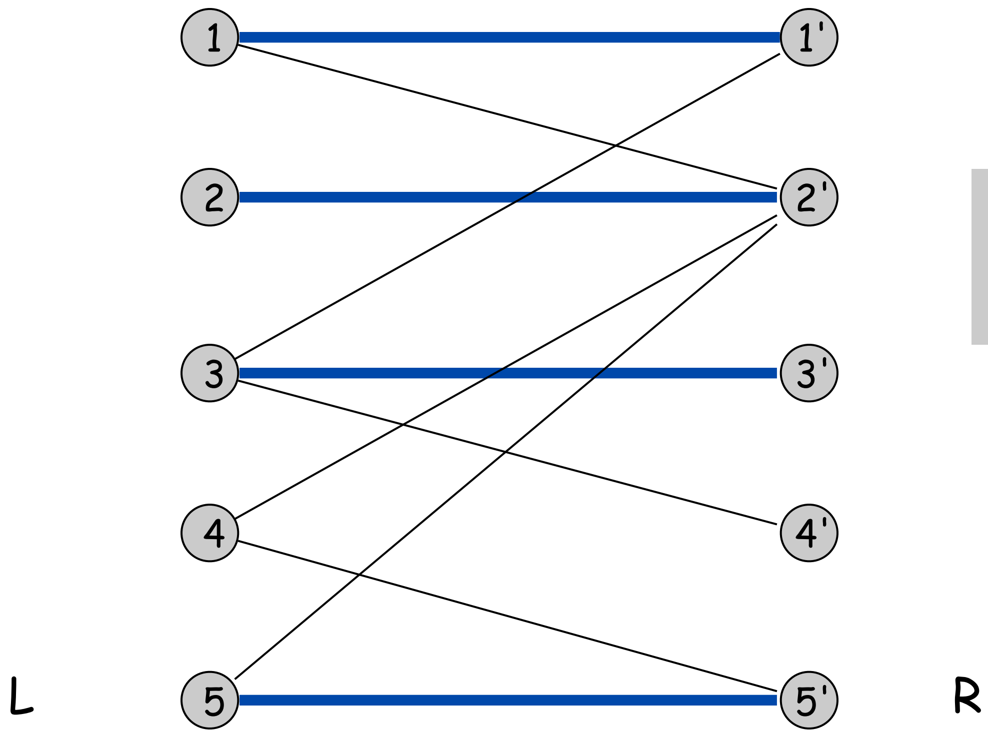
7.5 Bipartite Matching

Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in M.
- Max matching: find a max cardinality matching.



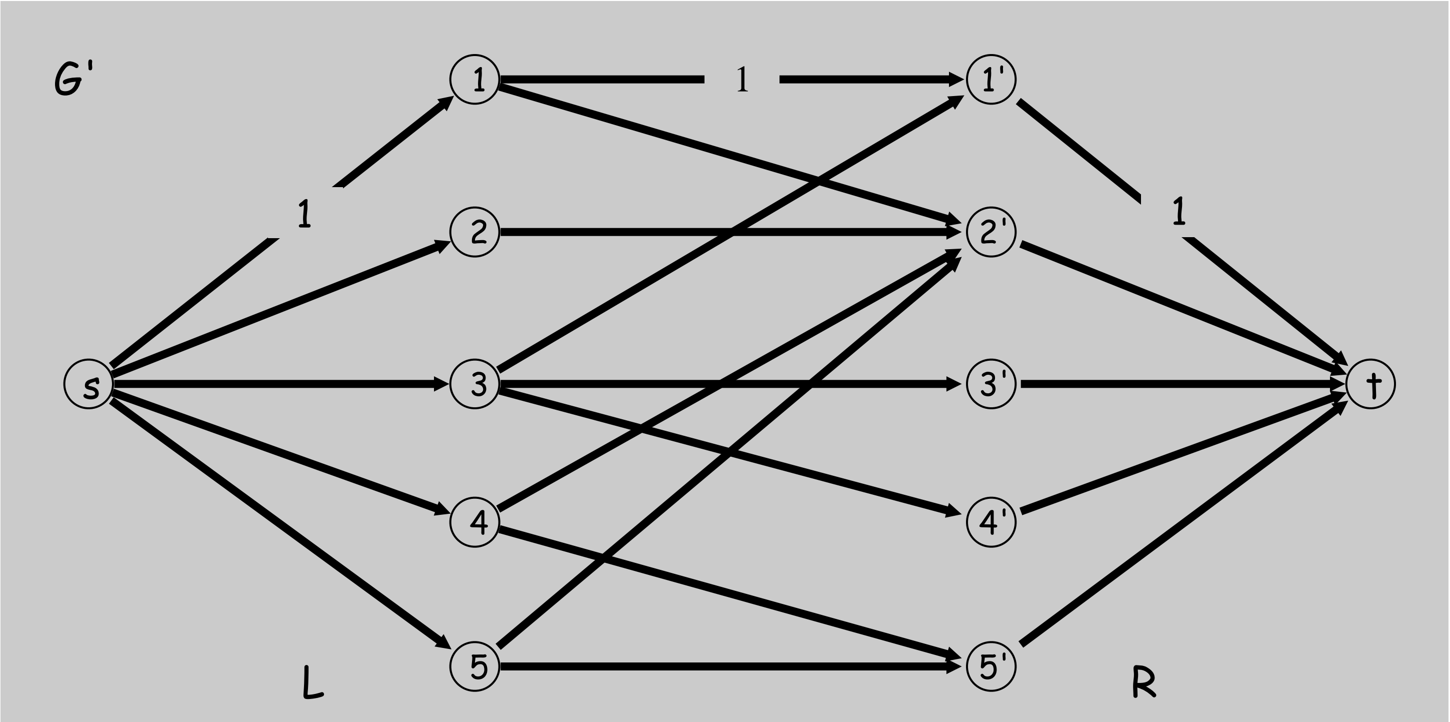
max matching

1-1', 2-2', 3-3' 4-4'

Bipartite Matching

Max flow formulation.

- Create digraph $G' = (L \cup R \cup \{s, t\},\ E')$.
- Direct all edges from L to R, and assign unit (or infinite) capacity.
- Add source s, and unit capacity edges from s to each node in L.
- Add sink t, and unit capacity edges from each node in R to t.

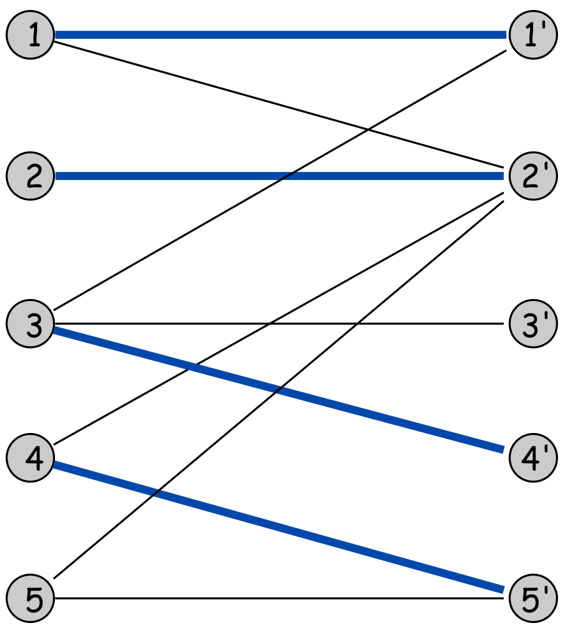


Bipartite Matching: Proof of Correctness

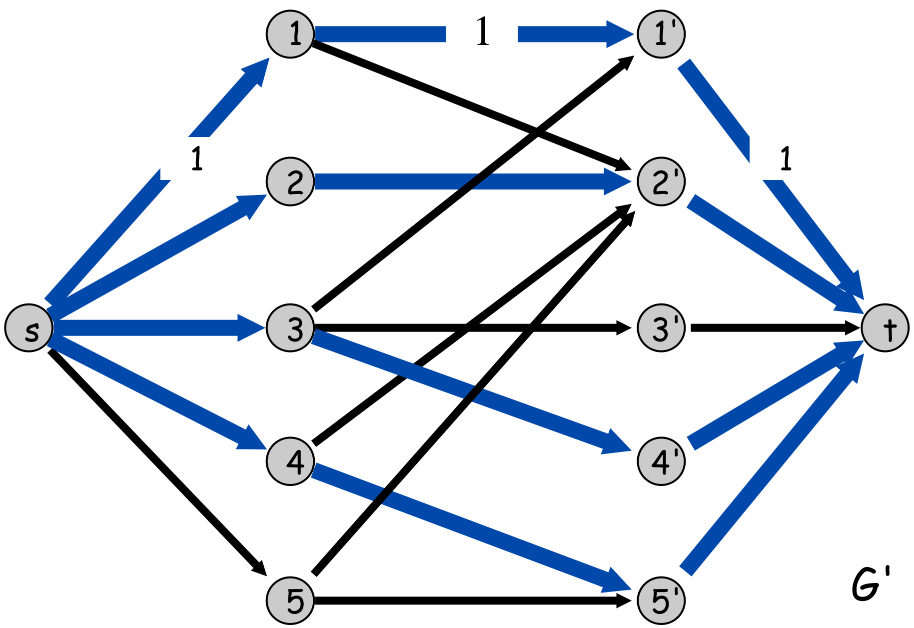
Theorem. Max cardinality matching in G = value of max flow in G'.

Pf. $\leq$

- Given max matching M of cardinality k.
- Consider flow f that sends 1 unit along each of k paths.
- f is a flow, and has value at least k. ▪



G



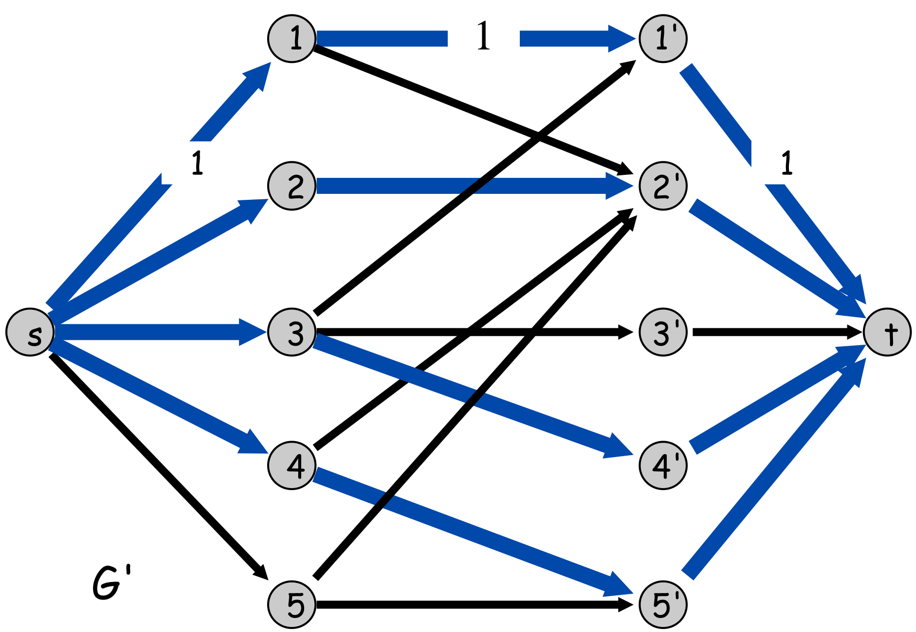
G'

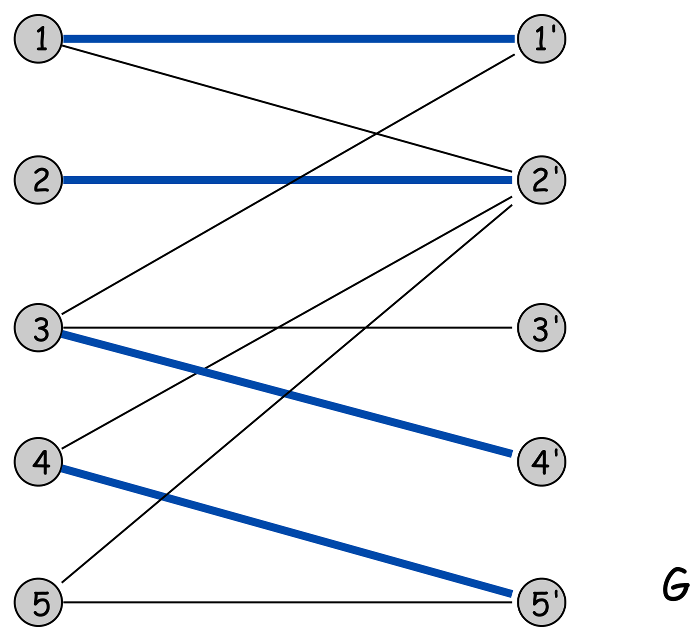
Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in G = value of max flow in G'.

Pf. $\geq$

- Let f be a max flow in G' of value k.
- Integrality theorem $\Rightarrow$ k is integral and can assume f is 0-1.
- Consider M = set of edges from L to R with f(e) = 1.
  - each node in L and R participates in at most one edge in M
  - $|M|$ = k: consider cut $(L \cup s, R \cup t)$ ▪





Bipartite Matching: Running Time

Which max flow algorithm to use for bipartite matching?

- Generic augmenting path: $O(m\ \text{val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path: $O(m\, n^{1/2})$.

Which max flow algorithm to use for Non-bipartite matching?

- Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
- Blossom algorithm: $O(n^4)$. [Edmonds 1965]
- Best known: $O(m\, n^{1/2})$. [Micali-Vazirani 1980]

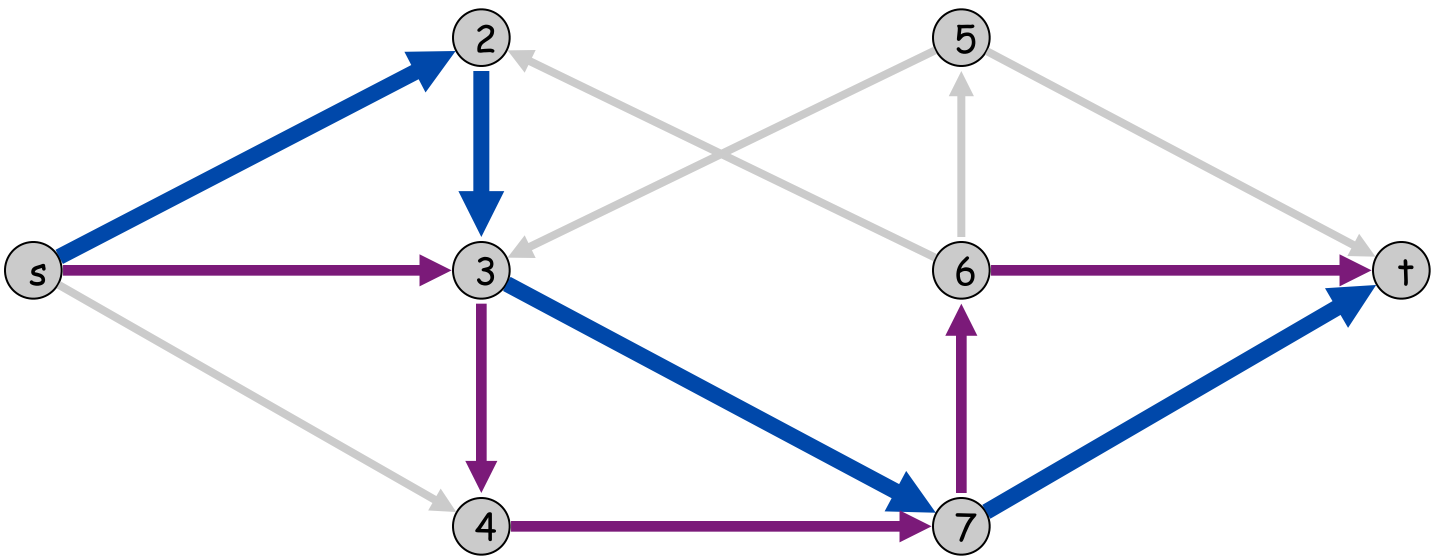
7.6 Disjoint Paths

Edge Disjoint Paths

Disjoint path problem. Given a digraph G = (V, E) and two nodes s and t, find the max number of edge-disjoint s-t paths.

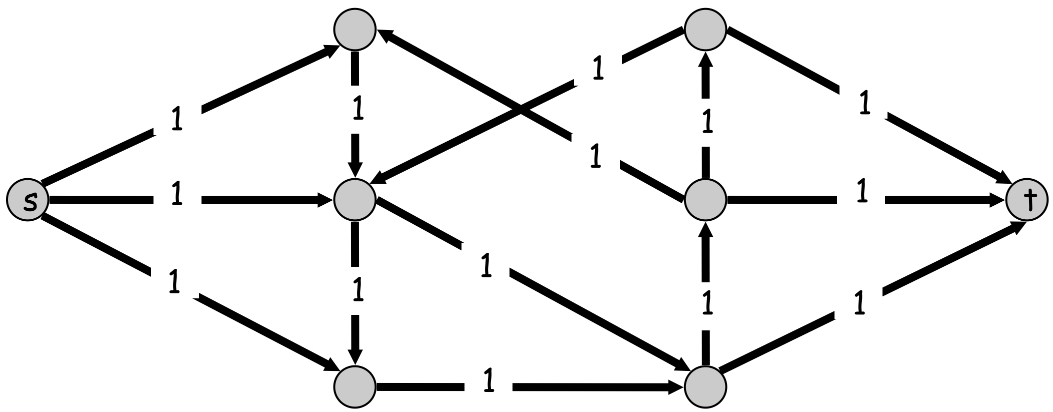
Def. Two paths are edge-disjoint if they have no edge in common.

Ex: communication networks.



Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.



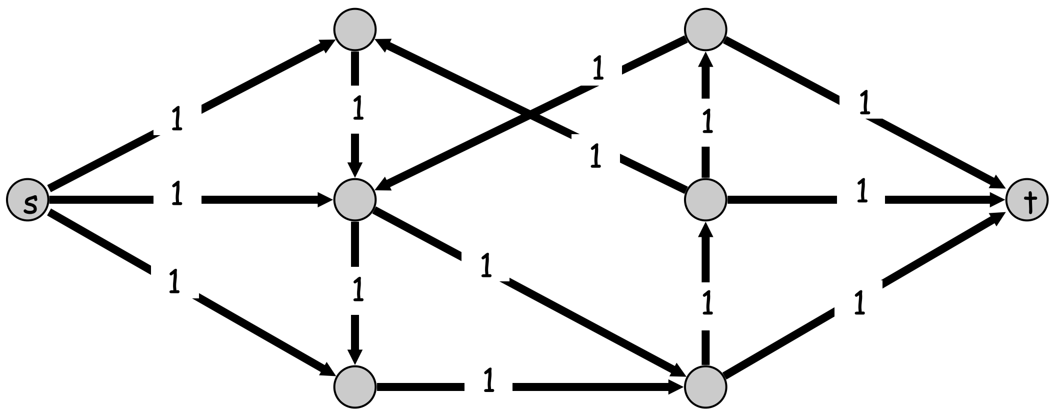
Theorem. Max number edge-disjoint s-t paths equals max flow value.

Pf. $\leq$

- Suppose there are k edge-disjoint paths $P_1, \ldots, P_k$.
- Set $f(e) = 1$ if e participates in some path P_i ; else set $f(e) = 0$.
- Since paths are edge-disjoint, f is a flow of value k. ▪

Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge.



Theorem. Max number edge-disjoint s-t paths equals max flow value.

Pf. $\geq$

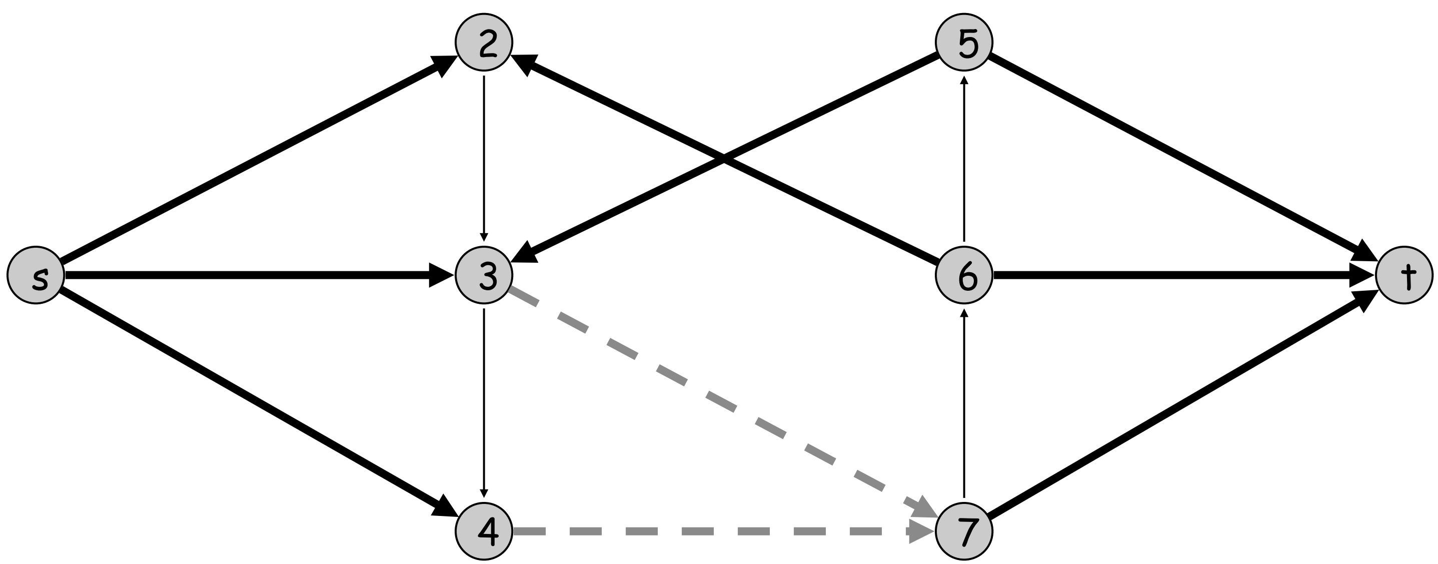
- Suppose max flow value is k.
- Integrality theorem $\Rightarrow$ there exists 0-1 flow f of value k.
- Consider edge (s, u) with f(s, u) = 1.
  - by conservation, there exists an edge (u, v) with f(u, v) = 1
  - continue until reach t, always choosing a new edge
- Produces k (not necessarily simple) edge-disjoint paths. ▪

can eliminate cycles to get simple paths if desired

Network Connectivity

Network connectivity. Given a digraph $G = (V, E)$ and two nodes s and t, find min number of edges whose removal disconnects t from s.

Def. A set of edges $F \subseteq E$ disconnects t from s if all s-t paths uses at least one edge in F.

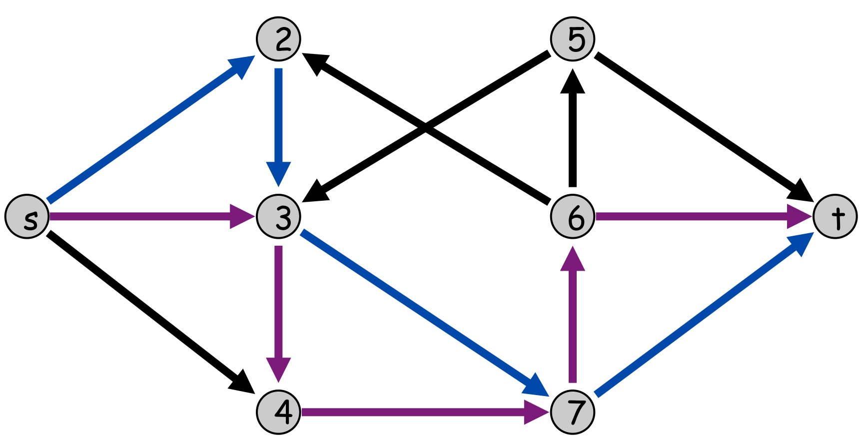


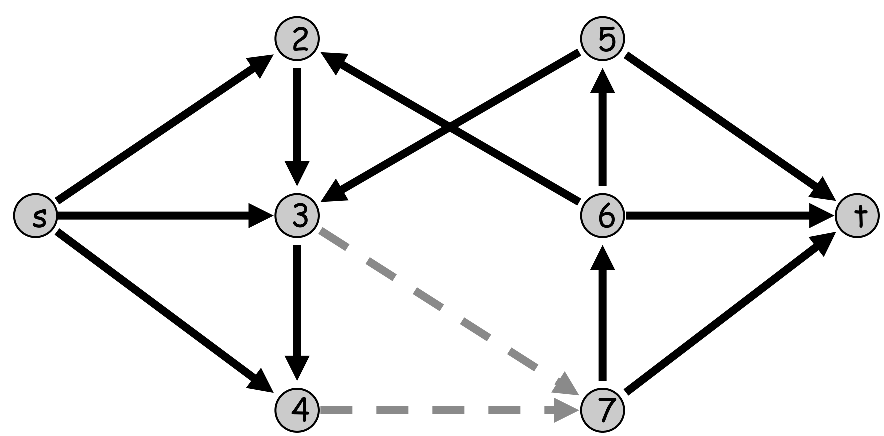
Edge Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

Pf. $\leq$

- Suppose the removal of $F \subseteq E$ disconnects t from s, and $|F| = k$.
- All s-t paths use at least one edge of F. Hence, the number of edge-disjoint paths is at most k. ▪



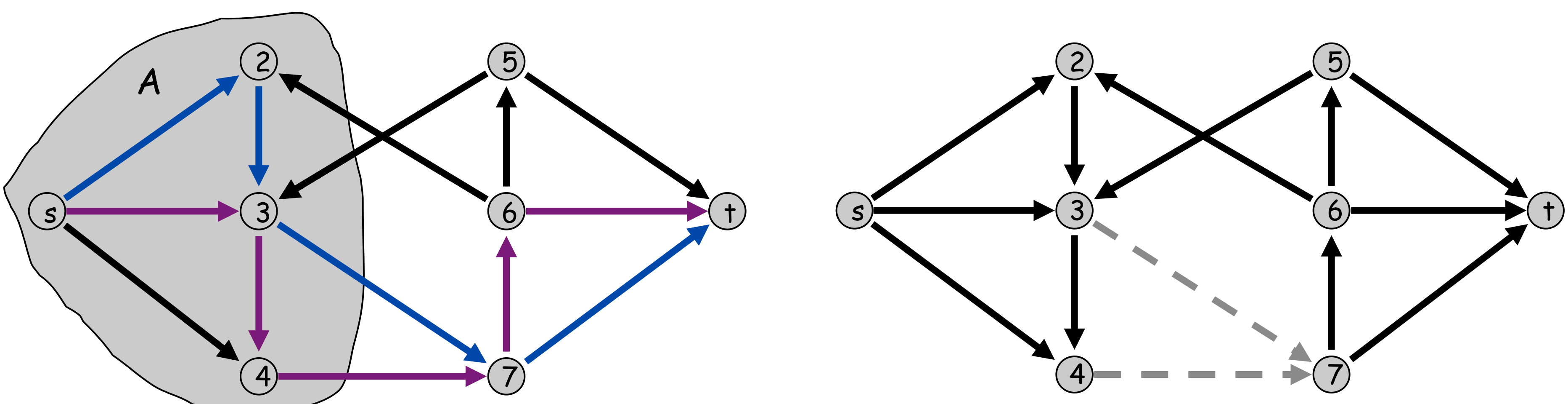


Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

Pf. $\geq$

- Suppose max number of edge-disjoint paths is k.
- Then max flow value is k.
- Max-flow min-cut $\Rightarrow$ cut (A, B) of capacity k.
- Let F be set of edges going from A to B.
- |F| = k and disconnects t from s. ▪



7.12 Baseball Elimination

Acton Vale

Thetford Mines

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}			
				AV	SHA	TM	MTL
AV	83	71	8	-	1	6	1
SHA	80	79	3	1	-	0	2
TM	78	78	6	6	0	-	0
MTL	77	82	3	1	2	0	-

Which teams have a chance of finishing the season with most wins?

- Montréal eliminated since it can finish with at most 80 wins, but Acton Vale already has 83.
- $w_i + r_i < w_j \Rightarrow$ team i eliminated.
- Only reason sports writers appear to be aware of.
- Sufficient, but not necessary!

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}			
AV	83	71	8	-	1	6	1
SHA	80	79	3	1	-	0	2
TM	78	78	6	6	0	-	0
MTL	77	82	3	1	2	0	-

Which teams have a chance of finishing the season with most wins?

- Shawinigan can win 83, but is still eliminated . . .
- If Acton Vale loses a game, then some other team wins one.

Remark. Answer depends not just on how many games already won and left to play, but also on whom they're against.

Baseball Elimination

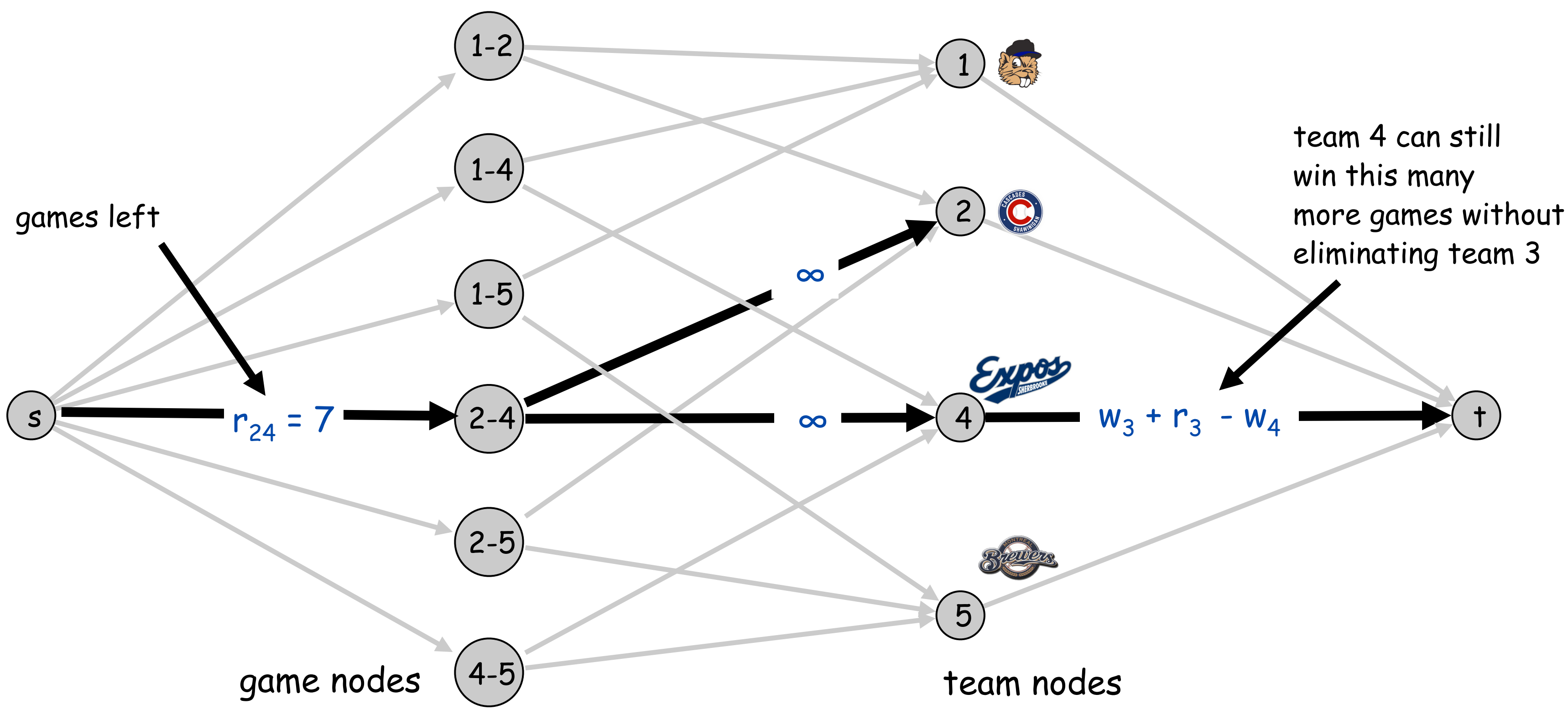
Baseball elimination problem.

- Set of teams S.
- Distinguished team $x \in S$.
- Team x has won w_x games already.
- Teams x and y play each other r_{xy} additional times.
- Is there any outcome of the remaining games in which team x finishes with the most (or tied for the most) wins?

Baseball Elimination: Max Flow Formulation

Can team 3 finish with most wins?

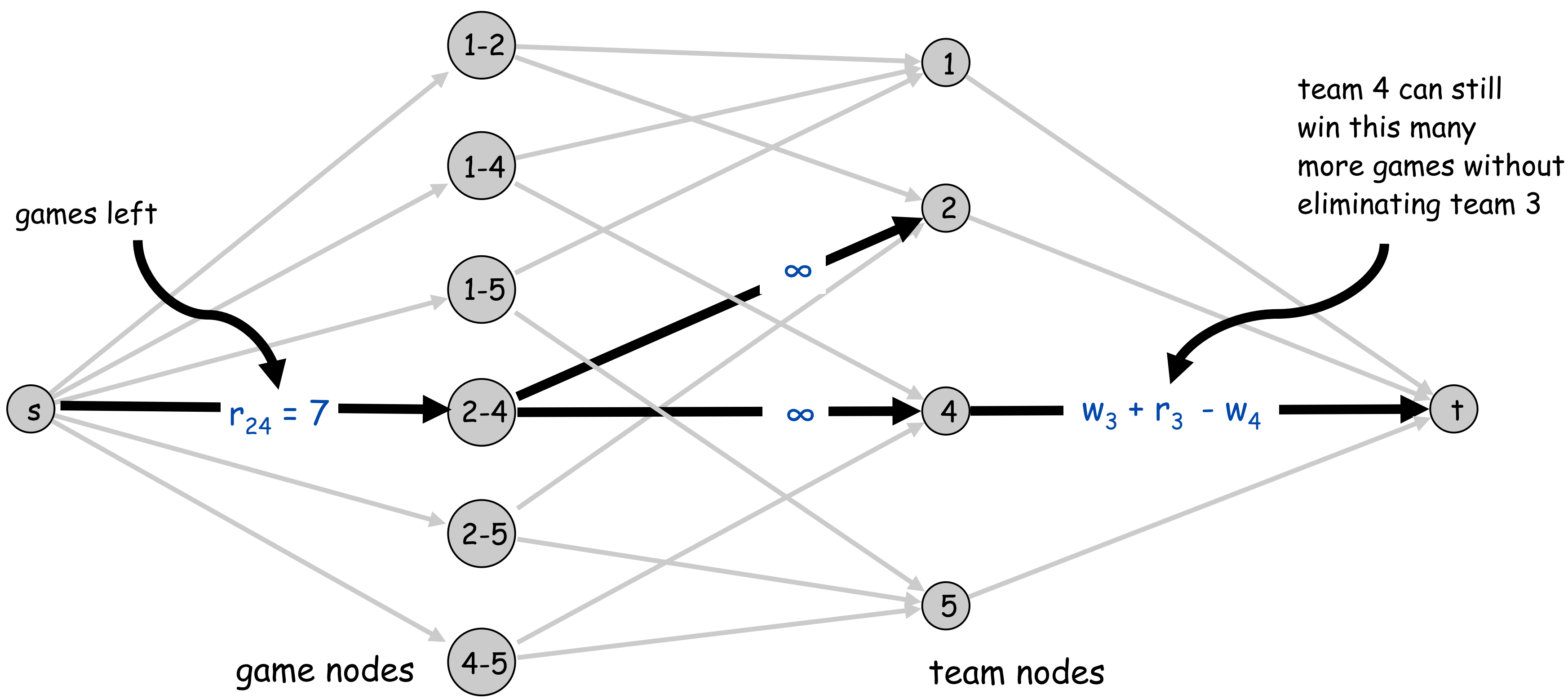
- Assume team 3 wins all remaining games $\Rightarrow$ $w_3 + r_3$ wins.
- Divide remaining games so that all teams have $\leq w_3 + r_3$ wins.



Baseball Elimination: Max Flow Formulation

Theorem. Team 3 is not eliminated iff max flow saturates all edges leaving source s.

- Integrality theorem $\Rightarrow$ each remaining game between x and y added to number of wins for team x or team y.
- Capacity on (x, t) edges ensure no team wins too many games.



Baseball Elimination: Explanation for Sports Writers

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}				
				AV	Sha	TM	She	Mtl
AV	75	59	28	-	3	8	7	3
Sha	71	63	28	3	-	2	7	4
TM	69	66	27	8	2	-	0	0
She	63	72	27	7	7	0	-	-
Mtl	49	86	27	3	4	0	0	-

54

Which teams have a chance of finishing the season with most wins?

- Montréal could finish season with 49 + 27 = 76 wins.

Certificate of elimination. R = {AV, Sha, TM, She}

- Have already won w(R) = 278 games.
- Must win at least r(R) = 27 more (loosing all non-R games).
- Average team in R wins at least 305/4 > 76 games.

Baseball Elimination: Explanation for Sports Writers

Certificate of elimination.

$$T \subseteq S, \quad w(T) := \overbrace{\sum_{i \in T} w_i}^{\#\text{ wins}}, \quad r(T) := \overbrace{\sum_{\{x,y\} \in T} r_{xy}/2}^{\#\text{ remaining games}},$$

If $\overbrace{\dfrac{w(T) + r(T)}{|T|}}^{\text{LB on avg \# games won}} > w_z + r_z$ then z is eliminated (by subset T).

Theorem. [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset T* that eliminates z.

Proof idea. Let T* = team nodes in A (on source side) of min cut (A,B).

Baseball Elimination: Explanation for Sports Writers

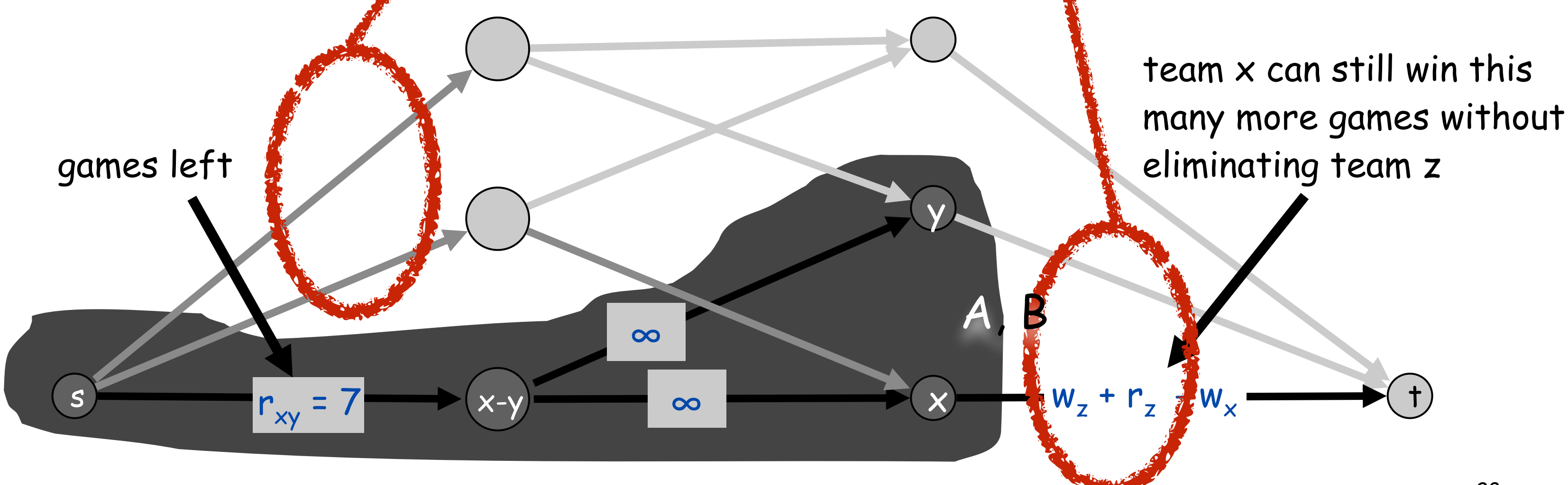
Pf of theorem.

- Use max flow formulation, and consider min cut (A, B).
- Define T* = team nodes in A (on source side of min cut).
- Observe x-y ∈ A iff both x ∈ T* and y ∈ T*.
- $r(S - \{z\}) \;>\; cap(A, B)$

$$= \overbrace{r(S-\{z\}) - r(T^*)}^{\text{capacity of game edges leaving } A} + \overbrace{\sum_{x \in T^*} (w_z + r_z - w_x)}^{\text{capacity of team edges leaving } A}$$

$$= r(S-\{z\}) - r(T^*) - w(T^*) + |T^*|(w_z + r_z)$$

- Rearranging terms: $w_z + r_z < \dfrac{w(T^*) + r(T^*)}{|T^*|}$ ▪



7.10 Image Segmentation

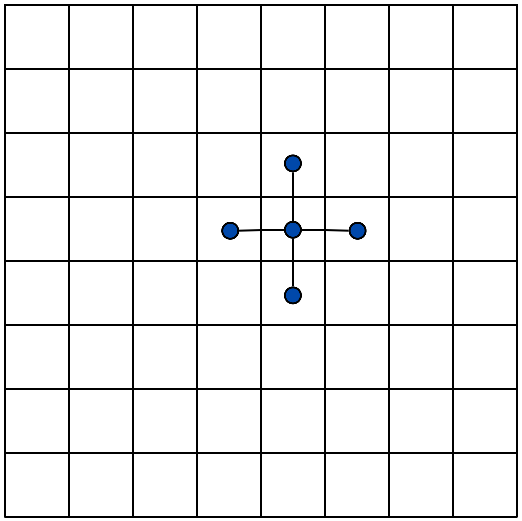
Image Segmentation

Image segmentation.

- Central problem in image processing.
- Divide image into coherent regions.

Ex: Three people standing in front of complex background scene.
Identify each person as a coherent object.

Image Segmentation



Foreground / background segmentation.

- Label each pixel in picture as belonging to foreground or background.
- V = set of pixels, E = pairs of neighbouring pixels.
- $a_i \geq 0$ is likelihood pixel i in foreground.
- $b_i \geq 0$ is likelihood pixel i in background.
- $p_{ij} \geq 0$ is separation penalty for labelling one of i and j as foreground, and the other as background.

Goals.

- Accuracy: if $a_i > b_i$ in isolation, prefer to label i in foreground.
- Smoothness: if many neighbours of i are labeled foreground, we should be inclined to label i as foreground.
- Find partition (A, B) that maximizes: $\displaystyle \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

foreground background

Image Segmentation

Formulate as min cut problem.

- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

- Maximizing $$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

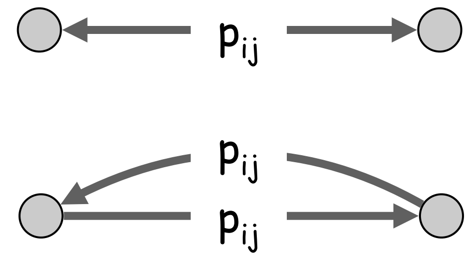
  is equivalent to minimizing $$\underbrace{\left(\sum_{i \in V} a_i + \sum_{j \in V} b_j\right)}_{\text{a constant}} - \sum_{i \in A} a_i - \sum_{j \in B} b_j + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

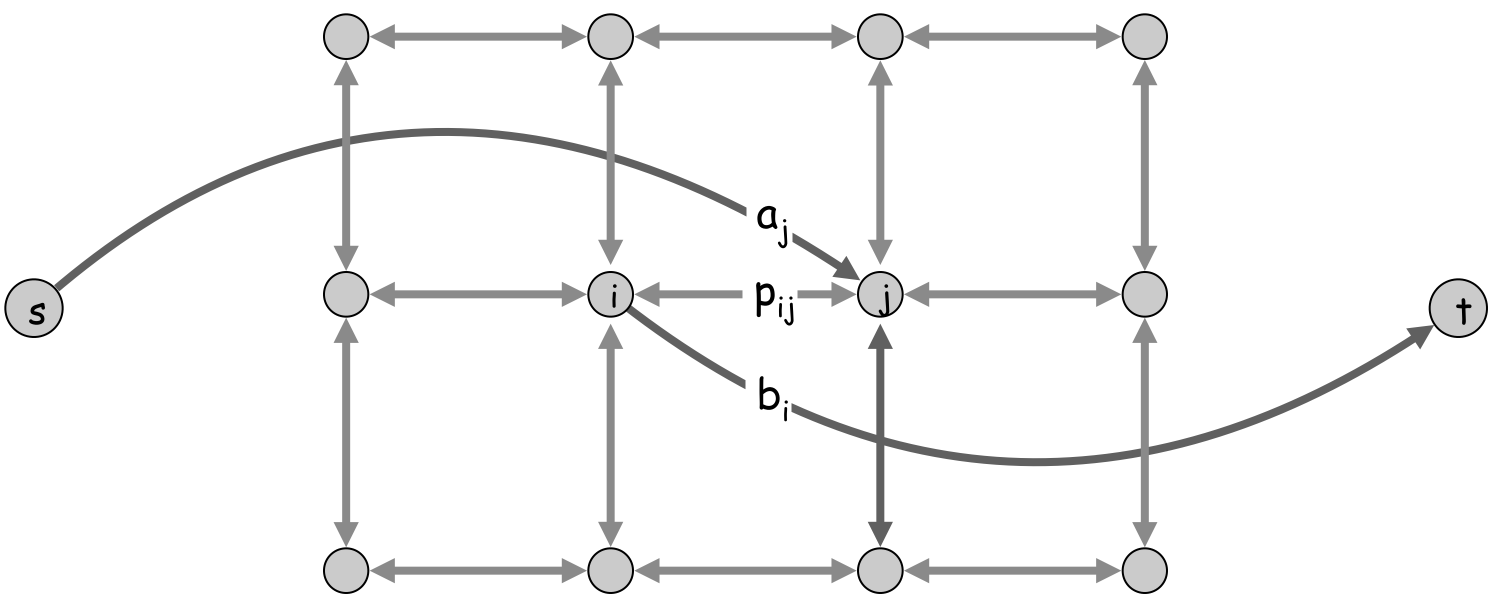
- or alternatively $$\sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

Image Segmentation

Formulate as min cut problem.

- G' = (V', E').
- Add source s to correspond to foreground; add sink t to correspond to background
- Use two anti-parallel edges instead of undirected edge.





G'

Image Segmentation

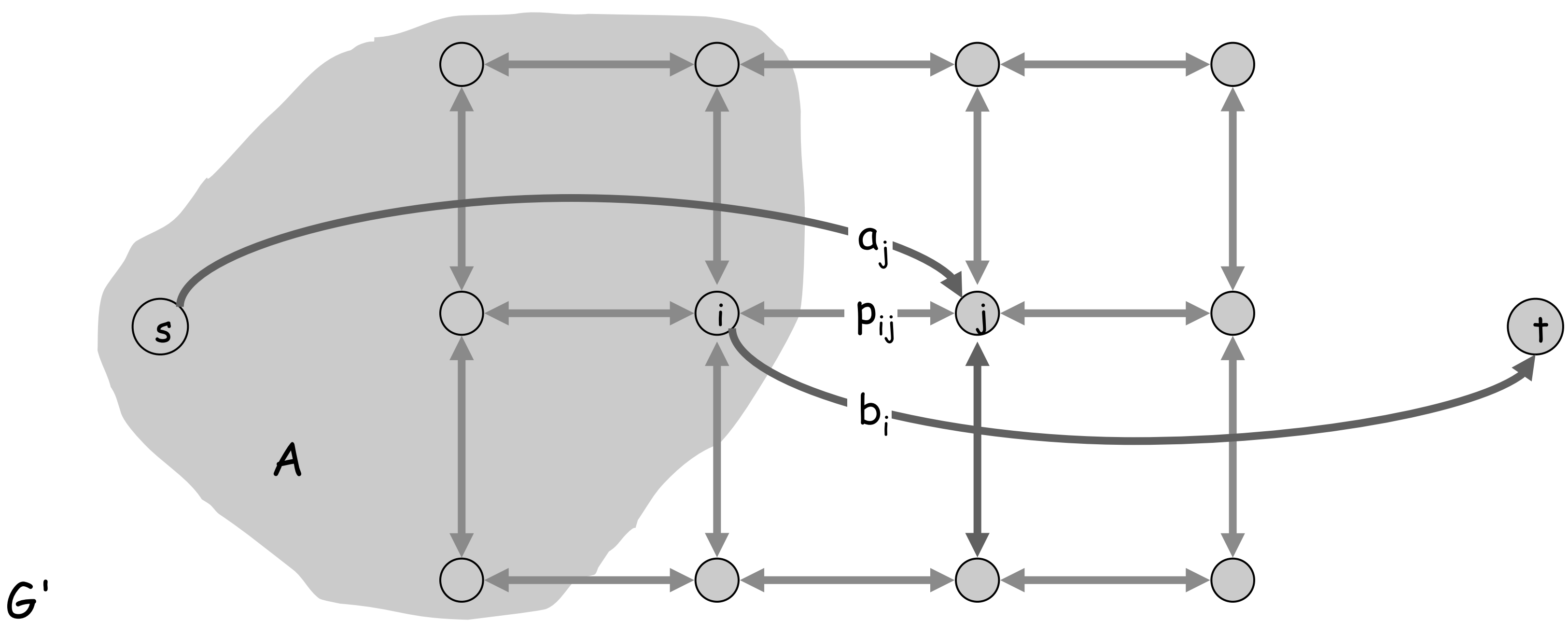
Consider min cut (A, B) in G'.

- A = foreground.

$$cap(A,B) \;=\; \sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ i \in A,\ j \in B}} p_{ij}$$

if i and j on different sides, p_{ij} counted exactly once

- Precisely the quantity we want to minimize.



7.11 Project Selection

Project Selection

can be positive or negative

Projects with prerequisites.

- Set P of possible projects. Project v has associated revenue p_v.
  - some projects generate money: create interactive e-commerce interface, redesign web page
  - others cost money: upgrade computers, get site license
- Set of prerequisites E. If $(v, w) \in E$, can't do project v and unless also do project w.
- A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in A also belongs to A.

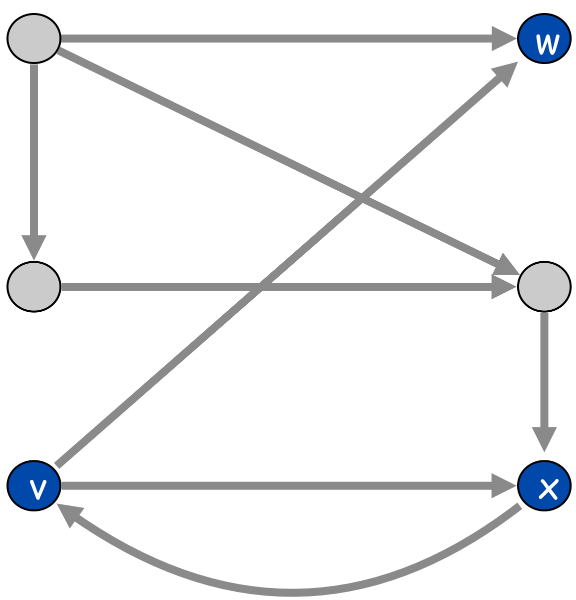
Project selection. Choose a feasible subset of projects to maximize revenue:

$$\sum_{v \in A} p_v$$

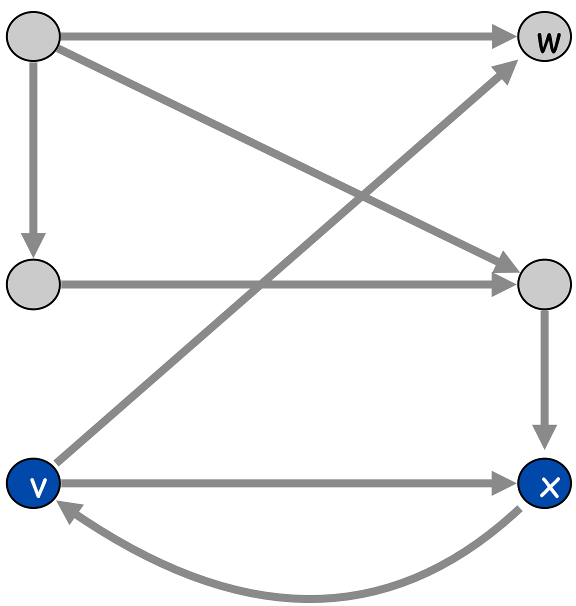
Project Selection: Prerequisite Graph

Prerequisite graph.

- Include an edge from v to w if can't do v without also doing w.
- {v, w, x} is feasible subset of projects.
- {v, x} is infeasible subset of projects.



feasible

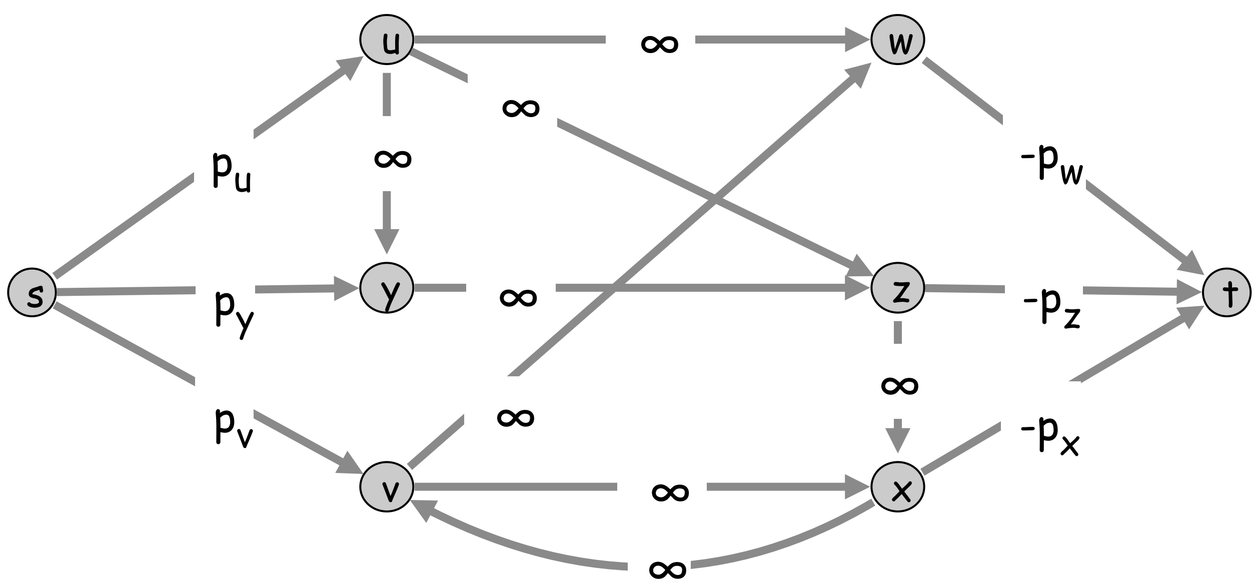


infeasible

Project Selection: Min Cut Formulation

Min cut formulation.

- Assign capacity ∞ to all prerequisite edge.
- Add edge (s, v) with capacity p_v if $p_v > 0$.
- Add edge (v, t) with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$.

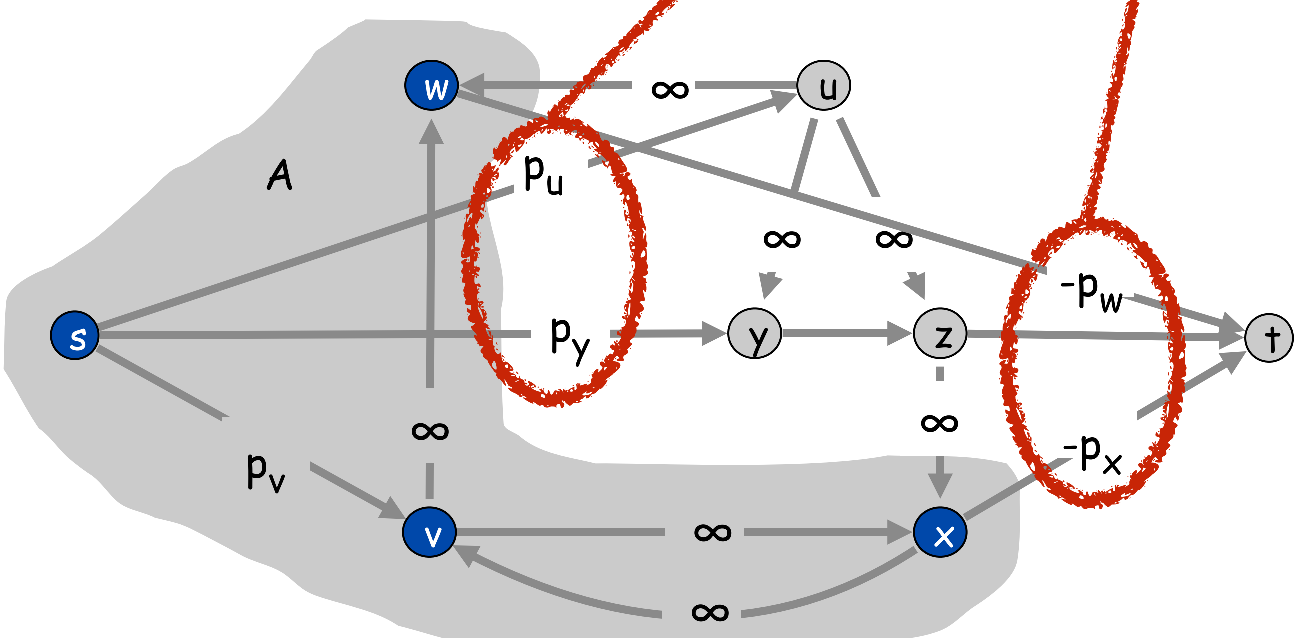


Project Selection: Min Cut Formulation

Claim. (A, B) is min cut iff A – { s } is optimal set of projects.

- Infinite capacity edges ensure A – { s } is feasible.
- Max revenue because:

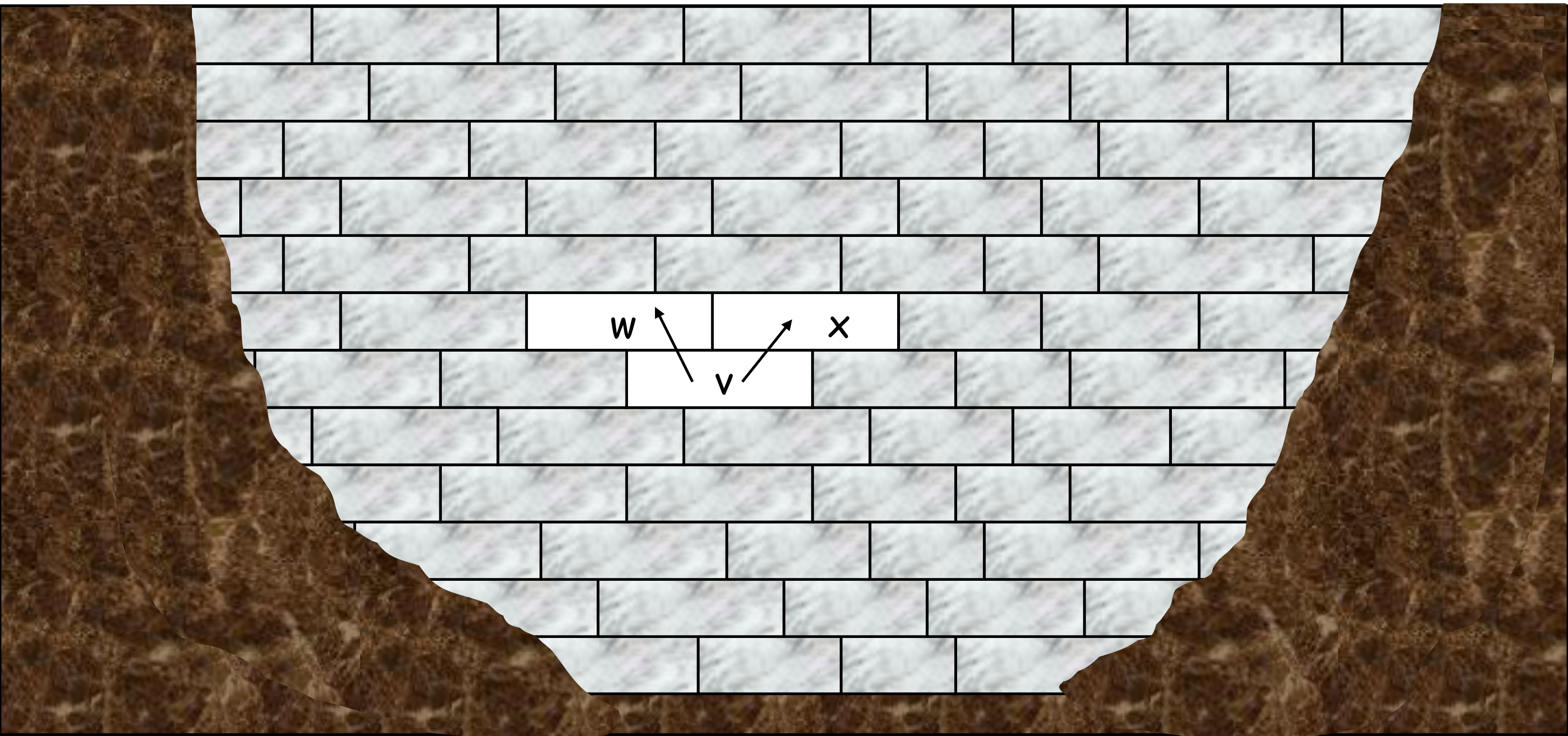
$$cap(A,\ B) \;=\; \sum_{v \in B:\, p_v > 0} p_v \;+\; \sum_{v \in A:\, p_v < 0} (-p_v)$$
$$=\; \underbrace{\sum_{v:\, p_v > 0} p_v}_{\text{constant}} \;-\; \sum_{v \in A} p_v$$



Example: Open Pit Mining

Open-pit mining. (studied since early 1960s)

- Blocks of earth are extracted from surface to retrieve ore.
- Each block v has net value p_v = value of ore – processing cost.
- Can't remove block v before w or x.



7.7 Extensions to Max Flow

Circulation with Demands

Circulation with demands.

- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

demand if $d(v) > 0$; supply if $d(v) < 0$; transshipment if $d(v) = 0$

Def. A circulation is a function f that satisfies:

- For each $e \in E$: $0 \le f(e) \le c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

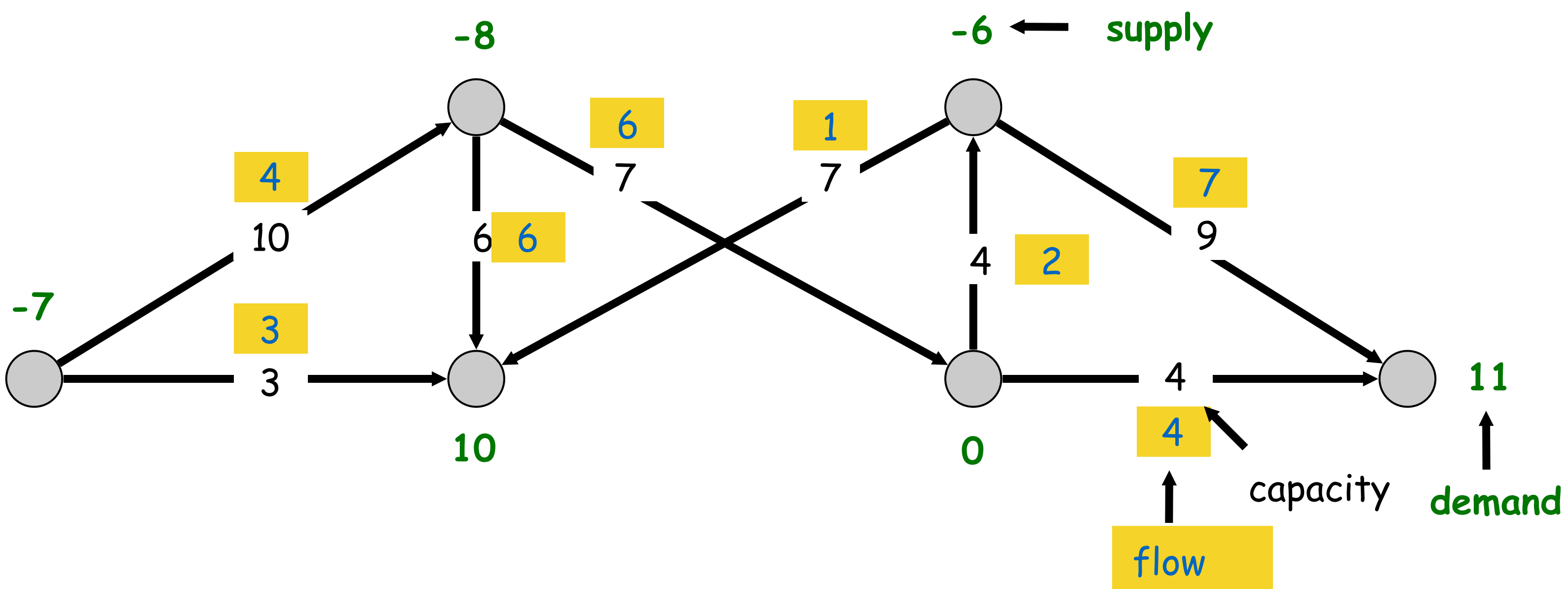
Circulation problem: given (G, c, d), does there exist a circulation?

Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

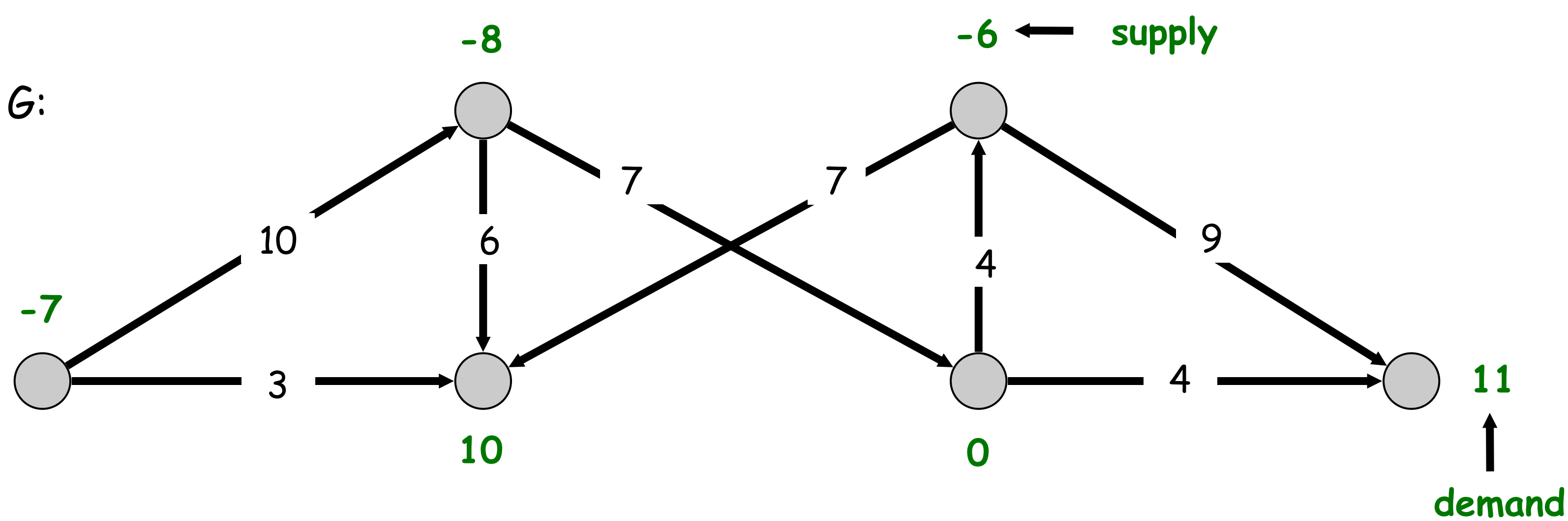
$$\sum_{v\,:\,d(v)>0} d(v) \;=\; \sum_{v\,:\,d(v)<0} -\,d(v) \;=:\; D$$

Pf. Sum conservation constraints for every demand node v.



Circulation with Demands

Max flow formulation.



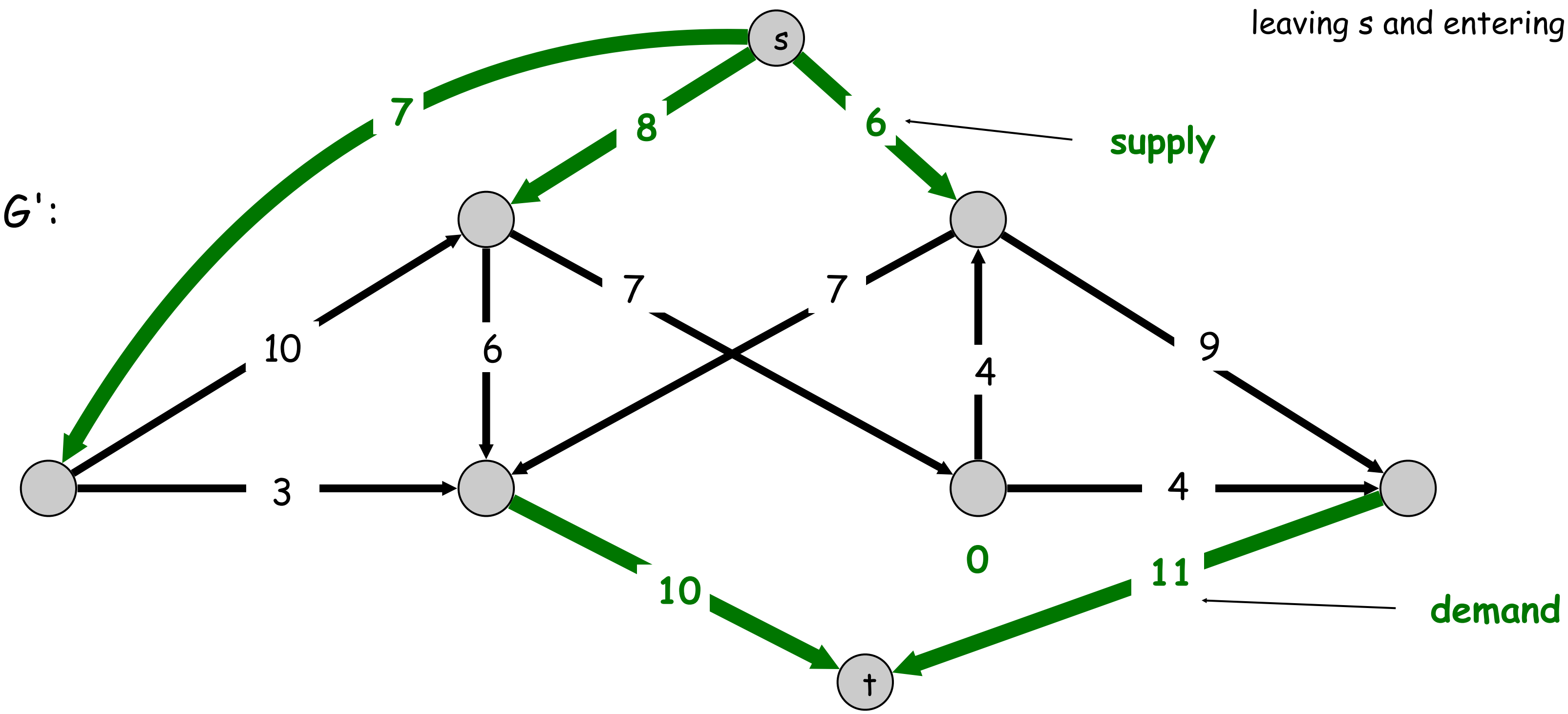
Circulation with Demands

Max flow formulation.

- Add new source s and sink t.
- For each v with $d(v) < 0$, add edge (s, v) with capacity $-d(v)$.
- For each v with $d(v) > 0$, add edge (v, t) with capacity $d(v)$.
- Claim: G has circulation iff G' has max flow of value D.

saturates all edges leaving s and entering t

G':



Circulation with Demands

Integrality theorem. If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

Pf. Follows from max flow formulation and integrality theorem for max flow.

Characterization. Given (G, c, d), there does not exist a circulation iff there exists a node partition (A, B) such that $\Sigma_{v \in B}\, d(v) > cap(A, B)$

Pf idea. Look at min cut in G'.

demand by nodes in B exceeds supply of nodes in B plus max capacity of edges going from A to B

Circulation with Demands and Lower Bounds

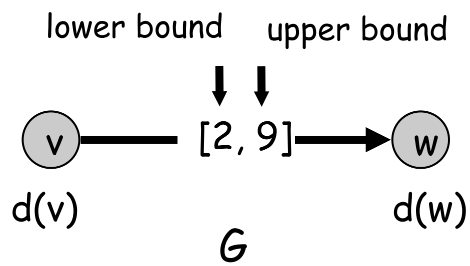
Feasible circulation.
- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$ and lower bounds $\ell(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

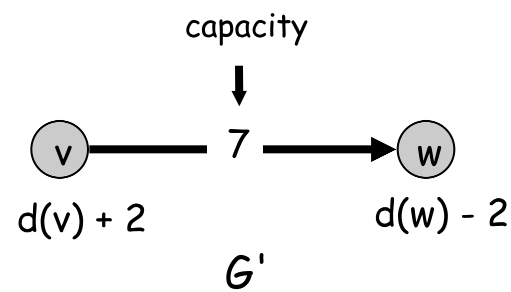
Def. A circulation is a function that satisfies:
- For each $e \in E$: $\ell(e) \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem with lower bounds. Given (G, ℓ, c, d), does there exist a circulation?

Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.
- Send $\ell(e)$ units of flow along edge e.
- Update demands of both endpoints.





Theorem. There exists a circulation in G iff there exists a circulation in G'. If all demands, capacities, and lower bounds in G are integers, then there is a circulation in G that is integer-valued.

Pf sketch. $f(e)$ is a circulation in G iff $f'(e) = f(e) - \ell(e)$ is a circulation in G'.

7.8 Survey Design

Survey Design

Survey design.

- Design survey asking n_c consumers about n_p products.
- Can only survey consumer i about a product j if he owns it.
- Ask consumer i between c_i and c_i' questions.
- Ask between p_j and p_j' consumers about product j.

Goal. Design a survey that meets these specs, if possible.

Survey Design

Algorithm. Formulate as a circulation problem with lower bounds.
- Include an edge (i, j) if customer i owns product j.
- Integer circulation ⇔ feasible survey design.
- (t,s) makes sure problem has a circulation.