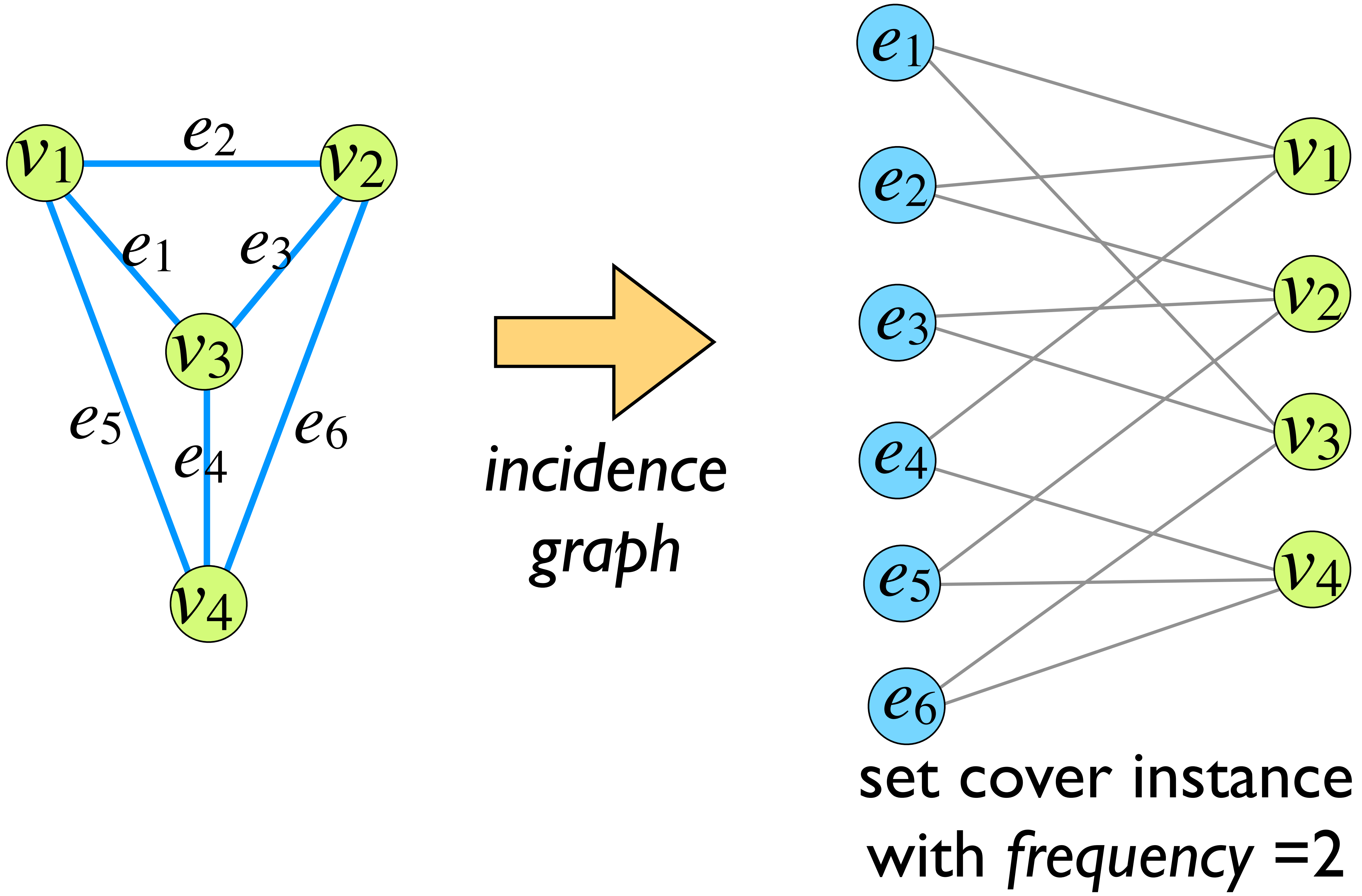
Ellipsoid Method (Khachiyan 1979)**:**

- maintain an ellipsoid that contains the feasible region
- cut the ellipsoid in half, find smaller ellipsoid to enclose the half-ellipsoid, and repeat



Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.



Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- **Integer Linear Program (ILP)** for vertex cover:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} x_v && \text{linear objective function} \\
\text{subject to} \quad & \sum_{v \in e} x_v \geq 1, \quad && e \in E \qquad \text{linear constraints} \\
& x_v \in \{0,1\}, && v \in V \qquad \text{integer domains}
\end{aligned}
$$

- Solving integer linear program is **NP**-hard.

Vertex Cover

Instance: An undirected graph $G(V, E)$.
Find the smallest $C \subseteq V$ that intersects all edges.

- Linear Program (LP) *relaxation*:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} x_v \\
\text{subject to} \quad & \sum_{v \in e} x_v \geq 1, && e \in E \\
& x_v \in [0,1], && v \in V
\end{aligned}
$$

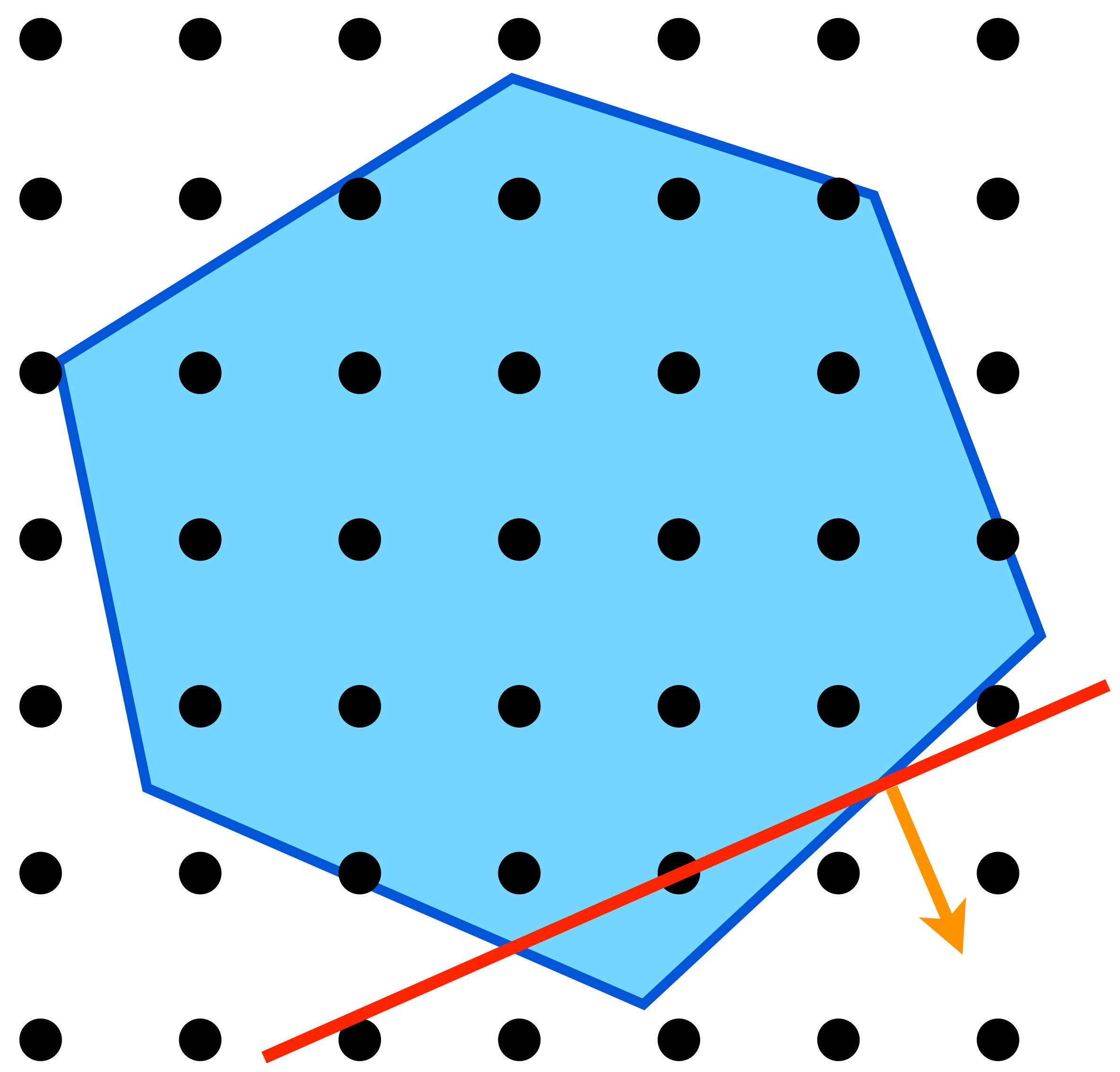
fractional domains

- linear programs are solvable in polynomial time!

Integrality

$$\begin{aligned} \text{min} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \cancel{\boldsymbol{x} \in \mathbb{Z}^n} \end{aligned}$$

LP-relaxation



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

$$\text{s.t.} \sum_{v\in e} x_v \geq 1, \qquad e \in E$$

$$x_v \in [0,1], \qquad v \in V$$

LP Relax & Round:

find **OPT** $\boldsymbol{x}^* \in [0,1]^V$;

round $\boldsymbol{x}^*$ to **feasible *integral*** $\hat{\boldsymbol{x}}$:

$$\hat{x}_v = \left.\begin{cases} 1 & \text{if } x_v^* \geq 0.5 \\ 0 & \text{otherwise} \end{cases}\right\} \leq 2x_v^*$$

- **Soundness** of rounded solution $\hat{\boldsymbol{x}}$ (as a **vertex cover**):

$$\sum_{v\in e} x_v^* \geq 1 \implies \sum_{v\in e} \hat{x}_v \geq 1$$

- **Approximation ratio:**

$$OPT = OPT_{Int} \geq OPT_{LP} = \sum_{v\in V} x_v^*$$

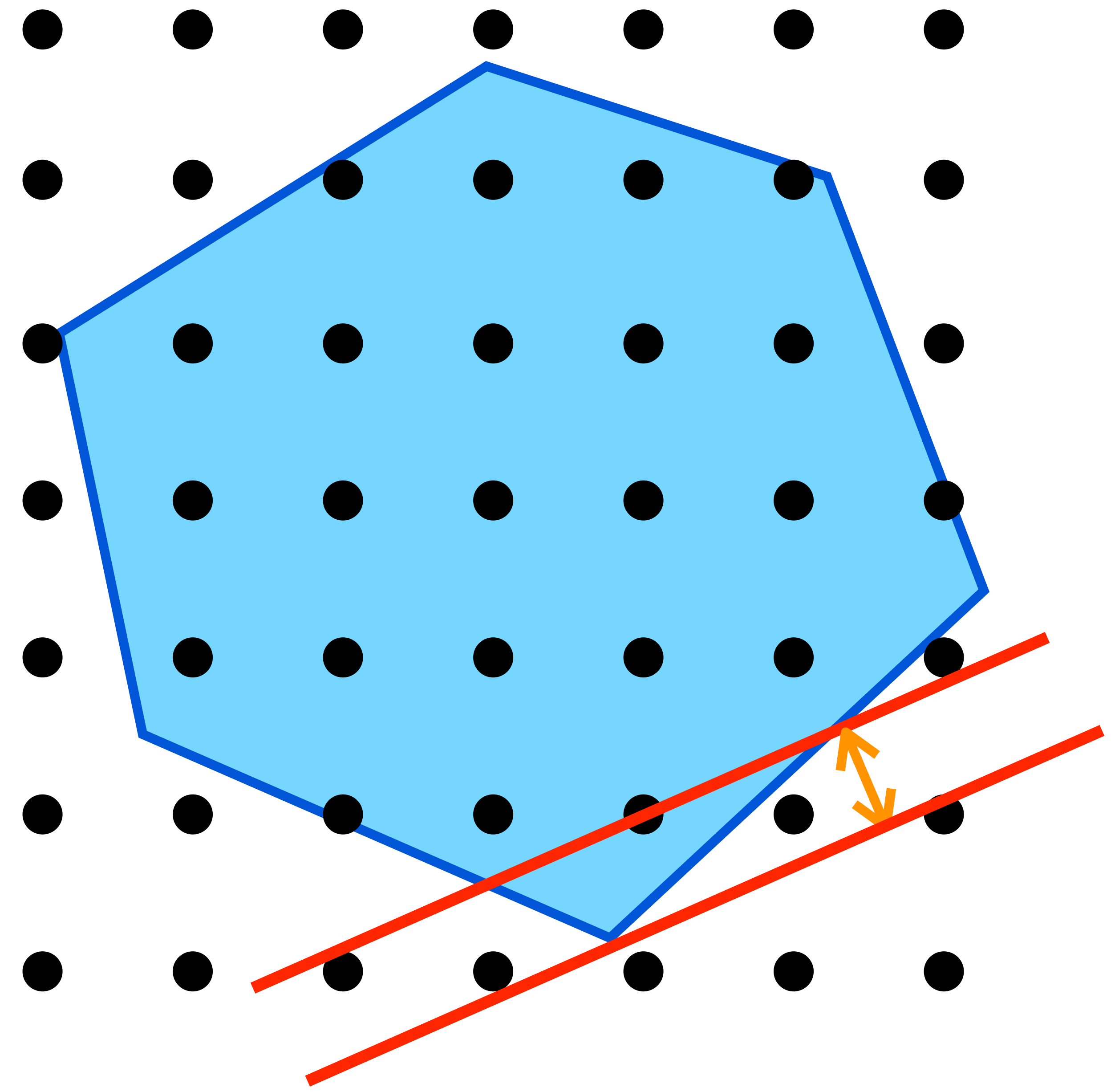
$$SOL = \sum_{v\in V} \hat{x}_v \leq 2\sum_{v\in V} x_v^* \leq 2OPT$$

LP Relaxation & Rounding

- Modeling: Express the optimization problem as an **Integer Linear Program (ILP)**.
- Relaxation: Relax the **ILP** to a **Linear Program** (**LP**).
- Solving: Find the ***optimal*** solution by an efficient LP solver.
- Rounding: Round the optimal solution to a ***feasible integral*** solution.
- Analysis: Prove that the rounded solution is not too far away from the ***optimal integral*** solution (usually by **comparing with the optimal solution**).

Integrality Gap

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \cancel{\boldsymbol{x} \in \mathbb{Z}^n} \end{aligned}$$



$$\text{integrality gap} = \sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_{\mathrm{LP}}(I)}$$

Integrality Gap

- minimum vertex cover of $G(V,E)$:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} x_v \\
\text{subject to} \quad & \sum_{v \in e} x_v \geq 1, && e \in E \\
& x_v \in \{0,1\}, && v \in V
\end{aligned}
$$

$$\text{integrality gap} = \sup_I \frac{\mathrm{OPT}(I)}{\mathrm{OPT}_{\mathrm{LP}}(I)}$$

- For LP relaxation of **vertex cover**: integrality gap $= 2$
- [Singh '19] int. gap on G = $\left(2 - \dfrac{2}{\chi^f(G)}\right)$ fractional chromatic number

MAX-SAT

$(x \text{ OR } y \text{ OR } z) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } z) \text{ AND}$

$(x \text{ OR } y \text{ OR } \bar{z}) \text{ AND } (x \text{ OR } \bar{y} \text{ OR } \bar{z}) \text{ AND}$

$(\bar{x} \text{ OR } y \text{ OR } z) \text{ AND } (\bar{x} \text{ OR } \bar{y} \text{ OR } \bar{z})$

Max-SAT

Instance: A CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

Find an assignment $\boldsymbol{x} \in \{\mathtt{T}, \mathtt{F}\}^n$ that maximizes the number of satisfied clauses.

- **CNF (Conjunctive Normal Form)**: conjunction ($\wedge$) of *clauses*

$$\Phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$$

- **Boolean variables:** $x_1, x_2, \ldots, x_n \in \{\mathtt{T}, \mathtt{F}\}$
- **Clause:** disjunction ($\vee$) of *literals*
- **literal:** x_i or $\neg x_i$
- Max-SAT: **NP**-hard

Random Assignment

Instance: A CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

Find an assignment $\boldsymbol{x} \in \{\mathtt{T}, \mathtt{F}\}^n$ that maximizes the number of satisfied clauses.

Random assignment:

each x_i is assigned a value from $\{\mathtt{T}, \mathtt{F}\}$ uniformly and independently at random

- A clause $C_j = (\ell_1 \vee \cdots \vee \ell_k)$ of k_j literals:

$$\Pr[\, C_j \text{ is satisfied} \,] = 1 - 2^{-k_j} \geq \frac{1}{2}$$

$$\mathbb{E}\,[\text{\# of satisfied clauses}] = \sum_{j=1}^{m} \Pr[\, C_j \text{ is satisfied} \,] \geq \frac{m}{2} \geq \frac{1}{2} OPT$$

Integer Program

Instance: A CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

$$\Phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \wedge (x_3 \vee \neg x_4 \vee \neg x_5)$$

- **Boolean variables:** $x_1, \ldots, x_n \in \{0, 1\}$ $\quad x_i = 1 \iff x_i = \mathtt{T}$
- **Clauses:** $C_j = (\ell_1 \vee \cdots \vee \ell_k) = \bigvee_{i \in S_j^+} x_i \vee \bigvee_{i \in S_j^-} \neg x_i$
  - $S_j^+ = \left\{ i \mid x_i \text{ appears in } C_j \right\}$
  - $S_j^- = \left\{ i \mid \neg x_i \text{ appears in } C_j \right\}$

$$C_j \text{ is satisfied} \iff \sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq 1$$

Integer Program

Instance: Clauses $C_1, \cdots, C_m$, where for each clause C_j:

$$S_j^+ = \left\{ i \mid x_i \text{ appears in } C_j \right\},\ S_j^- = \left\{ i \mid \neg x_i \text{ appears in } C_j \right\}$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to

$$\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

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

$$y_j \in \{0,1\} \qquad 1 \leq j \leq m$$

LP Relaxation

Instance: Clauses $C_1, \cdots, C_m$, where for each clause C_j:

$$S_j^+ = \{ i \mid x_i \text{ appears in } C_j \}, \; S_j^- = \{ i \mid \neg x_i \text{ appears in } C_j \}$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to

$$\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

$$x_i \in [0,1] \qquad 1 \leq i \leq n$$

$$y_j \in [0,1] \qquad 1 \leq j \leq m$$

Linear Randomized Rounding

Instance: Clauses $C_1, \cdots, C_m$, where for each clause C_j:

$$S_j^+ = \{i \mid x_i \text{ appears in } C_j\}, \; S_j^- = \{i \mid \neg x_i \text{ appears in } C_j\}$$

maximize $\quad \displaystyle\sum_{j=1}^{m} y_j$

subject to $\quad \displaystyle\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$$x_i, y_j \in [0,1] \qquad \forall i, j$$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n, \; \boldsymbol{y}^* \in [0,1]^m$

Linear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } x_i^* \\ 0 & \text{with prob. } 1 - x_i^* \end{cases}$ (independently)

$$\text{maximize} \quad \sum_{j=1}^{m} y_j$$

$$\text{subject to} \quad \sum_{i \in S_j^+} x_i^* + \sum_{i \in S_j^-} (1 - x_i^*) \geq y_j^* \qquad 1 \leq j \leq m$$

$$x_i, y_j \in [0,1] \qquad \forall i, j$$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n,\ \boldsymbol{y}^* \in [0,1]^m$

Linear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } x_i^* \\ 0 & \text{with prob. } 1 - x_i^* \end{cases}$ (independently)

- ß $= \sum_{j=1}^{m} y_j^*$

AM-GM

$$= 1 - \prod_{i \in S_j^+} (1 - x_i^*) \prod_{i \in S_j^-} x_i^* \geq 1 - \left(1 - y_j^*/k_j\right)^{k_j}$$

(C_j has k_j literals) Jenssen's inequality $\geq \left[1 - \left(1 - 1/k_j\right)^{k_j}\right] y_j^*$

$$\geq (1 - 1/\mathrm{e})\, y_j^*$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to $\displaystyle\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$x_i, y_j \in [0,1] \qquad \forall i, j$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n,\ \boldsymbol{y}^* \in [0,1]^m$

Linear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } x_i^* \\ 0 & \text{with prob. } 1 - x_i^* \end{cases}$ (independently)

- $OPT = OPT_{Int} \leq OPT_{LP} \quad = \sum_{j=1}^{m} y_j^*$
- $\Pr[\ C_j$ is satisfied $] \quad = 1 - \displaystyle\prod_{i \in S_j^+} (1 - x_i^*) \prod_{i \in S_j^-} x_i^* \qquad \geq (1 - 1/\mathrm{e})\, y_j^*$

$$\mathbb{E}[\,SOL\,] = \sum_{j=1}^{m} \Pr[\,C_j \text{ is satisfied}\,] \geq (1 - 1/\mathrm{e}) \sum_{j=1}^{m} y_j^* \ \geq (1 - 1/\mathrm{e})OPT$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to $\displaystyle\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$$x_i, y_j \in [0,1] \qquad \forall i, j$$

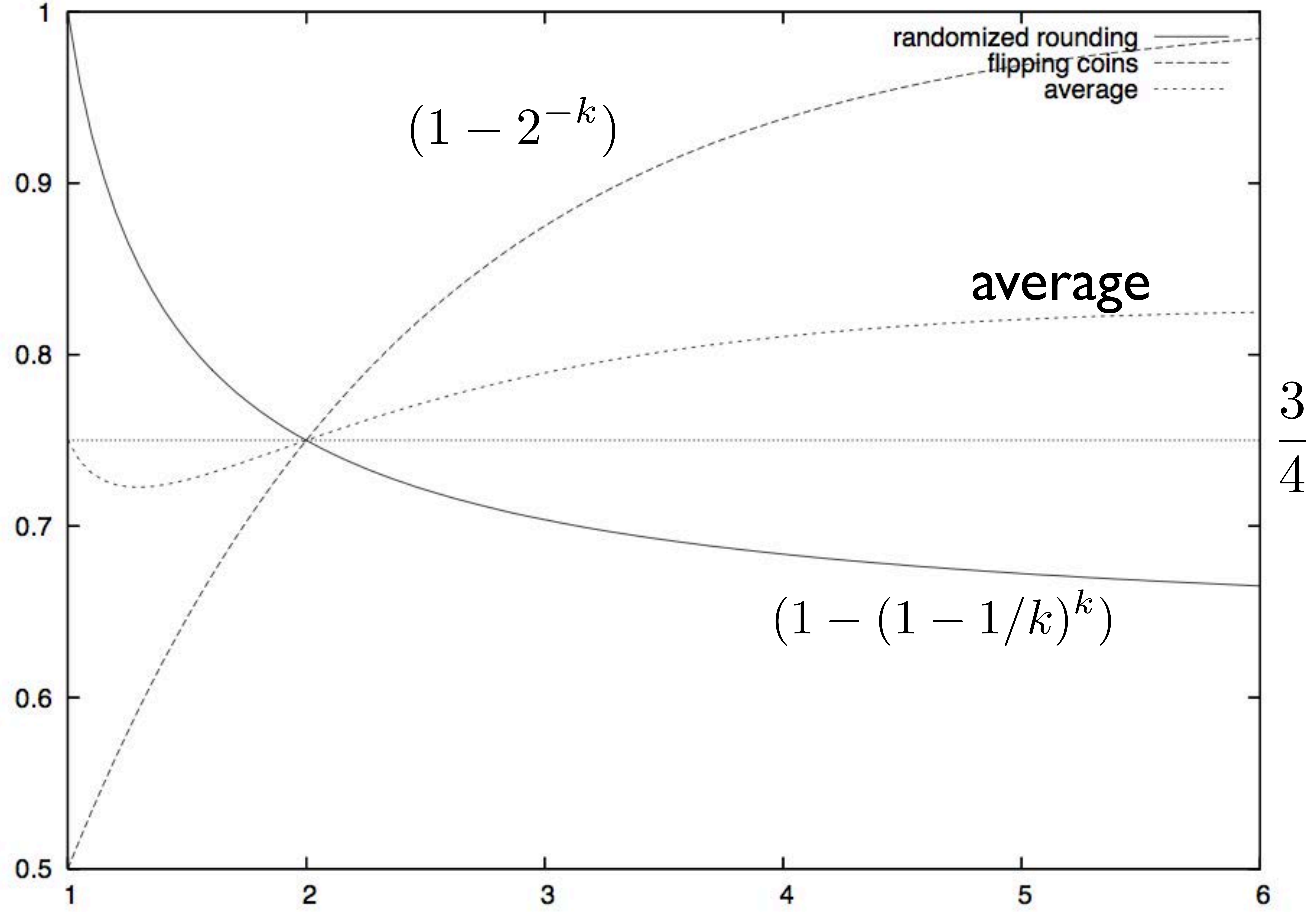
Optimal solution: $\boldsymbol{x}^* \in [0,1]^n,\ \boldsymbol{y}^* \in [0,1]^m$

Linear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } x_i^* \\ 0 & \text{with prob. } 1 - x_i^* \end{cases}$ (independently)

- $\Pr[\ C_j$ is satisfied $] \geq (1 - (1 - 1/k_j)^{k_j}) \cdot y_j^*$

Random assignment: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } 1/2 \\ 0 & \text{with prob. } 1/2 \end{cases}$ (independently)

- $\Pr[\ C_j$ is satisfied $] = 1 - 2^{-k_j}$ $\quad \geq (1 - 2^{-k_j}) \cdot y_j^* \geq (1 - 1/e) \cdot y_j^*$


randomized rounding
flipping coins
average
$(1-2^{-k})$
average
$\frac{3}{4}$
$(1-(1-1/k)^k)$
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6

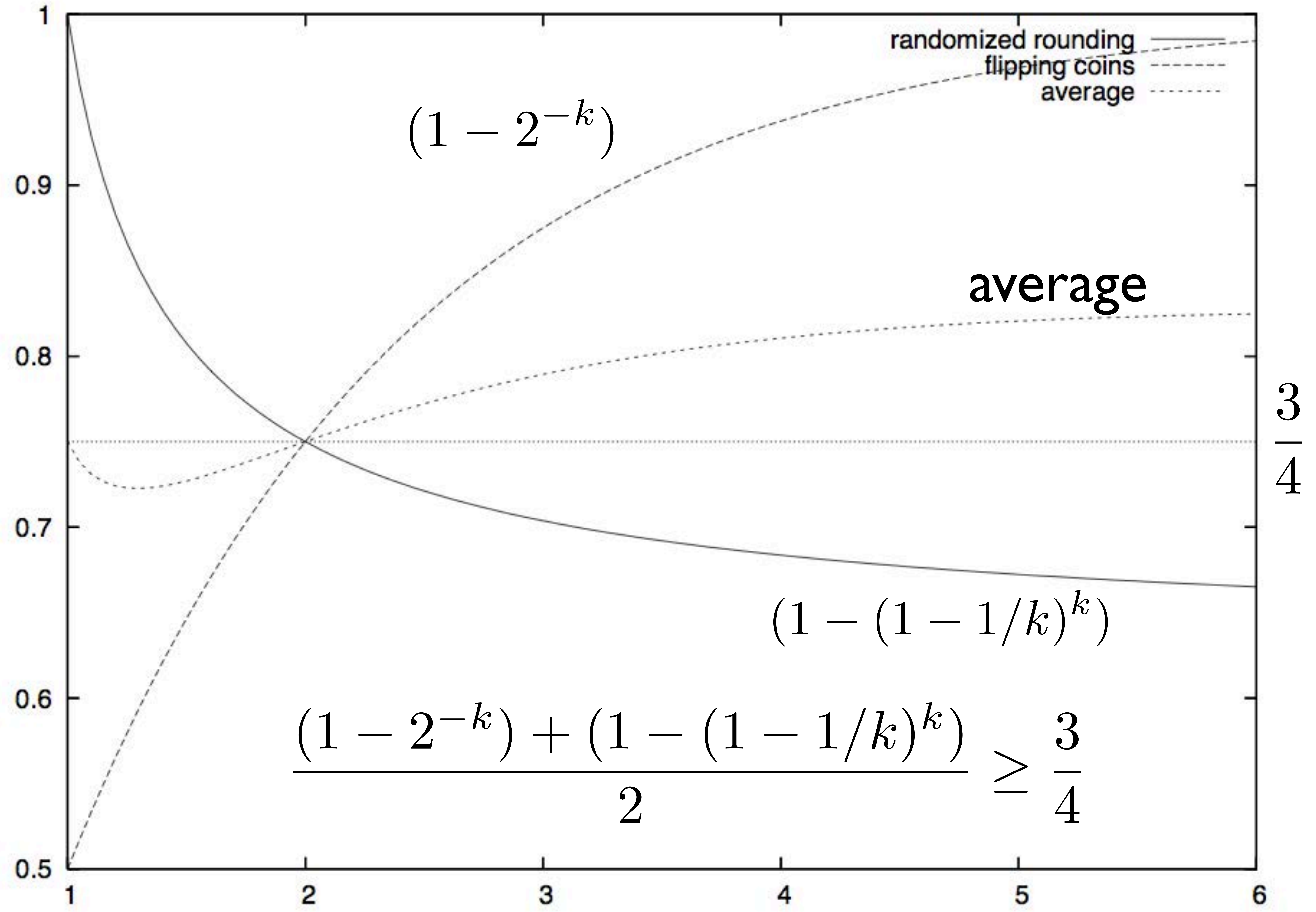
Power of Two Choices

- Use random assignment to satisfy M_1 clauses;
- Use linear rounding of LP relaxation to satisfy M_2 clauses;
- Return the solution with more satisfied clauses.

$$\mathbb{E}\left[\max\{M_1, M_2\}\right] \geq \mathbb{E}\left[\frac{M_1 + M_2}{2}\right]$$

$$\mathbb{E}\left[M_1\right] \geq \sum_{j=1}^{m} \left(1 - 2^{-k_j}\right) \cdot y_j^*$$

$$\mathbb{E}\left[M_2\right] \geq \sum_{j=1}^{m} \left(1 - (1 - 1/k_j)^{k_j}\right) \cdot y_j^*$$



randomized rounding
flipping coins
average
$(1-2^{-k})$
average
$\frac{3}{4}$
$(1-(1-1/k)^k)$
$\frac{(1-2^{-k})+(1-(1-1/k)^k)}{2} \geq \frac{3}{4}$
1
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6

Power of Two Choices

- Use random assignment to satisfy M_1 clauses;
- Use linear rounding of LP relaxation to satisfy M_2 clauses;
- Return the solution with more satisfied clauses.

$$\mathbb{E}\left[\max\{M_1, M_2\}\right] \geq \mathbb{E}\left[\frac{M_1 + M_2}{2}\right] \geq \frac{3}{4}\sum_{j=1}^{m} y_j^* \geq \frac{3}{4} OPT$$

$$\mathbb{E}\left[M_1\right] \geq \sum_{j=1}^{m}\left(1 - 2^{-k_j}\right) \cdot y_j^*$$

$$\mathbb{E}\left[M_2\right] \geq \sum_{j=1}^{m}\left(1 - (1 - 1/k_j)^{k_j}\right) \cdot y_j^*$$

Max-SAT

Instance: A CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

Find an assignment $\boldsymbol{x} \in \{\mathtt{T}, \mathtt{F}\}^n$ that maximizes the number of satisfied clauses.

- Use random assignment to satisfy M_1 clauses;
- Use linear rounding of LP relaxation to satisfy M_2 clauses;
- Return the solution with more satisfied clauses.

- This combined algorithm outputs a random solution that satisfies $\geq \frac{3}{4} OPT$ clauses in expectation.
- Can this be achieved by a single algorithm?

Non-Linear Rounding

Instance: Clauses $C_1, \cdots, C_m$, where for each clause C_j:

$$S_j^+ = \left\{ i \mid x_i \text{ appears in } C_j \right\},\ S_j^- = \left\{ i \mid \neg x_i \text{ appears in } C_j \right\}$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to $\displaystyle\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$$x_i, y_j \in [0,1] \qquad \forall i, j$$

Optimal fractional solution: $\boldsymbol{x}^* \in [0,1]^n$, $\boldsymbol{y}^* \in [0,1]^m$

Nonlinear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } f(x_i^*) \\ 0 & \text{with prob. } 1 - f(x_i^*) \end{cases}$ (independently)

maximize $\sum_{j=1}^{m} y_j$

subject to $\boxed{\sum_{i\in S_j^+} x_i^* + \sum_{i\in S_j^-} (1 - x_i^*) \geq y_j^*}$ $\quad 1 \leq j \leq m$

$x_i, y_j \in [0,1]$ $\quad \forall i, j$

Optimal fractional solution: $\boldsymbol{x}^* \in [0,1]^n, \;\; \boldsymbol{y}^* \in [0,1]^m$

Nonlinear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } f(x_i^*) \\ 0 & \text{with prob. } 1-f(x_i^*) \end{cases}$ (independently)

$$\Pr[\, C_j \text{ is unsatisfied} \,] = \prod_{i\in S_j^+} (1-f(x_i^*)) \prod_{i\in S_j^-} f(x_i^*)$$

$$\leq c^{-\left(\sum_{i\in S_j^+} x_i^* + \sum_{i\in S_j^-} (1 - x_i^*)\right)}$$

$$\leq c^{-y_j^*}$$

Suppose: for some $c > 1$

$1 - c^{-x} \leq f(x) \leq c^{x-1}$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to $\displaystyle\sum_{i\in S_j^+} x_i + \sum_{i\in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$x_i, y_j \in [0,1] \qquad \forall i, j$

Optimal fractional solution: $\boldsymbol{x}^* \in [0,1]^n, \;\; \boldsymbol{y}^* \in [0,1]^m$

Nonlinear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } f(x_i^*) \\ 0 & \text{with prob. } 1 - f(x_i^*) \end{cases}$ (independently)

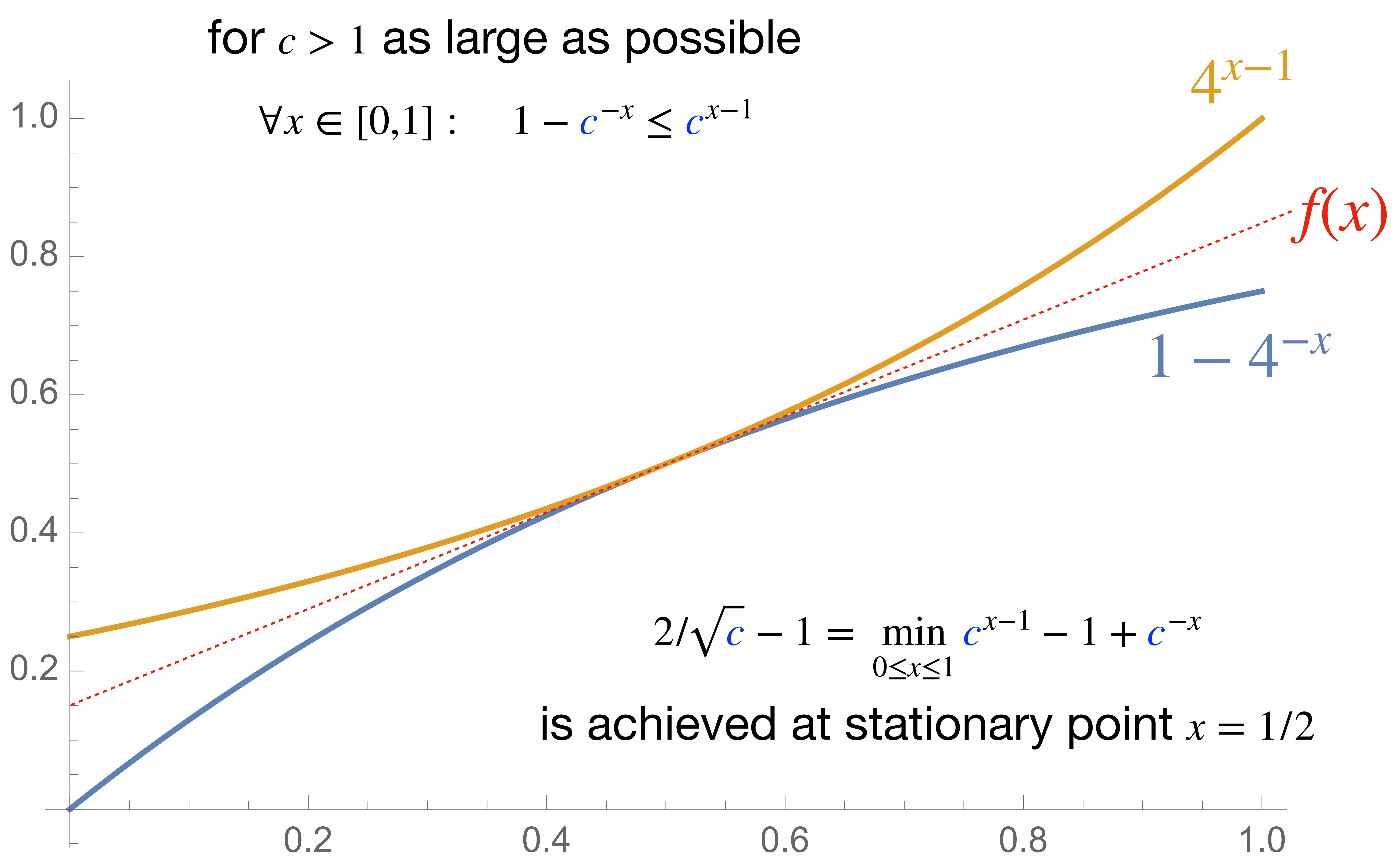
Suppose: for some $c > 1$

$1 - c^{-x} \leq f(x) \leq c^{x-1}$

$\Pr[\, C_j \text{ is unsatisfied} \,] \leq c^{-y_j^*}$

Jenssen's inequality

$$\mathbb{E}[\, SOL \,] \geq \sum_{j=1}^{m} \left(1 - c^{-y_j^*}\right) \geq (1 - 1/c) \sum_{j=1}^{m} y_j^* \geq (1 - 1/c) OPT$$


for $c > 1$ as large as possible
$\forall x \in [0,1]: \quad 1 - c^{-x} \le c^{x-1}$
4^{x-1}
$f(x)$
$1 - 4^{-x}$
$2/\sqrt{c} - 1 = \min_{0 \le x \le 1} c^{x-1} - 1 + c^{-x}$
is achieved at stationary point $x = 1/2$
1.0
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
1.0

maximize $\sum_{j=1}^{m} y_j$

subject to $\sum_{i \in S_j^{+}} x_i + \sum_{i \in S_j^{-}} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$x_i, y_j \in [0,1] \qquad \forall i, j$

Optimal fractional solution: $\boldsymbol{x}^* \in [0,1]^n, \ \boldsymbol{y}^* \in [0,1]^m$

Nonlinear rounding: $\hat{x}_i = \begin{cases} 1 & \text{with prob. } f(x_i^*) \\ 0 & \text{with prob. } 1 - f(x_i^*) \end{cases}$ (independently)

Suppose:

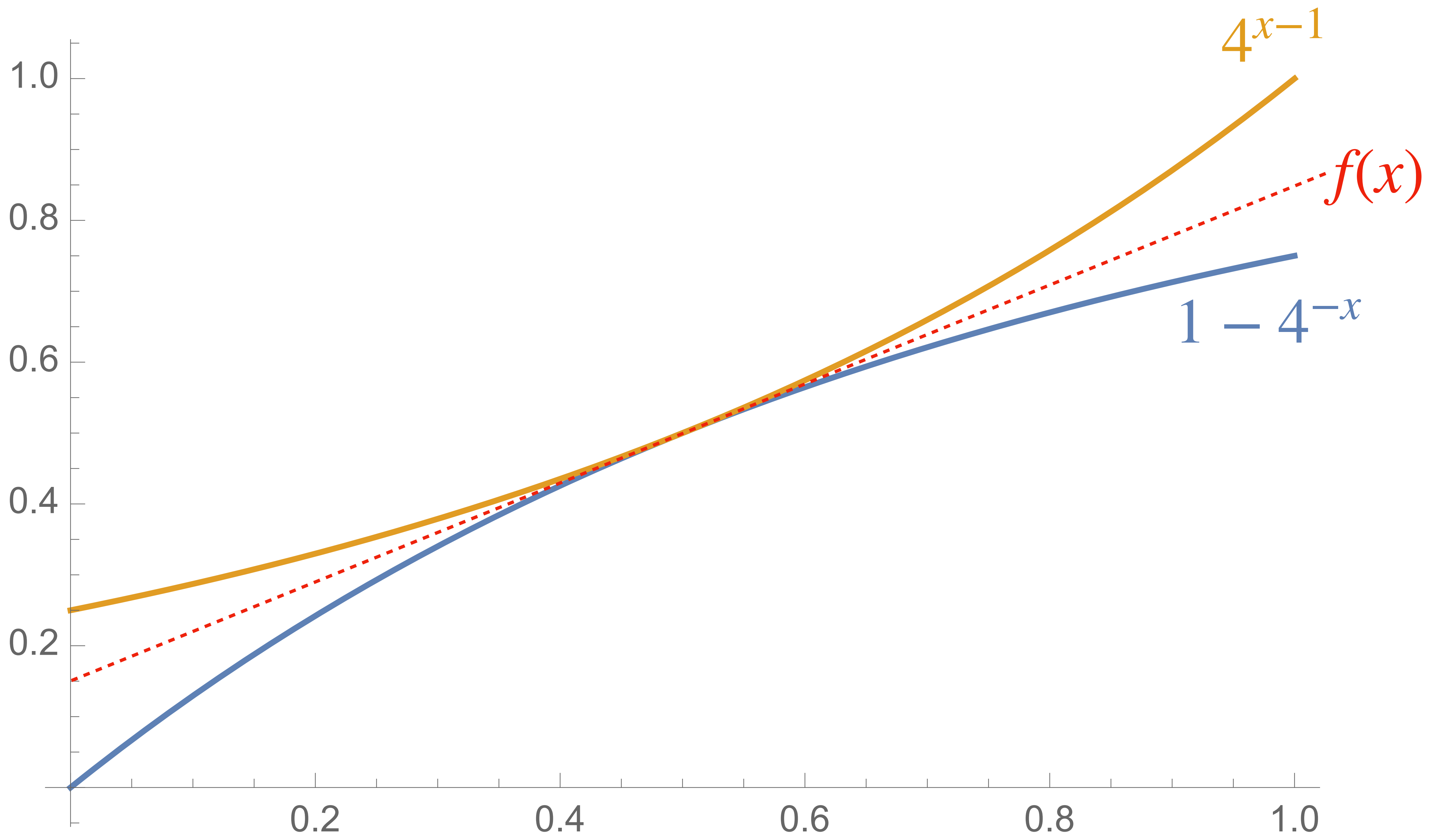
$1 - 4^{-x} \leq f(x) \leq 4^{x-1}$

$\Pr[\ C_j$ is unsatisfied $] \leq 4^{-y_j^*}$

Jenssen's inequality

$$\mathbb{E}[\ SOL\ ] \geq \sum_{j=1}^{m} \left(1 - 4^{-y_j^*}\right) \geq \frac{3}{4} \sum_{j=1}^{m} y_j^* \geq \frac{3}{4} OPT$$

Max-SAT



- Nonlinear rounding of LP-relaxation for Max-SAT satisfies $\geq \frac{3}{4}OPT$ clauses in expectation.
- Easy-to-compute ***explicit*** rounding $f(\cdot)$?

Power of Two Choices

- Use random assignment to satisfy M_1 clauses;
- Use linear rounding of LP relaxation to satisfy M_2 clauses;
- Return the solution with more satisfied clauses.

$$\mathbb{E}\left[\max\{M_1, M_2\}\right] \geq \boxed{\mathbb{E}\left[\frac{M_1 + M_2}{2}\right]} \geq \frac{3}{4}\sum_{j=1}^{m} y_j^* \geq \frac{3}{4} OPT$$

average of two
rounding schemes

$$\text{maximize} \quad \sum_{j=1}^{m} y_j$$

$$\text{subject to} \quad \sum_{i \in S_j^{+}} x_i + \sum_{i \in S_j^{-}} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$$

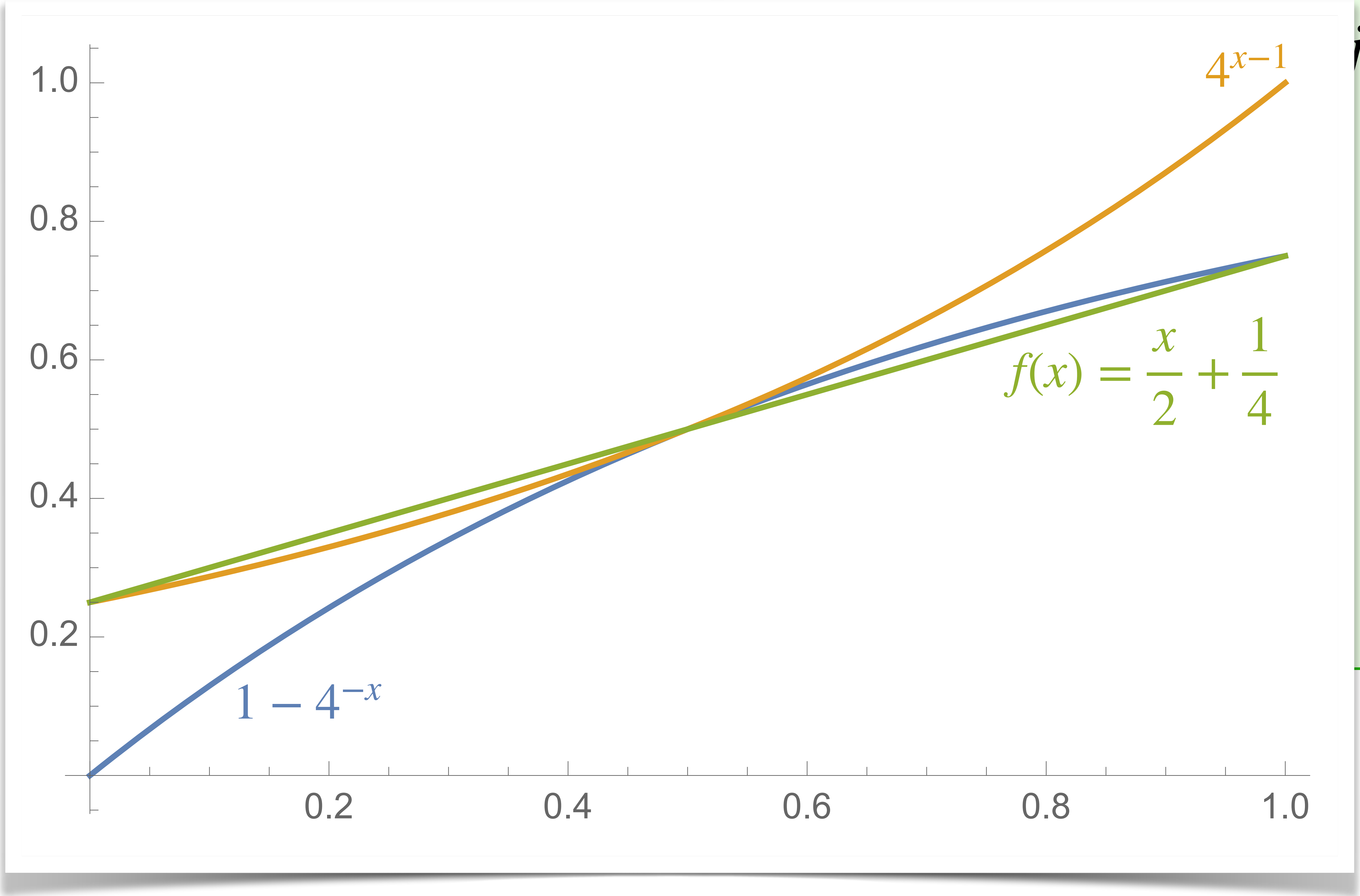
$$x_i, y_j \in [0,1] \qquad \forall i, j$$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n$, $\boldsymbol{y}^* \in [0,1]^m$

$$\hat{x}_i = \begin{cases} \begin{cases} 1 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases} & \text{w.p. } \frac{1}{2} \qquad \text{(random assign.)} \\ \begin{cases} 1 & \text{w.p. } x_i^* \\ 0 & \text{w.p. } 1 - x_i^* \end{cases} & \text{w.p. } \frac{1}{2} \qquad \text{(linear rounding)} \end{cases}$$

maximize $\sum_{j=1}^{m} y_j$

$j \leq m$



$$\text{maximize} \quad \sum_{j=1}^{m} y_j$$

$$\text{subject to} \quad \boxed{\sum_{i \in S_j^+} x_i^* + \sum_{i \in S_j^-} (1 - x_i^*) \geq y_j^*} \qquad 1 \leq j \leq m$$

$$x_i, y_j \in [0,1] \qquad \forall i, j$$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n, \ \boldsymbol{y}^* \in [0,1]^m$

$$\hat{x}_i = \begin{cases} 1 & \text{with prob. } \frac{1}{2}x_i^* + \frac{1}{4} = f(x_i^*) \\ 0 & \text{with prob. } -\frac{1}{2}x_i^* + \frac{3}{4} = 1 - f(x_i^*) \end{cases}$$

$$\Pr[\, C_j \text{ is unsatisfied} \,] = \prod_{i \in S_j^+} \left(-\frac{1}{2}x_i^* + \frac{3}{4} \right) \prod_{i \in S_j^-} \left(\frac{1}{2}x_i^* + \frac{1}{4} \right)$$

$$\overset{\text{AM-GM}}{\leq} \left(\frac{1}{k_j} \left(\sum_{i \in S_j^+} \left(-\frac{1}{2}x_i^* + \frac{3}{4} \right) + \sum_{i \in S_j^-} \left(\frac{1}{2}x_i^* + \frac{1}{4} \right) \right) \right)^{k_j} \leq \left(\frac{3}{4} - \frac{y_j^*}{2k_j} \right)^{k_j}$$

maximize $\displaystyle\sum_{j=1}^{m} y_j$

subject to $\displaystyle\sum_{i \in S_j^+} x_i + \sum_{i \in S_j^-} (1 - x_i) \geq y_j \qquad 1 \leq j \leq m$

$x_i, y_j \in [0,1] \qquad \forall i, j$

Optimal solution: $\boldsymbol{x}^* \in [0,1]^n$, $\boldsymbol{y}^* \in [0,1]^m$

$$\hat{x}_i = \begin{cases} 1 & \text{with prob. } \frac{1}{2}x_i^* + \frac{1}{4} = f(x_i^*) \\ 0 & \text{with prob. } -\frac{1}{2}x_i^* + \frac{3}{4} = 1 - f(x_i^*) \end{cases}$$

$$\Pr[\, C_j \text{ is satisfied }] \geq 1 - \left(\frac{3}{4} - \frac{y_j^*}{2k_j}\right)^{k_j} \geq \left(1 - \left(\frac{3}{4} - \frac{1}{2k_j}\right)^{k_j}\right) y_j^* \geq \frac{3}{4} y_j^*$$

Jenssen's inequality

$$\mathbb{E}[\, SOL \,] \geq \frac{3}{4} \sum_{j=1}^{m} y_j^* \geq \frac{3}{4} OPT$$

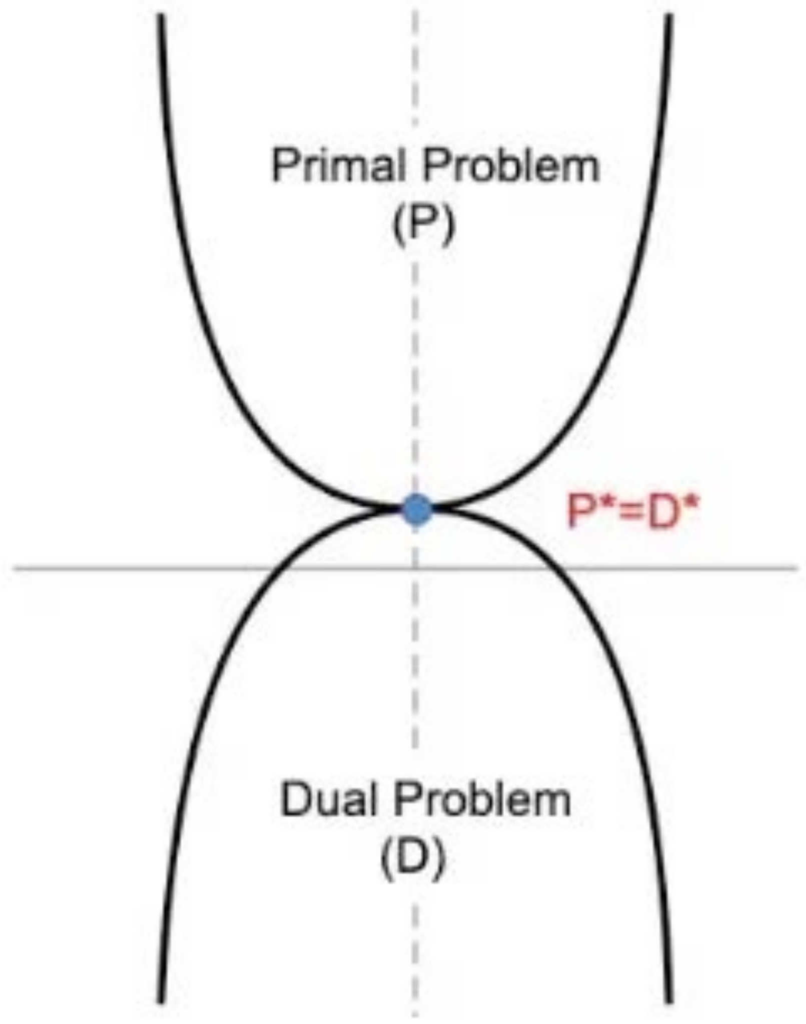
Max-SAT

Instance: A CNF formula $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$.

Find an assignment $x \in \{\mathtt{T}, \mathtt{F}\}^n$ that maximizes the number of satisfied clauses.

- Randomized rounding of LP-relaxation for Max-SAT satisfies $\geq \frac{3}{4} OPT$ clauses in expectation.
- It can be *de-randomized* via **conditional expectation**.
- The **integrality gap** of the LP-relaxation for Max-SAT is 3/4.
- Max-3SAT has a 7/8-approximation algorithm by rounding semidefinite programming (**SDP**) relaxation, and cannot have >7/8-approximation ratio in poly-time unless **NP**=**P**.

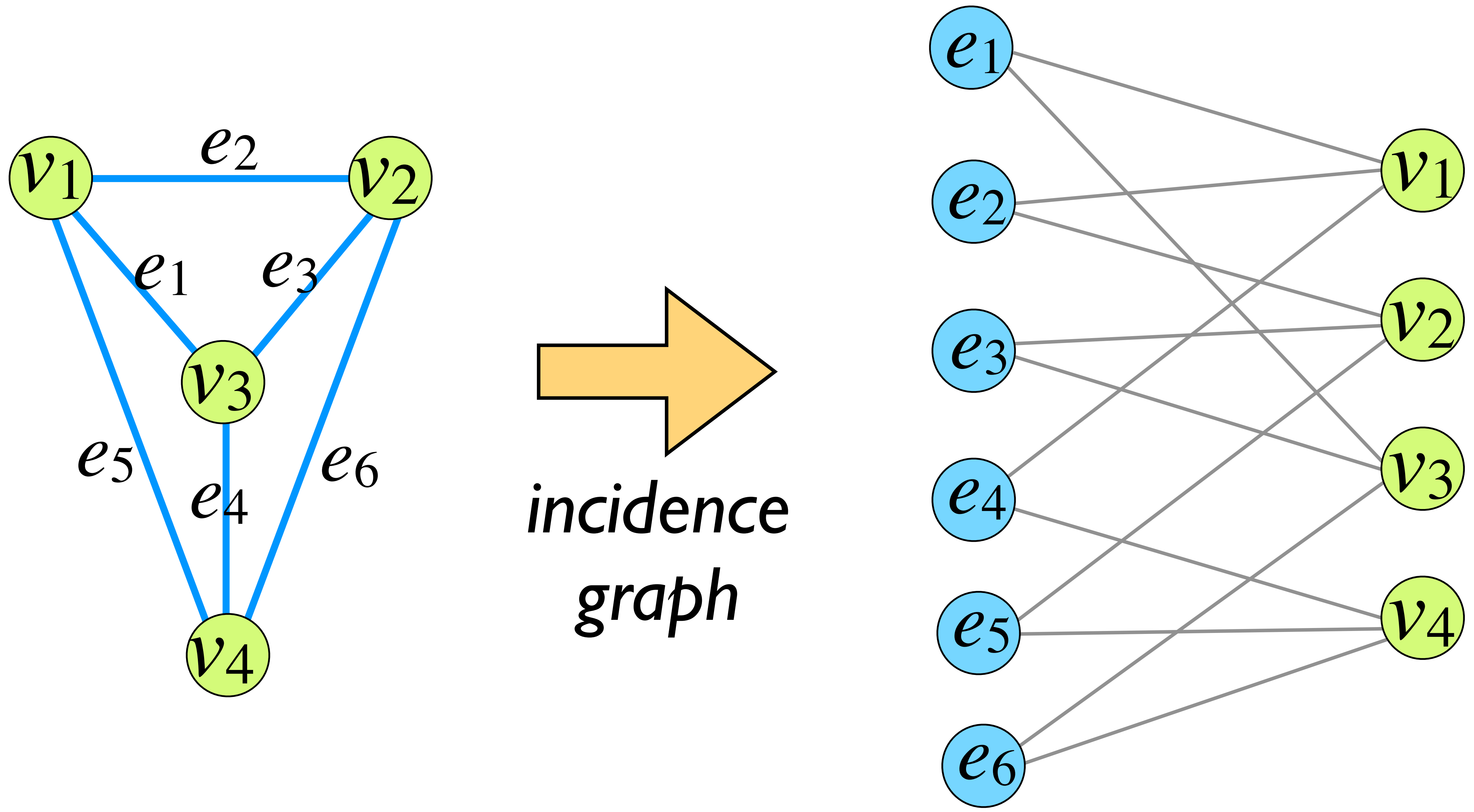
Duality



Vertex Cover

Instance: An undirected graph $G(V,E)$

Find the smallest $C \subseteq V$ that every edge has at least one endpoint in C.



instance of set cover with *frequency* =2

Instance: An undirected graph $G(V,E)$

Find the smallest $C \subseteq V$ that every edge has at least one endpoint in C.

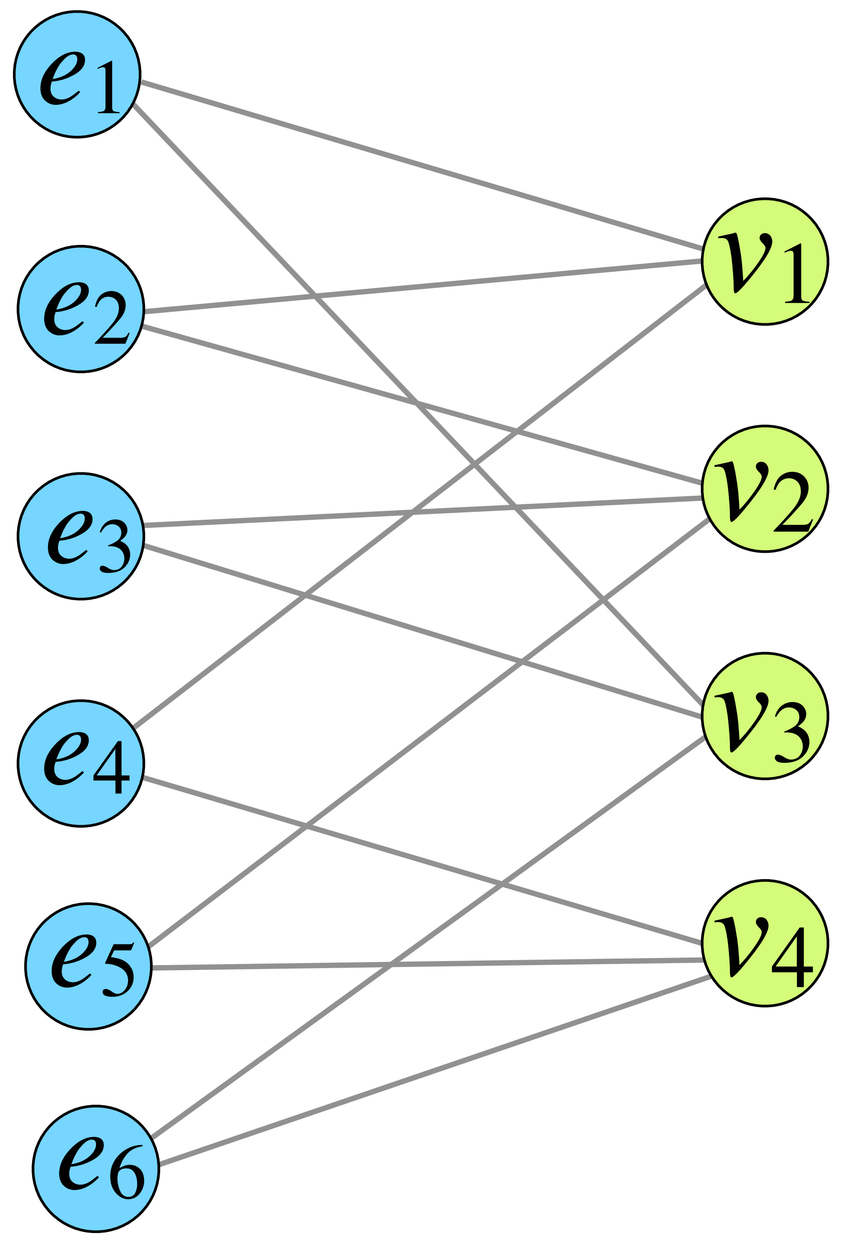
Find a *maximal matching* M;
return the set $C = \{v : (u,v) \in M\}$ of *matched* vertices;

maximality $\Rightarrow$ C is vertex cover

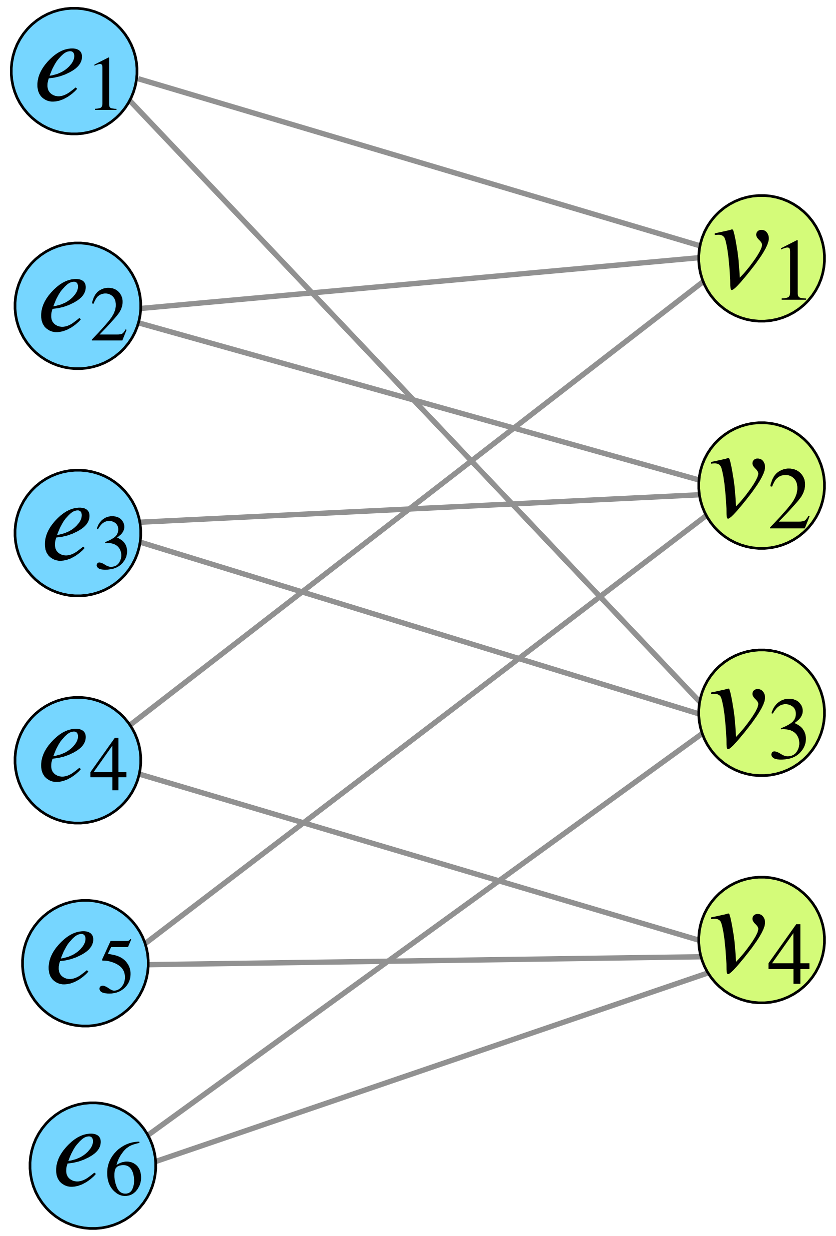
matching $\Rightarrow$ $|M| \leq \mathrm{OPT}_{\mathrm{VC}}$

(*weak duality*)

$|C| \leq 2|M| \leq 2\mathrm{OPT}$



Duality

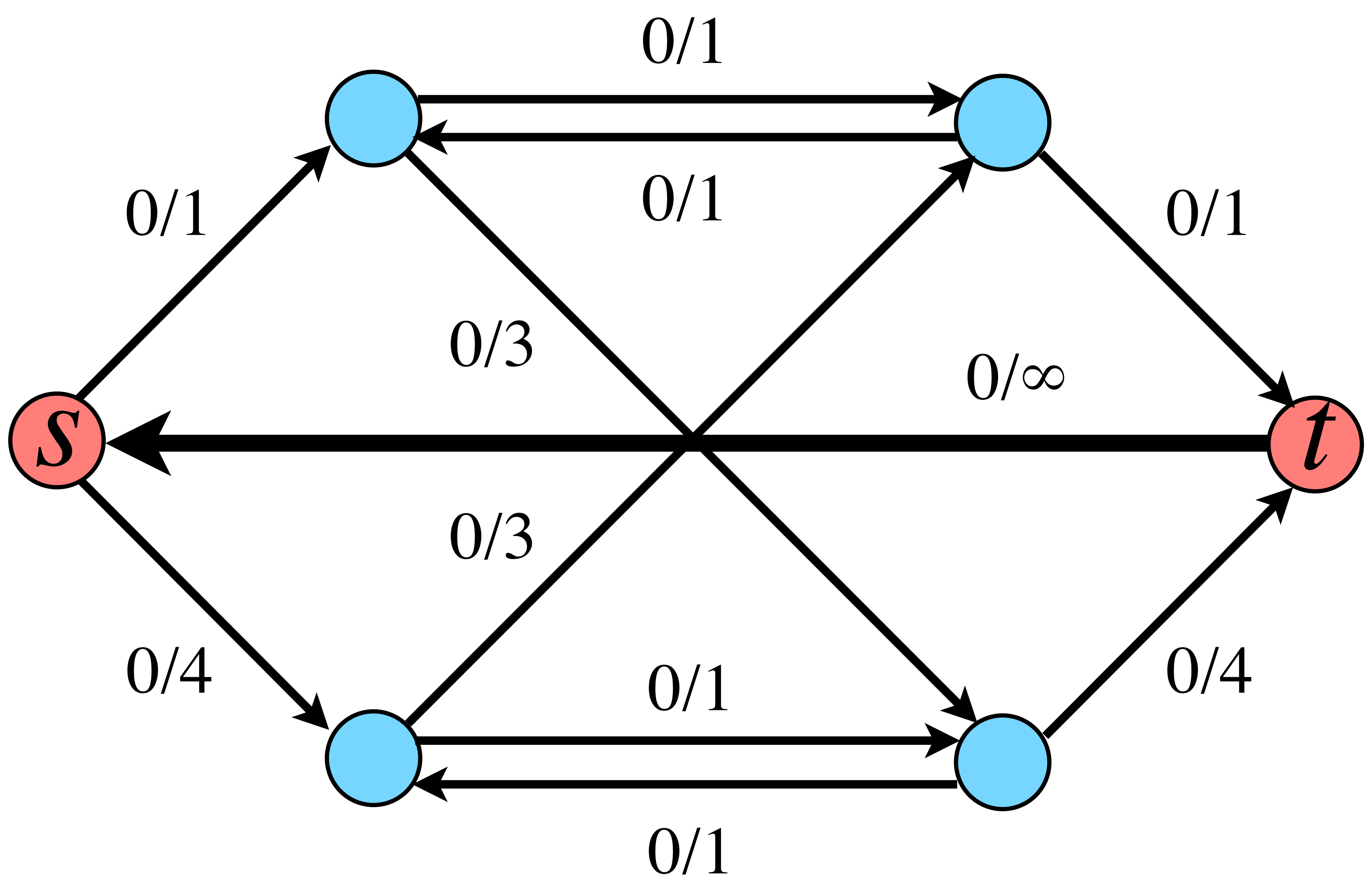


vertex cover:	constraints $\sum_{v \in e} x_v \geq 1$	variables $x_v \in \{0,1\}$
matching:	variables $y_e \in \{0,1\}$	constraints $\sum_{e \ni v} y_e \leq 1$

Max-Flow

digraph: $G = (V, E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



max f_{ts}

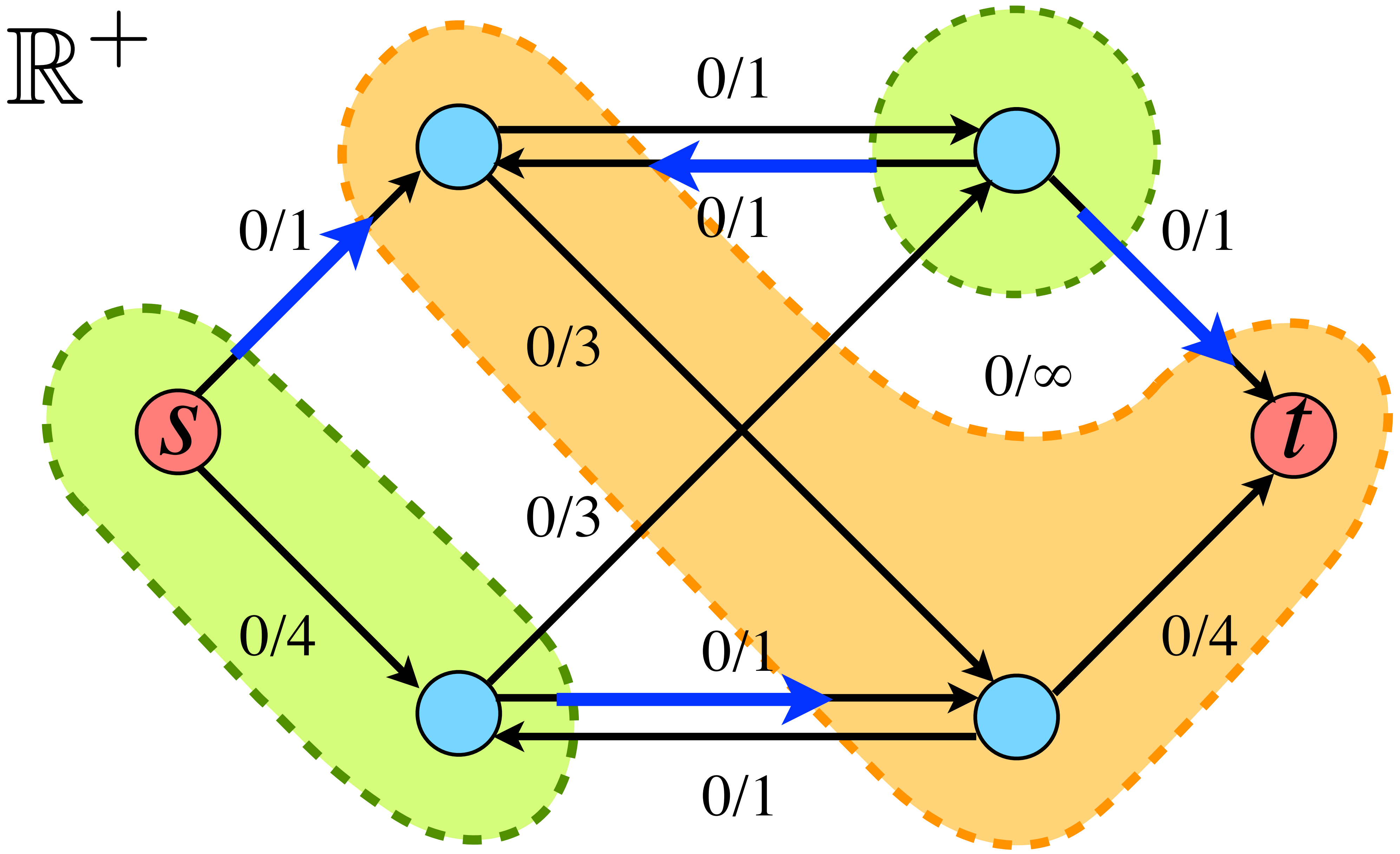
d_{uv} s.t. $0 \le f_{uv} \le c_{uv}$ $\forall (u,v) \in E$

p_u $\displaystyle\sum_{w:(w,u)\in E} f_{wu} - \sum_{v:(u,v)\in E} f_{uv} \ge 0$ $\forall u \in V$

Dual-LP

digraph: $G = (V, E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



min $\displaystyle\sum_{(u,v)\in E} c_{uv} d_{uv}$

s.t. $d_{uv} - p_u + p_v \geq 0 \qquad \forall (u,v) \in E$

$p_s - p_t \geq 1$

$\boldsymbol{d_{uv}}, p_u \geq 0 \quad d_{uv} \geq 0, p_u \in \{0, 1\} \qquad \forall (u,v) \in E \quad \forall u \in V$

Duality

Instance: graph $G(V,E)$

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

subject to $\sum_{v \in e} x_v \geq 1, \quad \forall e \in E$

$x_v \in \{0,1\}, \quad \forall v \in V$

vertex covers

dual: maximize $\sum_{e \in E} y_e$

subject to $\sum_{e \ni v} y_v \leq 1, \quad \forall v \in V$

$y_e \in \{0,1\}, \quad \forall e \in E$

matchings

Duality for LP-Relaxation

Instance: graph $G(V,E)$

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

subject to $\sum_{v \in e} x_v \geq 1, \quad \forall e \in E$

$x_v \geq 0, \quad \forall v \in V$

dual: maximize $\sum_{e \in E} y_e$

subject to $\sum_{e \ni v} y_v \leq 1, \quad \forall v \in V$

$y_e \geq 0, \quad \forall e \in E$

Estimate the Optima

$$
\begin{array}{lrcrcrcr}
\text{minimize} & \boxed{7x_1 & + & x_2 & + & 5x_3} & & \\
& & & \text{VI} & & & & \\
\text{subject to} & \boxed{x_1 & - & x_2 & + & 3x_3} & \geq & 10 \\
& & & + & & & & + \\
& \boxed{5x_1 & + & 2x_2 & - & x_3} & \geq & 6 \\
& & & & & & & = \\
& & x_1,\ x_2,\ x_3 \geq 0 & & & & & 16
\end{array}
$$

$16 \leq \text{OPT} \leq 30$ any feasible solution

$$\boldsymbol{x} = (2, 1, 3)$$

Estimate the Optima

$$\begin{array}{lccccccc}
\text{minimize} & \boxed{7x_1 & + & x_2 & + & 5x_3} & & \\
 & & & \mathbin{\rotatebox{90}{$\geq$}} & & & & \\
\text{subject to} & \boxed{x_1 & - & x_2 & + & 3x_3} & \geq & 10\, y_1 \\
 & & & + & & & & + \\
 & \boxed{5x_1 & + & 2x_2 & - & x_3} & \geq & 6\, y_2
\end{array}$$

$$x_1,\, x_2,\, x_3 \geq 0$$

$$10y_1 + 6y_2 \ \leq \ \text{OPT}$$

for any

$$\begin{array}{rcrcl}
y_1 & + & 5y_2 & \leq & 7 \\
-y_1 & + & 2y_2 & \leq & 1 \\
3y_1 & - & y_2 & \leq & 5
\end{array} \qquad y_1,\, y_2 \geq 0$$

Primal-Dual

Primal

$$\begin{array}{lrcrcrcl} \text{min} & 7x_1 & + & x_2 & + & 5x_3 & & \\ \text{s.t.} & x_1 & - & x_2 & + & 3x_3 & \geq & 10 \\ & 5x_1 & + & 2x_2 & - & x_3 & \geq & 6 \\ & \multicolumn{7}{l}{x_1,\ x_2,\ x_3 \geq 0} \end{array}$$

Dual

$$\begin{array}{lrcrcl} \text{max} & \multicolumn{5}{l}{10y_1 + 6y_2} \\ \text{s.t.} & y_1 & + & 5y_2 & \leq & 7 \\ & -y_1 & + & 2y_2 & \leq & 1 \\ & 3y_1 & - & y_2 & \leq & 5 \\ & \multicolumn{5}{l}{y_1,\ y_2 \geq 0} \end{array}$$

∀dual feasible
≤primal OPT

LP ∈ **NP**∩co**NP**

Surviving Problem

  · · · · 

price	c_1	c_2	· · · · ·	c_n	healthy
vitamin 1	a_{11}	a_{12}	· · · · ·	a_{1n}	$\geq b_1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
vitamin m	a_{m1}	a_{m2}	· · · · ·	a_{mn}	$\geq b_m$
solution:	x_1	x_2	· · · · ·	x_n	

minimize the total price while keeping healthy

Surviving Problem

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

price	c_1	c_2	$\cdots\cdots$	c_n	healthy
vitamin 1	a_{11}	a_{12}	$\cdots\cdots$	a_{1n}	$\geq b_1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
vitamin m	a_{m1}	a_{m2}	$\cdots\cdots$	a_{mn}	$\geq b_m$
solution:	x_1	x_2	$\cdots\cdots$	x_n	

minimize the total price while keeping healthy

LP Duality

Primal:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} \geq \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

Dual:

$$\max \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$$
$$\text{s.t.} \quad \boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$$
$$\boldsymbol{y} \geq \boldsymbol{0}$$

dual solution:	price	c_1	c_2	$\cdots\cdots$	c_n	healthy
y_1	vitamin 1	a_{11}	a_{12}	$\cdots\cdots$	a_{1n}	b_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_m	vitamin m	a_{m1}	a_{m2}	$\cdots\cdots$	a_{mn}	b_m

m types of vitamin pills, competitive to n natural foods,

design a pricing system max the total price

LP Duality

Primal:

$$\min \ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} \geq \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0}$$

$\geq$

Dual:

$$\max \ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$$
$$\text{s.t.} \quad \boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$$
$$\boldsymbol{y} \geq \boldsymbol{0}$$

Monogamy: dual(dual(LP)) = LP

Weak Duality:

∀ feasible primal solution $\boldsymbol{x}$ and dual solution $\boldsymbol{y}$

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \leq \boldsymbol{y}^{\mathrm{T}}A\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

LP Duality

Primal:

$$\min \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$
$$\text{s.t.} \quad A\boldsymbol{x} \ge \boldsymbol{b}$$
$$\boldsymbol{x} \ge \boldsymbol{0}$$

$\ge$

Dual:

$$\max \quad \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$$
$$\text{s.t.} \quad \boldsymbol{y}^{\mathrm{T}}A \le \boldsymbol{c}^{\mathrm{T}}$$
$$\boldsymbol{y} \ge \boldsymbol{0}$$

Weak Duality Theorem:

$\forall$ feasible primal solution $\boldsymbol{x}$ and dual solution $\boldsymbol{y}$

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \le \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

LP Duality

Primal:

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

Dual:

$$\begin{aligned} \max \quad & \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}} \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

Strong Duality Theorem:

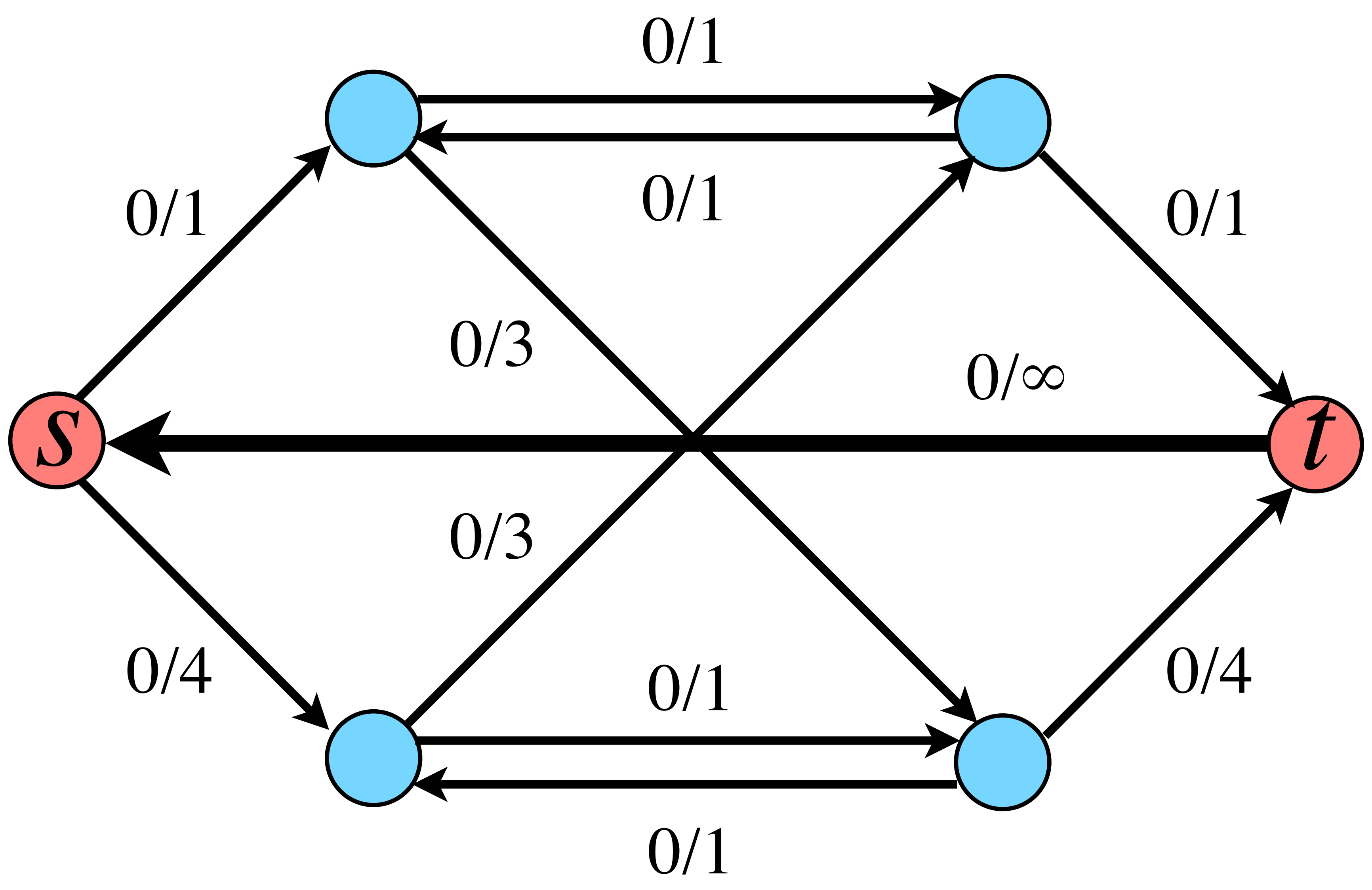
Primal LP has finite optimal solution $\boldsymbol{x}^*$
iff dual LP has finite optimal solution $\boldsymbol{y}^*$.

$$\boldsymbol{y}^{*\mathrm{T}}\boldsymbol{b} = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*$$

Max-Flow

digraph: $D(V,E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



max $\quad f_{ts}$

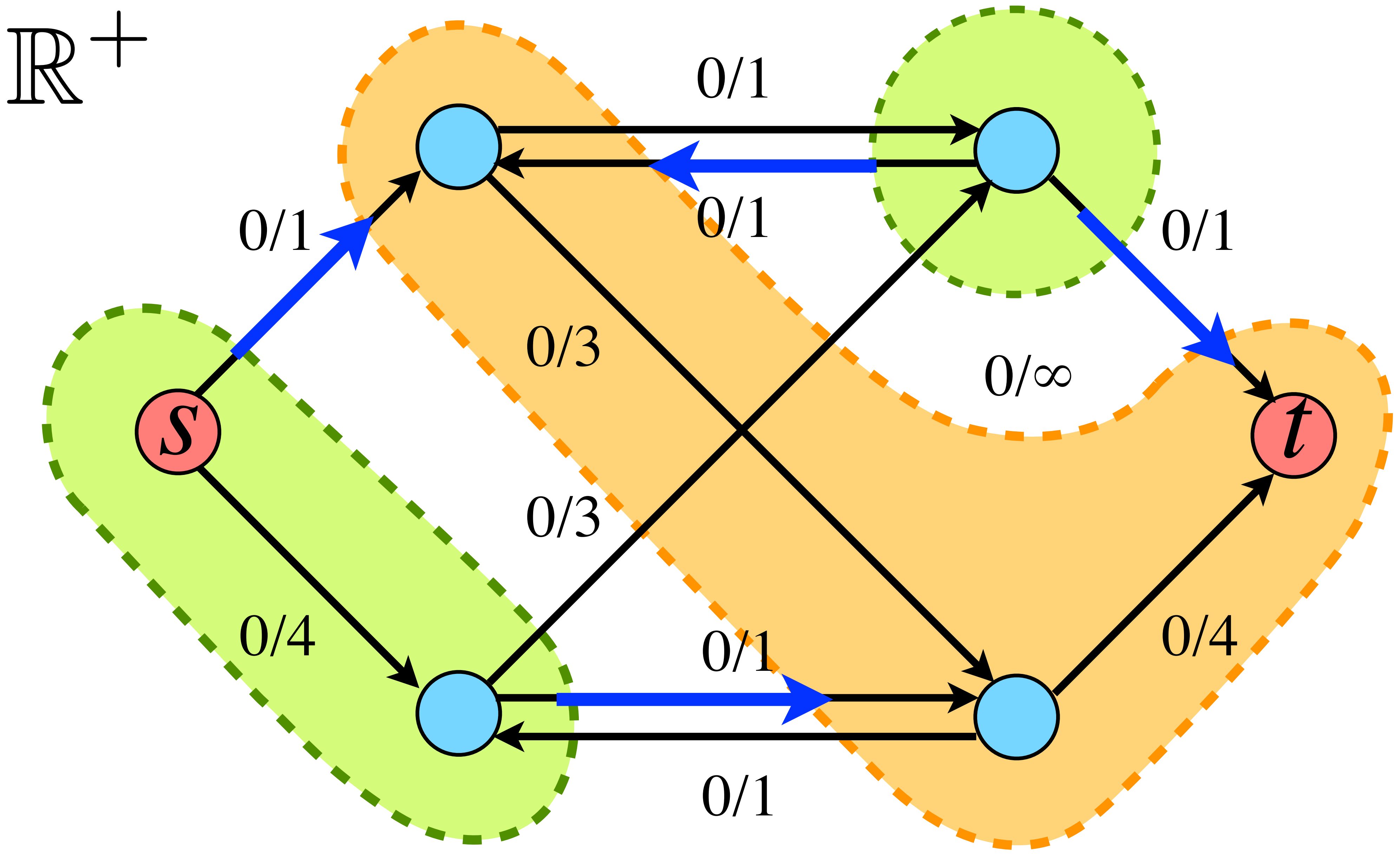
d_{uv} **s.t.** $0 \le f_{uv} \le c_{uv} \qquad \forall (u,v) \in E$

$p_u \qquad \displaystyle\sum_{w:(w,u)\in E} f_{wu} - \sum_{v:(u,v)\in E} f_{uv} \le 0 \qquad \forall u \in V$

Dual-LP

digraph: $D(V,E)$ source: s sink: t

capacity: $c : E \to \mathbb{R}^+$



min $\displaystyle\sum_{(u,v)\in E} c_{uv} d_{uv}$

s.t. $d_{uv} - p_u + p_v \geq 0 \quad \forall (u,v) \in E$

$p_s - p_t \geq 1$

$d_{uv},\ p_u \in \{0,1\} \quad \forall (u,v) \in E \quad \forall u \in V$

Primal: min $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$

s.t. $A\boldsymbol{x} \geq \boldsymbol{b}$

$\boldsymbol{x} \geq \boldsymbol{0}$

Dual: max $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$

s.t. $\boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$

$\boldsymbol{y} \geq \boldsymbol{0}$

$\forall$ feasible primal solution $\boldsymbol{x}$ and dual solution $\boldsymbol{y}$

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \leq \boldsymbol{y}^{\mathrm{T}}A\,\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

Strong Duality Theorem $\Rightarrow$ $\boldsymbol{x}$ and $\boldsymbol{y}$ are both optimal iff

$$\boxed{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} = \boldsymbol{y}^{\mathrm{T}}A\,\boldsymbol{x} = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}}$$

$\Updownarrow$

$$\sum_{i=1}^{m} b_i y_i = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right) y_i$$

$$\sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}y_i\right) x_j$$

$\Leftrightarrow$

$\forall i$: either $A_{i\cdot}\, x = b_i$ or $y_i = 0$

$\forall j$: either $y^{\mathrm{T}}A_{\cdot j} = c_j$ or $x_j = 0$

Complementary Slackness

Primal: min $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$
s.t. $A\boldsymbol{x} \geq \boldsymbol{b}$
$\boldsymbol{x} \geq \mathbf{0}$

Dual: max $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$
s.t. $\boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$
$\boldsymbol{y} \geq \mathbf{0}$

Complementary Slackness Conditions:

$\forall$ feasible primal solution $\boldsymbol{x}$ and dual solution $\boldsymbol{y}$

$\boldsymbol{x}$ and $\boldsymbol{y}$ are both optimal iff

$\forall i$: either $A_{i\cdot}\, x = b_i$ or $y_i = 0$

$\forall j$: either $y^{\mathrm{T}}A_{\cdot j} = c_j$ or $x_j = 0$

Relaxed Complementary Slackness

Primal: $\min\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ s.t. $A\boldsymbol{x} \geq \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$

Dual: $\max\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$ s.t. $\boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$, $\boldsymbol{y} \geq \boldsymbol{0}$

$\forall$ feasible primal solution $\boldsymbol{x}$ and dual solution $\boldsymbol{y}$

for $\alpha, \beta \geq 1$:

$\forall i$: either $A_{i\cdot}\, x \leq \alpha\, b_i$ or $y_i = 0$

$\forall j$: either $y^{\mathrm{T}}A_{\cdot j} \geq c_j/\beta$ or $x_j = 0$

$\Rightarrow$ $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \leq \alpha\beta\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \leq \alpha\beta\ \mathrm{OPT}_{\mathrm{LP}}$

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n}\left(\beta\sum_{i=1}^{m} a_{ij}y_i\right)x_j = \beta\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)y_i \leq \alpha\beta\sum_{i=1}^{m} b_i y_i$$

Primal-Dual Schema

Primal IP: min $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$

s.t. $A\boldsymbol{x} \geq \boldsymbol{b}$

$\boldsymbol{x} \in \mathbb{Z}_{\geq 0}$

Dual LP-relax: max $\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$

s.t. $\boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}}$

$\boldsymbol{y} \geq \boldsymbol{0}$

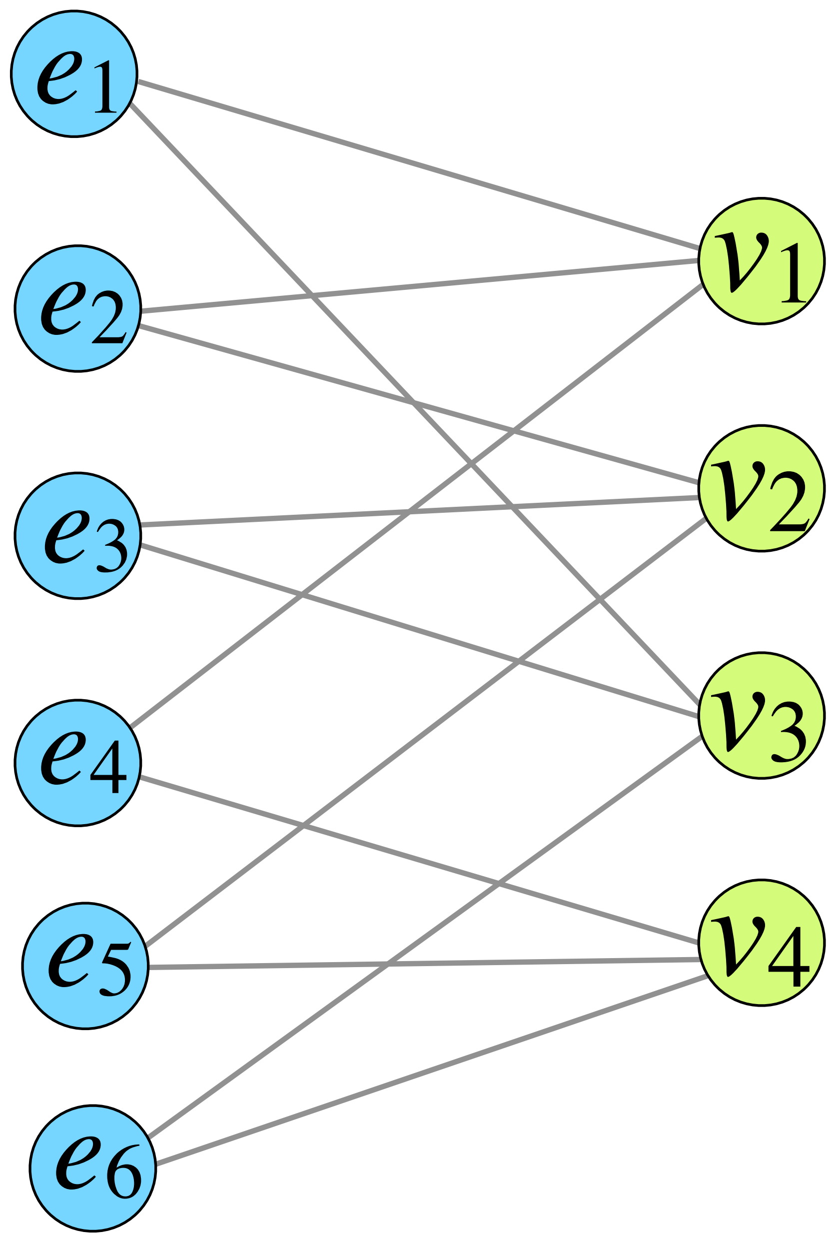
Find a primal *integral* solution $\boldsymbol{x}$ and a dual solution $\boldsymbol{y}$

for $\alpha, \beta \geq 1$:

$\forall i$: either $A_{i\cdot}\, x \leq \alpha\, b_i$ or $y_i = 0$

$\forall j$: either $y^{\mathrm{T}}A_{\cdot j} \geq c_j/\beta$ or $x_j = 0$

$$\Rightarrow \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \leq \alpha\beta\, \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \leq \alpha\beta\, \mathrm{OPT}_{\mathrm{LP}} \leq \alpha\beta\, \mathrm{OPT}_{\mathrm{IP}}$$



vertex cover:

constraints	variables
$\sum_{v\in e} x_v \geq 1$	$x_v \in \{0,1\}$

matching:

variables	constraints
$y_e \in \{0,1\}$	$\sum_{e\ni v} y_e \leq 1$

primal:

$$\begin{aligned} \min \quad & \sum_{v\in V} x_v \\ \text{s.t.} \quad & \sum_{v\in e} x_v \geq 1, \quad \forall e\in E \\ & x_v\in\{0,1\}, \quad \forall v\in V \end{aligned}$$

dual-relax:

$$\begin{aligned} \min \quad & \sum_{e\in E} y_e \\ \text{s.t.} \quad & \sum_{e\ni v} y_e \leq 1, \quad \forall v\in V \\ & y_e \geq 0, \quad \forall e\in E \end{aligned}$$

feasible (x, y) such that:

$\forall e$: either $\sum_{v\in e} x_v \leq \alpha$ or	$y_e = 0$
$\forall v$: either $\sum_{e\ni v} y_e = 1$ or	$x_v = 0$

primal:

$$\min \sum_{v\in V} x_v$$
$$\text{s.t.} \quad \sum_{v\in e} x_v \geq 1, \quad \forall e \in E$$
$$x_v \in \{0,1\}, \quad \forall v \in V$$

dual-relax:

$$\min \sum_{e\in E} y_e$$
$$\text{s.t.} \quad \sum_{e\ni v} y_e \leq 1, \quad \forall v \in V$$
$$y_e \geq 0, \quad \forall e \in E$$

event: "v is *tight* (*saturated*)" $\Leftrightarrow$ $\sum_{e\ni v} y_e = 1$

Initially $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$;
while $E \neq \emptyset$
pick an $e \in E$ and raise y_e ~~until some v goes tight;~~ to 1
set $x_v = 1$ for those ~~tight v~~ $v\in e$ and delete all $e\ni v$ from E;

every deleted e is incident to a v that $x_v = 1$
all edges are eventually deleted
$\Rightarrow$ $\forall e \in E$: $\sum_{v\in e} x_v \geq 1$
$\boldsymbol{x}$ is feasible

relaxed complementary slackness:

$\forall e$: either $\sum_{v\in e} x_v \leq 2$ or $y_e = 0$
$\forall v$: either $\sum_{e\ni v} y_e = 1$ or $x_v = 0$

$\Rightarrow$ $\sum_{v\in V} x_v \leq 2 \cdot OPT$

Initially $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$;
while $E \neq \varnothing$
  pick an $e \in E$ and raise y_e to 1 ~~until some v goes tight;~~
  set $x_v = 1$ for those ~~tight v~~ $v \in e$ and delete all $e \ni v$ from E;

Find a *maximal matching*;
return the set of *matched* vertices;

the returned set is a vertex cover

$$SOL \leq 2\ OPT$$

The Primal-Dual Schema

- Write down an LP-relaxation and its dual.

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{Z}_{\geq 0} \end{array} \qquad \begin{array}{ll} \max & \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \\ \text{s.t.} & \boldsymbol{y}^{\mathrm{T}}A \leq \boldsymbol{c}^{\mathrm{T}} \\ & \boldsymbol{y} \geq \boldsymbol{0} \end{array}$$

- Start with a primal *infeasible* solution $\boldsymbol{x}$ and a dual *feasible* solution $\boldsymbol{y}$ (usually $\boldsymbol{x}=\boldsymbol{0}, \boldsymbol{y}=\boldsymbol{0}$).
- Raise $\boldsymbol{x}$ and $\boldsymbol{y}$ until $\boldsymbol{x}$ is feasible:
  - raise $\boldsymbol{y}$ until some dual constraints gets *tight* $\boldsymbol{y}^{\mathrm{T}}A_{\cdot j} = c_j$;
  - raise x_j (integrally) corresponding to the *tight* dual constraints.
- Show the *complementary slackness conditions*:

$$\boxed{\begin{array}{l} \forall i\text{: either } A_{i\cdot}\,x \leq \alpha\, b_i \text{ or } y_i = 0 \\ \forall j\text{: either } y^{\mathrm{T}}A_{\cdot j} = c_j \text{ or } x_j = 0 \end{array}} \Rightarrow \begin{array}{l} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \leq \alpha\, \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \\ \quad \leq \alpha\, \mathrm{OPT} \end{array}$$