CS240 Algorithm Design and Analysis

Lecture 9

Network Flow (Cont.)

Quan Li
Fall 2024
2024.10.27

Baseball Elimination

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}			
				Atl	Phi	NY	Mon
Atlanta	83	71	8	-	1	6	1
Philly	80	79	3	1	-	0	2
New York	78	78	6	6	0	-	0
Montreal	77	82	3	1	2	0	-

- **Which teams have a chance of finishing the season with most wins?**
  - Montreal eliminated since it can finish with at most 80 wins, but Atlanta already has 83
  - $w_i + r_i < w_j \rightarrow$ team i eliminated
  - Sufficient, but not necessary!

Baseball Elimination

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}			
				Atl	Phi	NY	Mon
Atlanta	83	71	8	-	1	6	1
Philly	80	79	3	1	-	0	2
New York	78	78	6	6	0	-	0
Montreal	77	82	3	1	2	0	-

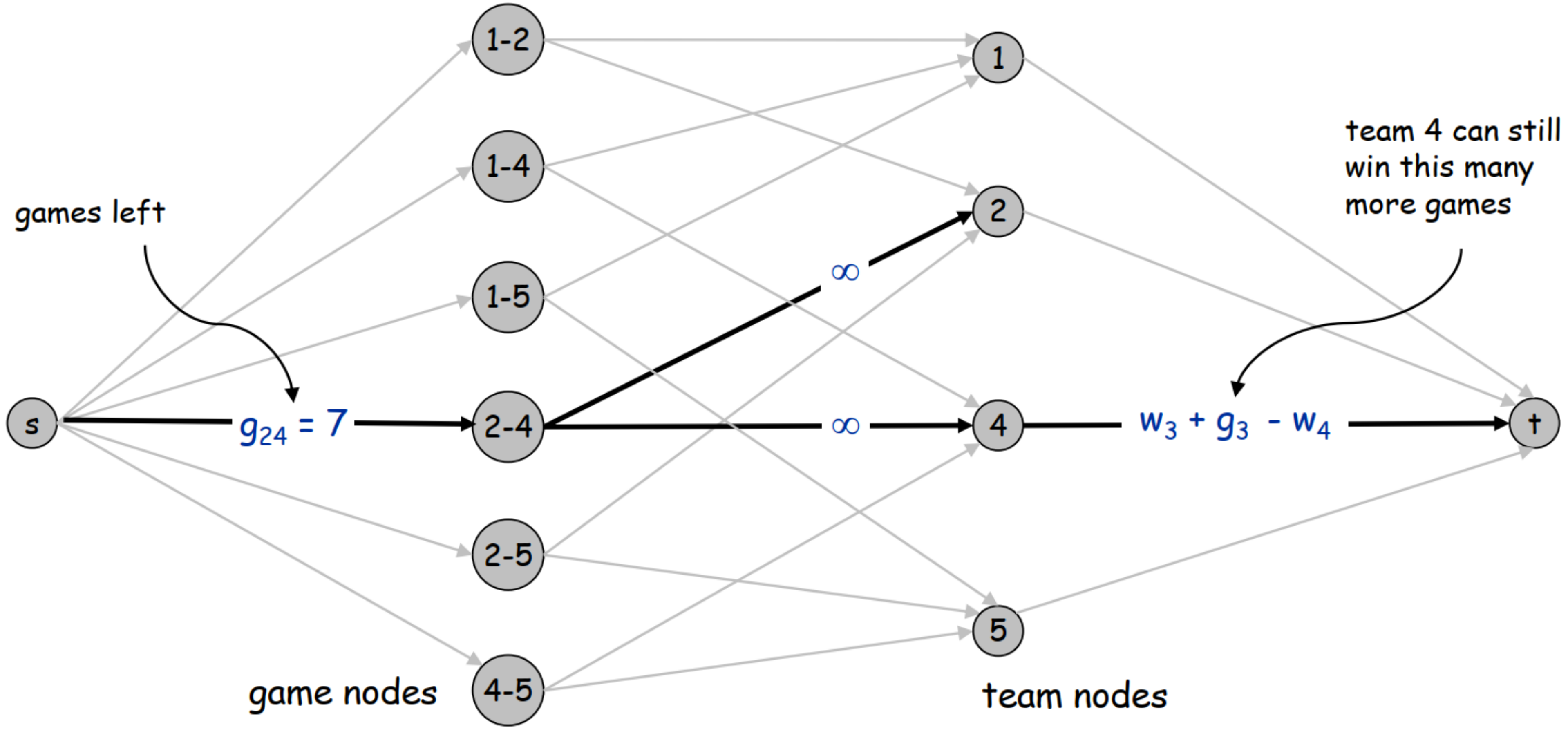
- **Which teams have a chance of finishing the season with most wins?**
  - Philly can win 83, but still eliminated…
  - If Atlanta loses all games, then New York wins 84…

Baseball Elimination

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

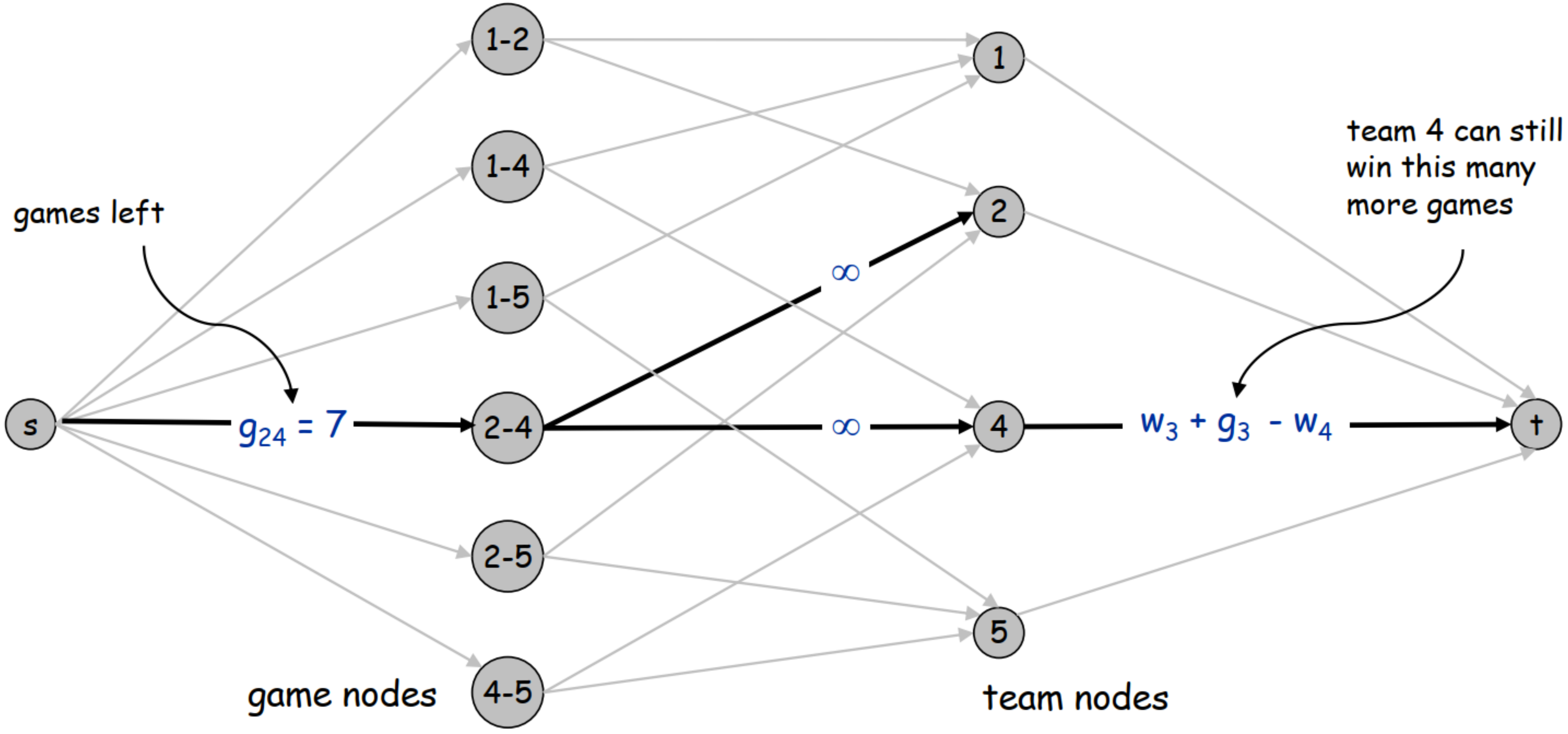
Baseball Elimination: Max Flow Formulation

- **Can team 3 finish with most wins?**
  - Assume team 3 wins all remaining games → $w_3 + g_3$ wins
  - Divvy remaining games so that all teams have <= $w_3 + g_3$ wins



Baseball Elimination: Max Flow Formulation

- **Theorem.** Team 3 is not eliminated iff max flow saturates all edges leaving source
  - Integrality theorem → 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}				
				NY	Bal	Bos	Tor	Det
NY	75	59	28	-	3	8	7	3
Baltimore	71	63	28	3	-	2	7	4
Boston	69	66	27	8	2	-	0	0
Toronto	63	72	27	7	7	0	-	-
Detroit	49	86	27	3	4	0	0	-

- **Which teams have a chance of finishing the season with most wins?**
  - Detroit could finish season with 49 + 27 = 76 wins

Baseball Elimination: Explanation for Sports Writers

Team i	Wins w_i	Losses l_i	To play r_i	Against = r_{ij}				
				NY	Bal	Bos	Tor	Det
NY	75	59	28	-	3	8	7	3
Baltimore	71	63	28	3	-	2	7	4
Boston	69	66	27	8	2	-	0	0
Toronto	63	72	27	7	7	0	-	-
Detroit	49	86	27	3	4	0	0	-

AL East: August 30, 1996

- **Which teams have a chance of finishing the season with most wins?**
  - Detroit could finish season with 49 + 27 = 76 wins
- **Certificate of elimination. R = {Ny, Bal, Bos, Tor}**
  - Have already won w(R) = 278 games
  - Remaining games among R is r(R) = 3+8+7+2+7 = 27
  - Average team in R wins at least (278 + 27)/4 > 76 games

Baseball Elimination: Explanation for Sports Writers

- **Certificate of elimination**

$$T \subseteq S, \quad w(T) \coloneqq \overbrace{\sum_{i \in T} w_i}^{\#\text{ wins}}, \quad g(T) \coloneqq \overbrace{\sum_{\{x,y\} \subseteq T} g_{xy}}^{\#\text{ remaining games}},$$

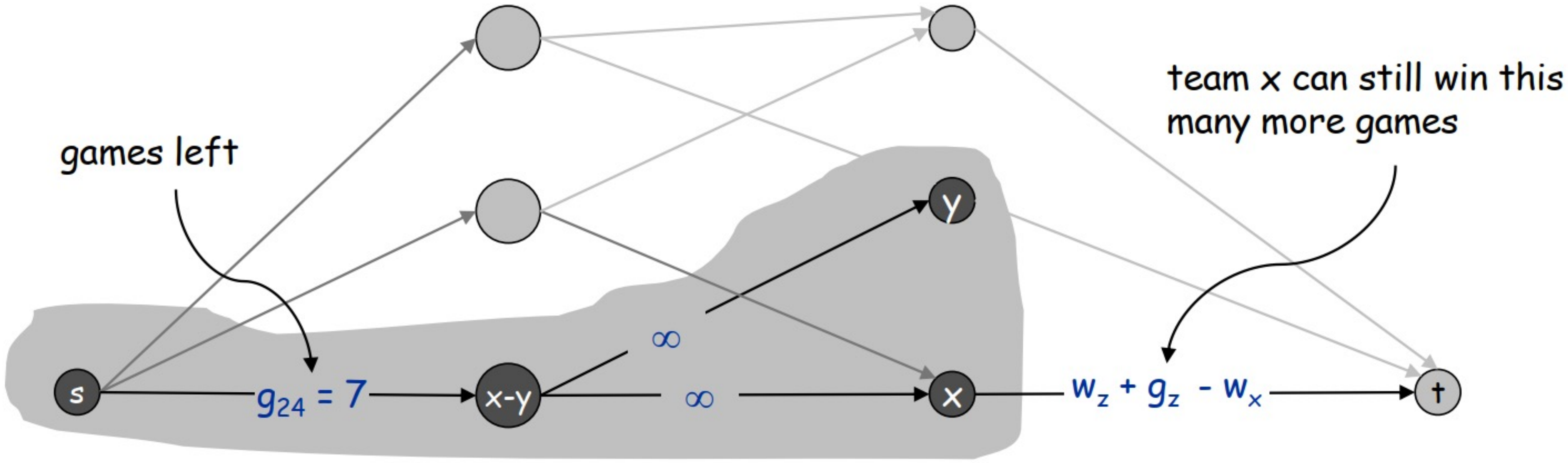
- **Theorem.** [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset T* such that

$$\frac{\overbrace{w(T^*) + g(T^*)}^{\text{LB on avg \# games won}}}{|T^*|} > w_z + g_z$$

- **Proof.** ←
  - The average number of wins of teams in T* is larger than the maximum number of wins of z

Baseball Elimination: Explanation for Sports Writers

- **Proof.** →
  - Use max flow formulation, and consider min cut (A, B)
  - Define T^* = team nodes on source side of min cut
  - Observer x-y ∈ A iff both x ∈ T^* and y ∈ T^*
    - Infinite capacity edges ensure if x-y ∈ A then x ∈ A and y ∈ A
    - If x ∈ A and y ∈ A but x-y ∈ B, then adding x-y to A decreases capacity of cut



Baseball Elimination: Explanation for Sports Writers

- **Proof.** →
  - Use max flow formulation, and consider min cut (A, B)
  - Define T* = team nodes on source side of min cut
  - Observer x-y ∈ A iff both x ∈ T* and y ∈ T*
  - Since z is eliminated, by max-flow min-cut theorem:

$$
\begin{aligned}
g(S-\{z\}) &> cap(A,B) \\
&= \overbrace{g(S-\{z\})-g(T^*)}^{\text{capacity of game edges leaving s}} + \overbrace{\sum_{x\in T^*}(w_z+g_z-w_x)}^{\text{capacity of team edges entering t}} \\
&= g(S-\{z\})-g(T^*)-w(T^*)+|T^*|(w_z+g_z)
\end{aligned}
$$

Rearranging terms: $w_z+g_z < \dfrac{w(T^*)+g(T^*)}{|T^*|}$ ▪

Project Selection

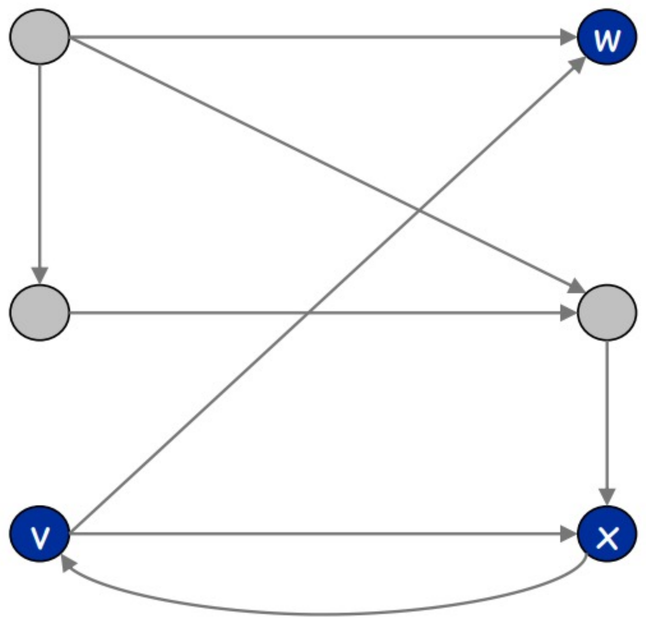
Project Selection

- **Projects with prerequisites**
  - Set P of possible projects. Project v has associated revenue p_v (Can be positive or negative)
    - Some projects generate money: create e-commerce interface, design web page
    - Others cost money: upgrade computers, get site license
  - Set of prerequisites E. If $(v, w) \in E$, can't do project v 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

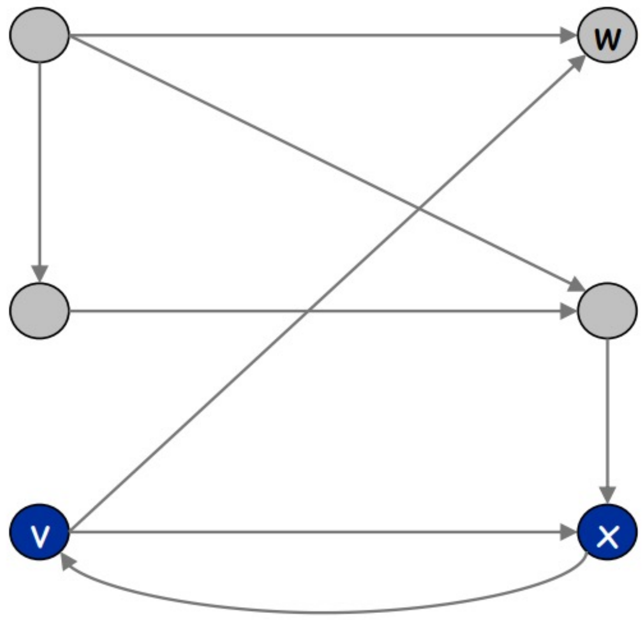
- **Project selection.** Choose a feasible subset of projects to maximize revenue

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



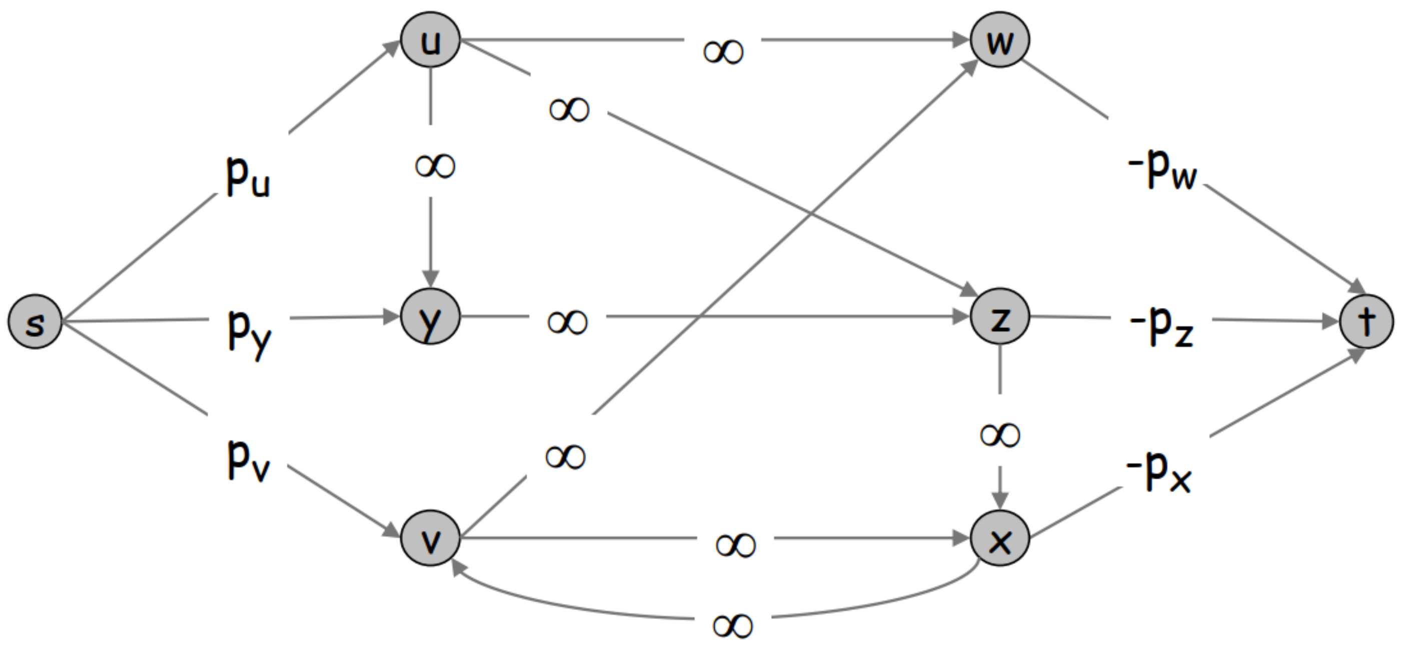
feasible



infeasible

Project Selection: Min Cut Formulation

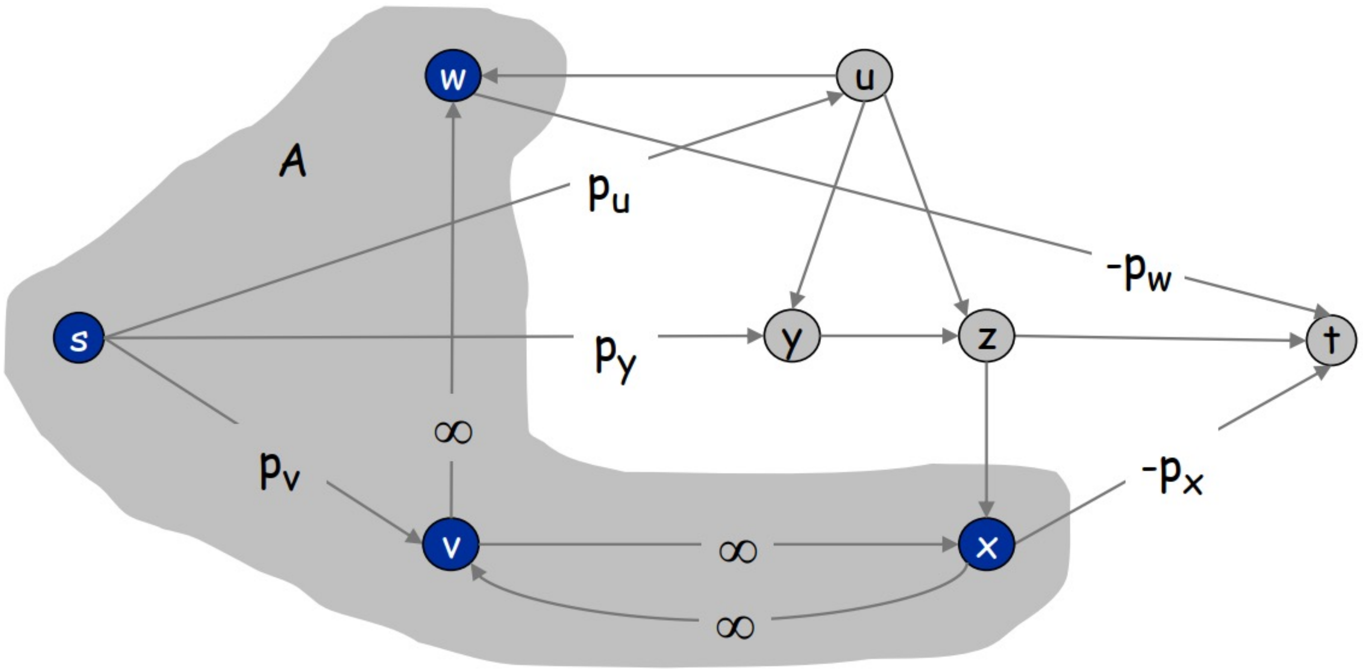
- **Min Cut formulation**
  - Assign capacity ∞ to all prerequisite edges
  - Add edge (s, v) with capacity p_v if $p_v > 0$
  - Add edge (v, t) with capacity $-p_v$ if $p_v < 0$
  - For notational convenience, define $p_s = p_t = 0$



Project Selection: Min Cut Formulation

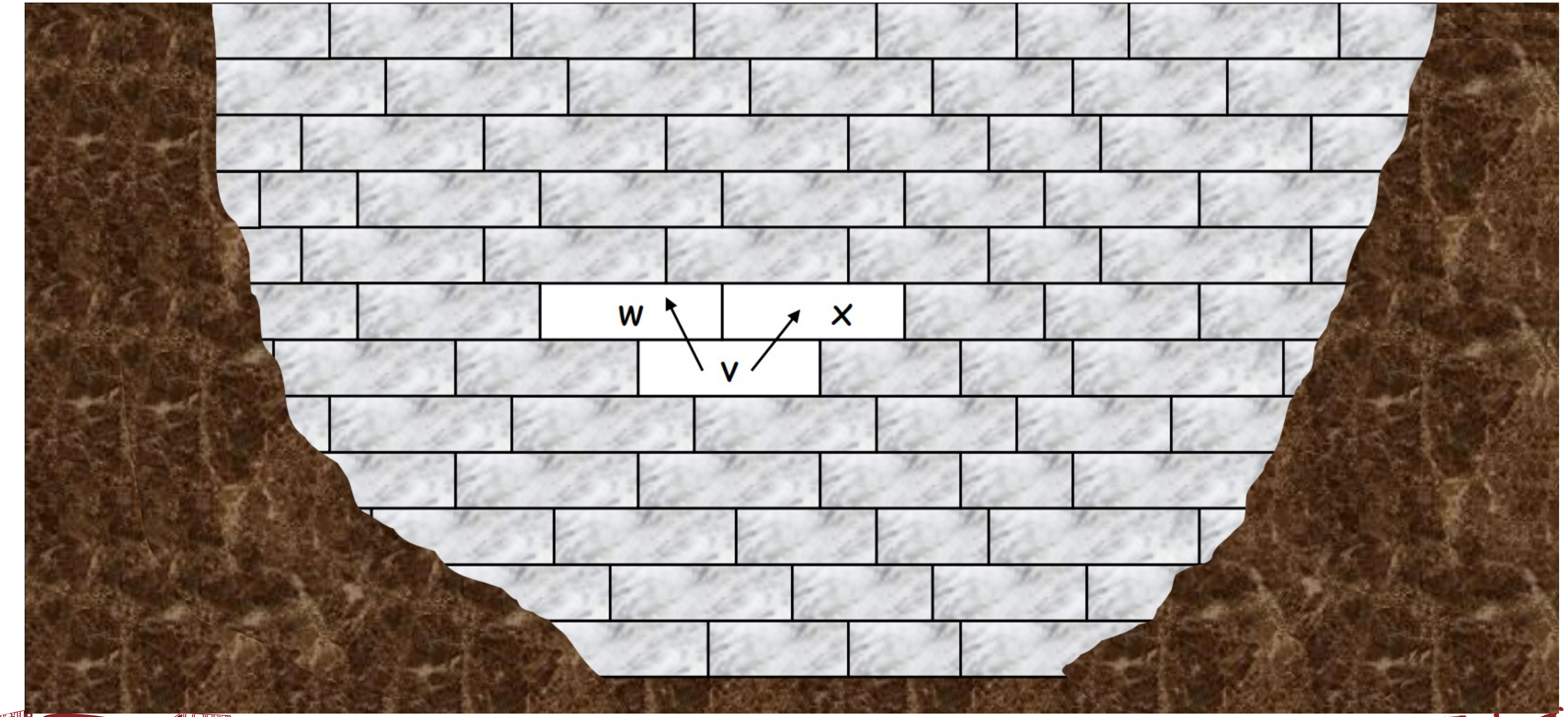
- **Claim.** (A, B) is min cut iff A – { s } is optimal set of project
  - Infinite capacity edges ensure A – { s } is feasible
  - Max revenue because:

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



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

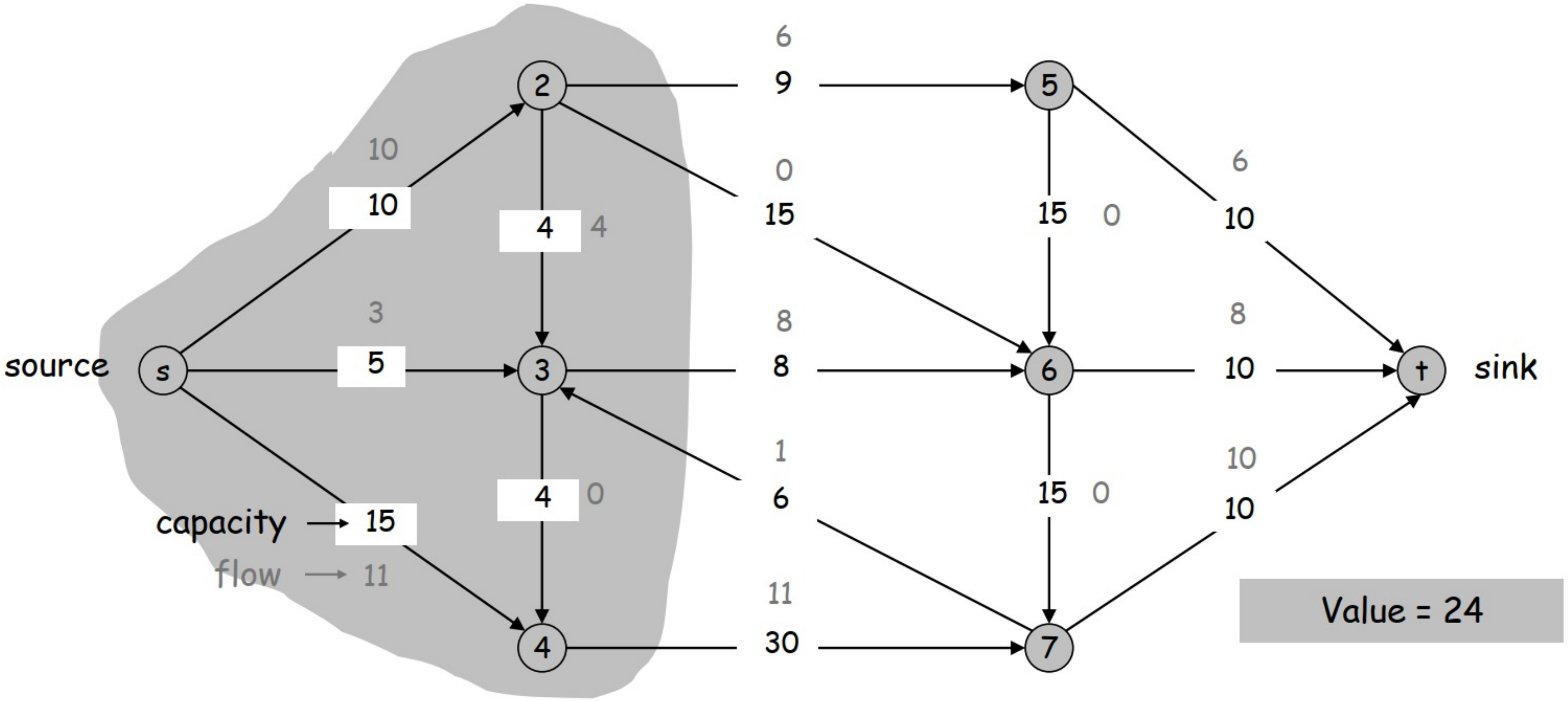


Summary

Flows

- **Concepts**
  - s-t flow
  - Max-flow
  - s-t cut
  - Min-cut

Max-flow min-cut theorem: The value of the max flow is equal to the value of the min cut

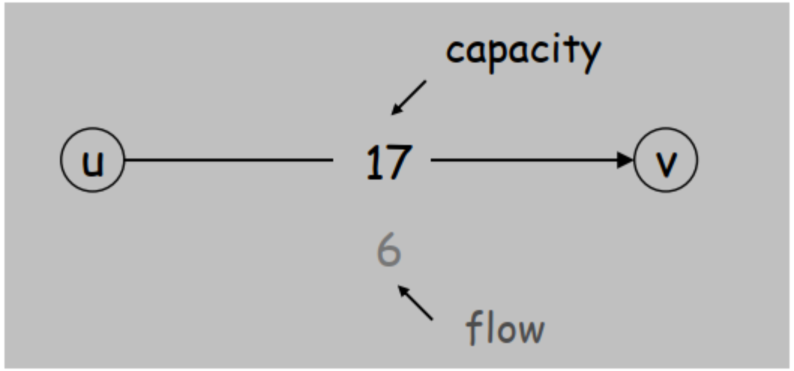


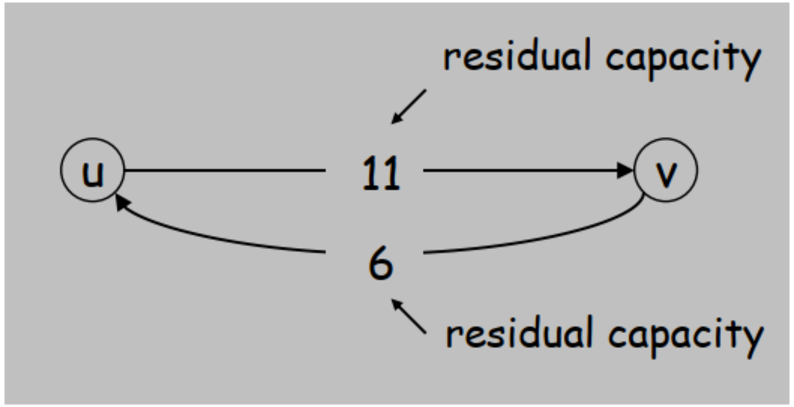
Ford-Fulkerson Algorithm

- **Ford-Fulkerson Algorithm**
  - Start with f(e) = 0 for all edge e ∈ E
  - Find an augmenting path P in the residual graph G_f
    - Can be chosen using capacity scaling
  - Augment flow along path P
  - Repeat until you get stuck

```
Ford-Fulkerson(G, s, t, c) {
   foreach e ∈ E  f(e) ← 0
   Gf ← residual graph

   while (there exists augmenting path P) {
      f ← Augment(f, c, P)
      update Gf
   }
   return f
}
```





Applications

- **Problems covered in class**
  - Bipartite Matching
  - Disjoint Paths
  - Circulation with Demands (+ edge lower bounds)
  - Survey Design
  - Image Segmentation
  - Baseball Elimination
  - Project Selection