CSE 421
Algorithms

Richard Anderson

Lecture 24

Network Flow Applications

Today’s topics

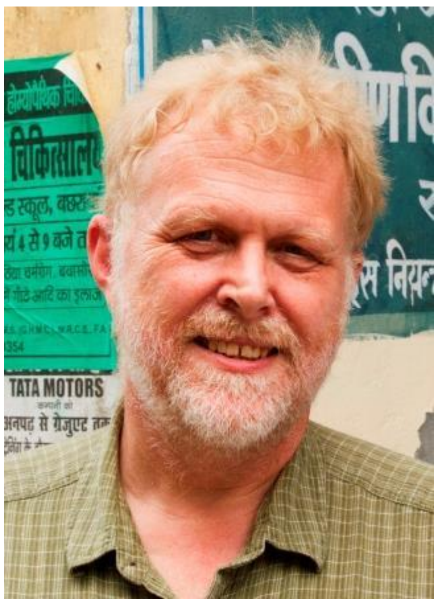
- Image Segmentation
- Strip Mining
- Reading: 7.5, 7.6, 7.10-7.12

Minimum Cut Applications

- Image Segmentation
- Open Pit Mining / Task Selection Problem
- Reduction to Min Cut problem

S, T is a cut if S, T is a partition of the vertices with
s in S and t in T
The capacity of an S, T cut is the sum of the capacities of
all edges going from S to T

Image Segmentation

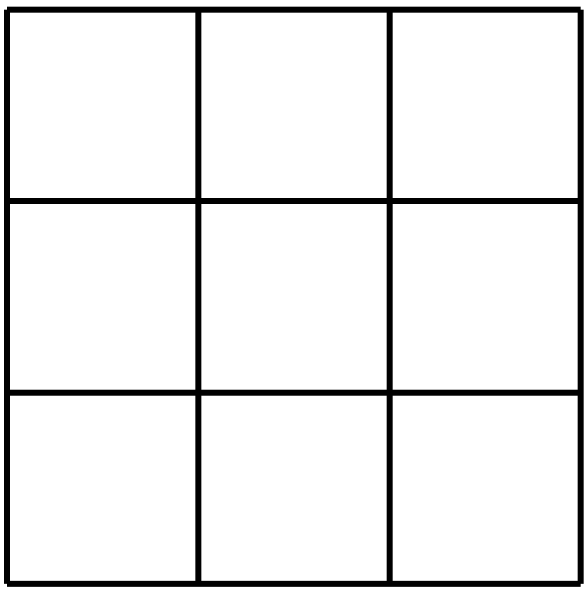


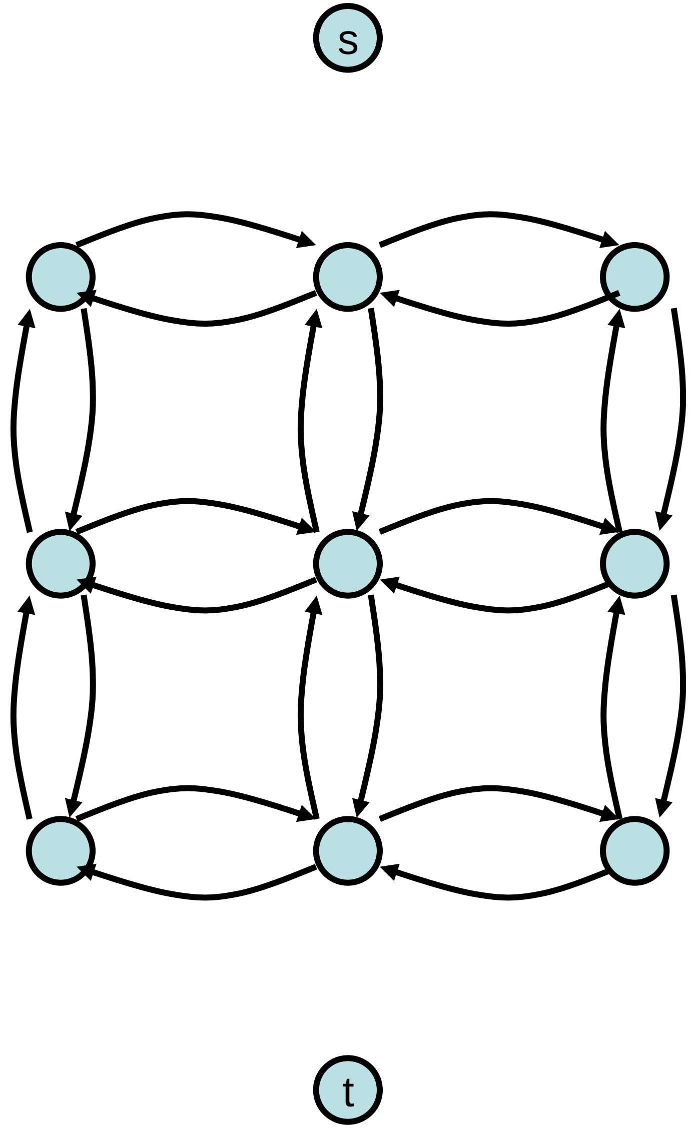
Separate Lion from Savana

Image analysis

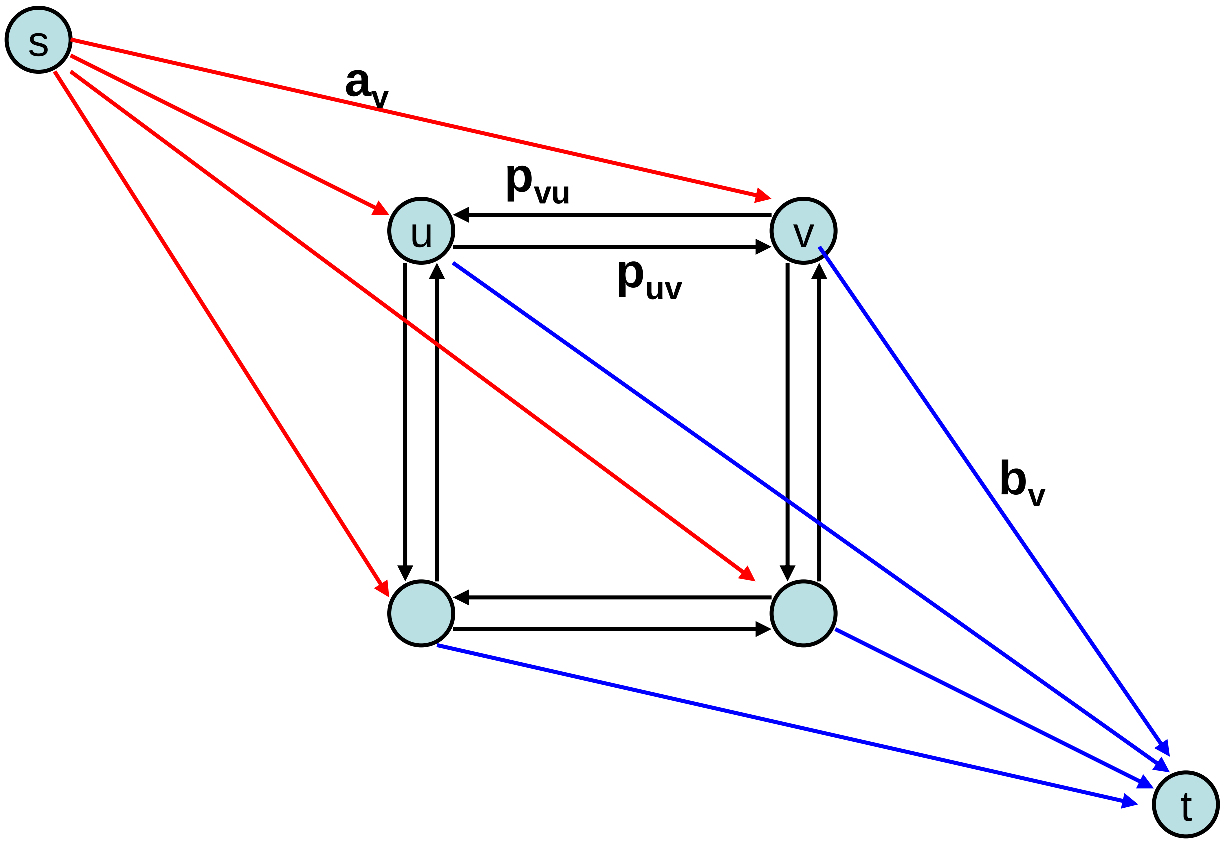
- a_i: value of assigning pixel i to the foreground
- b_i: value of assigning pixel i to the background
- p_{ij}: penalty for assigning i to the foreground, j to the background or vice versa
- A: foreground, B: background
- $Q(A,B) = \Sigma_{\{i \text{ in } A\}} a_i + \Sigma_{\{j \text{ in } B\}} b_j - \Sigma_{\{(i,j) \text{ in } E, \text{ i in } A, \text{ j in } B\}} p_{ij}$

Pixel graph to flow graph





Mincut Construction



Open Pit Mining

Application of Min-cut

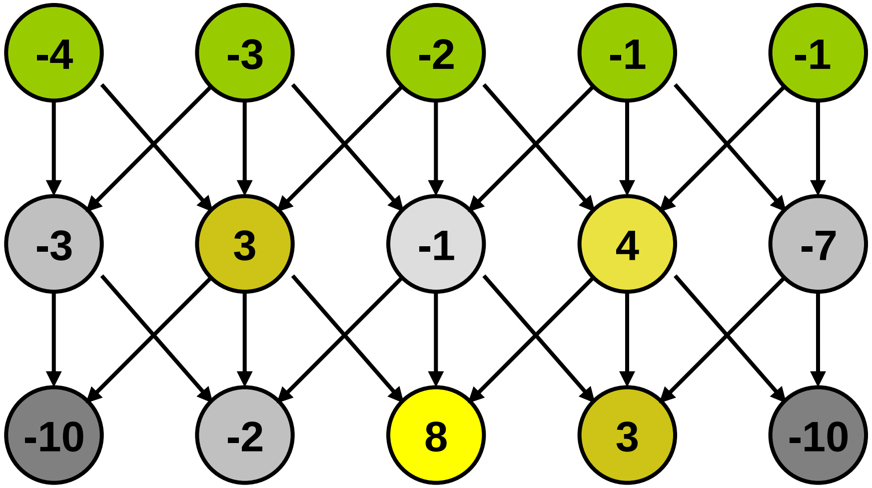
- Open Pit Mining Problem
- Task Selection Problem
- Reduction to Min Cut problem

S, T is a cut if S, T is a partition of the vertices with
s in S and t in T
The capacity of an S, T cut is the sum of the capacities of
all edges going from S to T

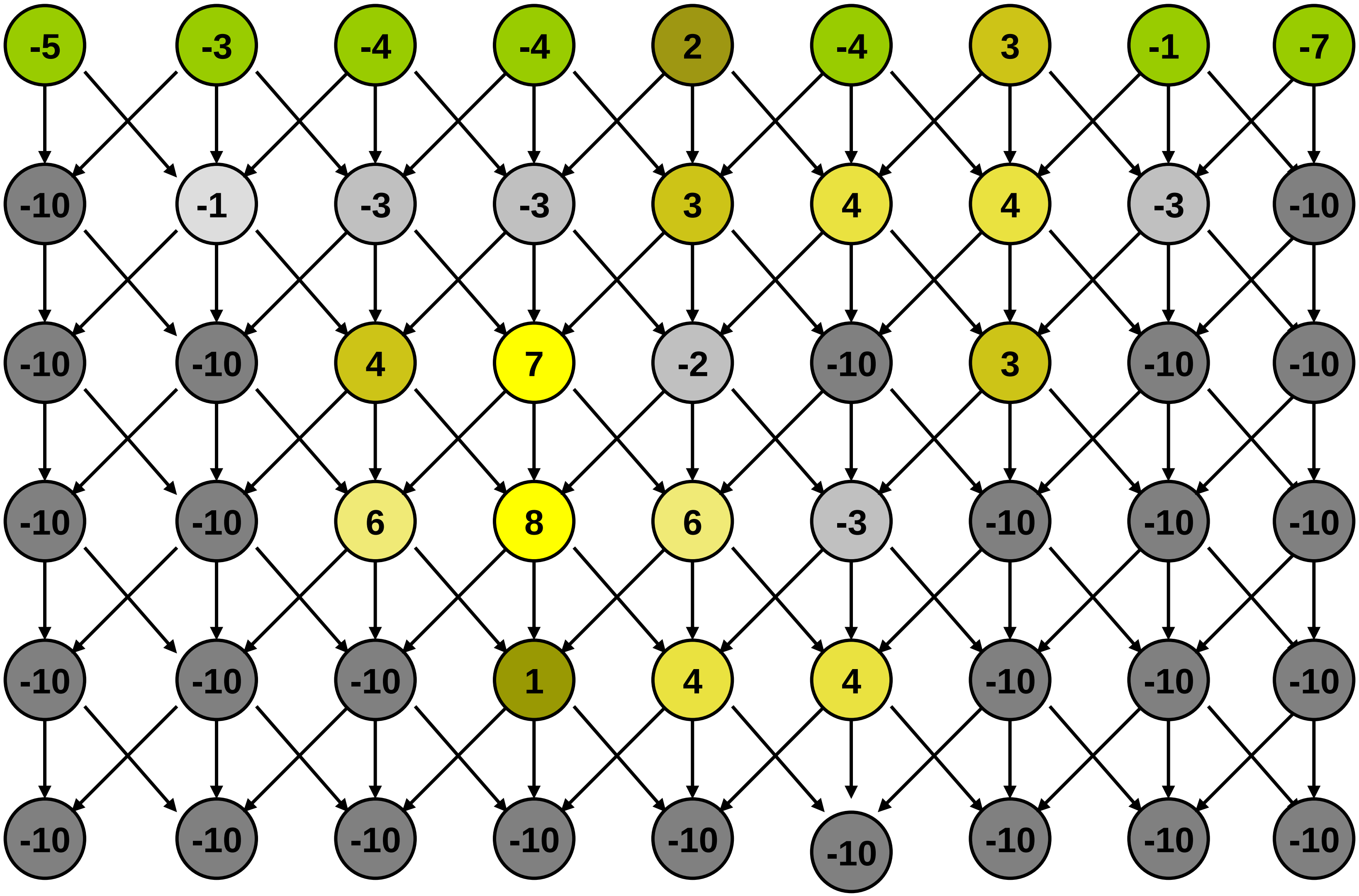
Open Pit Mining

- Each unit of earth has a profit (possibly negative)
- Getting to the ore below the surface requires removing the dirt above
- Test drilling gives reasonable estimates of costs
- Plan an optimal mining operation

Mine Graph

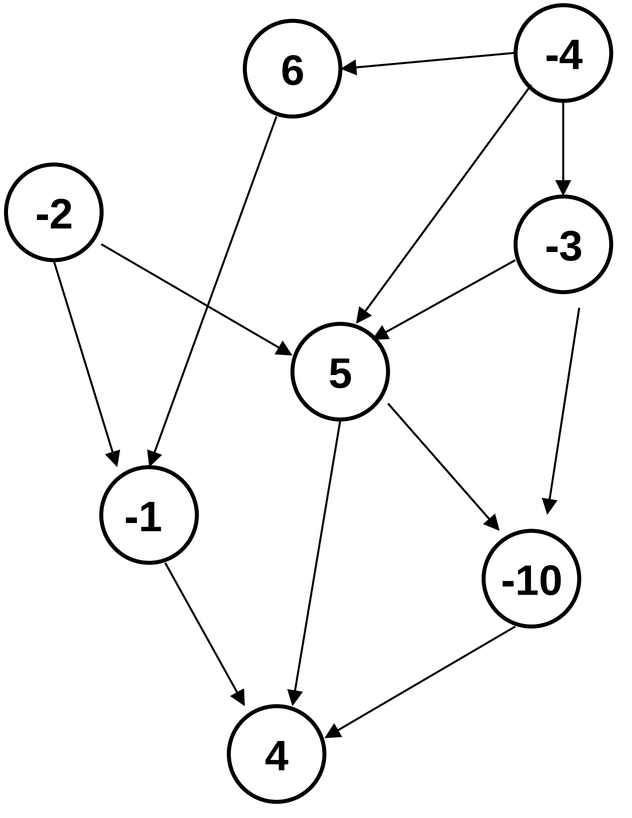


Determine an optimal mine



Generalization

- Precedence graph G=(V,E)
- Each v in V has a profit p(v)
- A set F is *feasible* if when w in F, and (v,w) in E, then v in F.
- Find a feasible set to maximize the profit

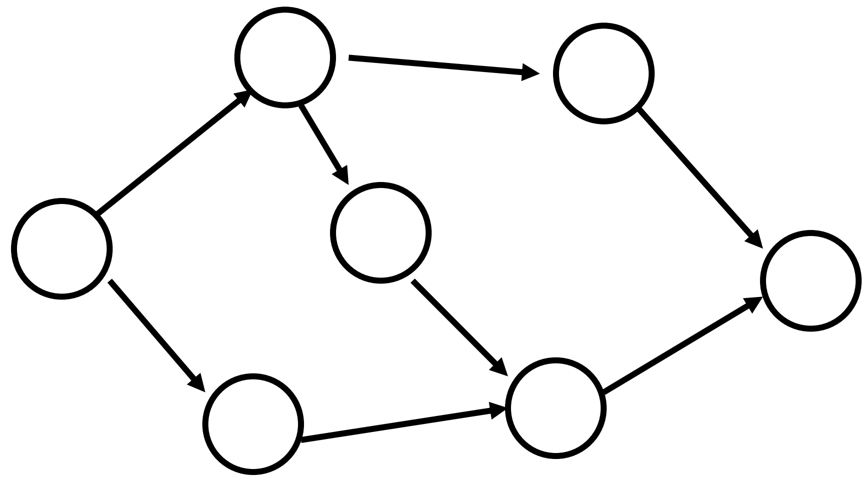


Min cut algorithm for profit maximization

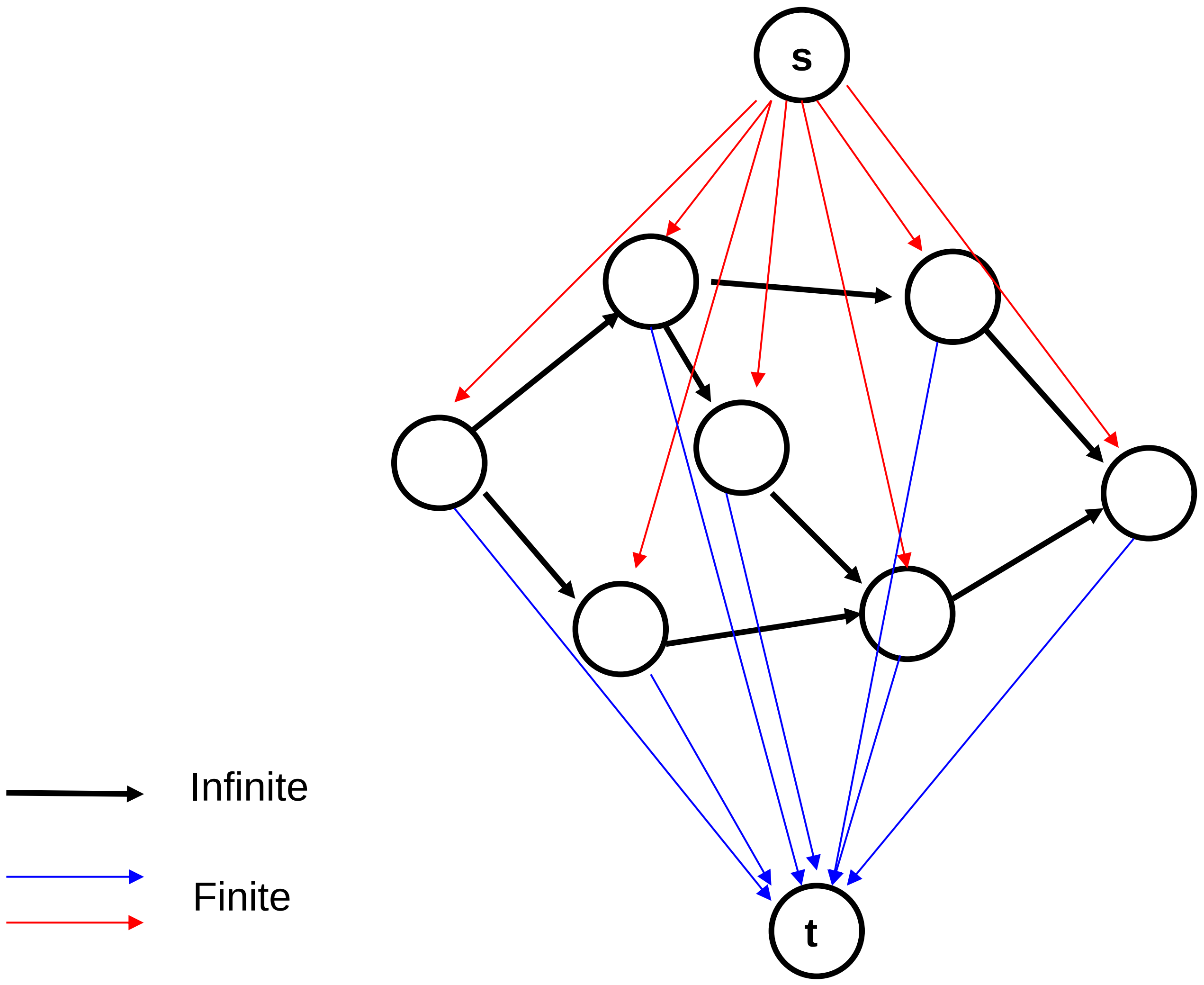
- Construct a flow graph where the minimum cut identifies a feasible set that maximizes profit

Precedence graph construction

- Precedence graph G=(V,E)
- Each edge in E has infinite capacity
- Add vertices s, t
- Each vertex in V is attached to s and t with finite capacity edges

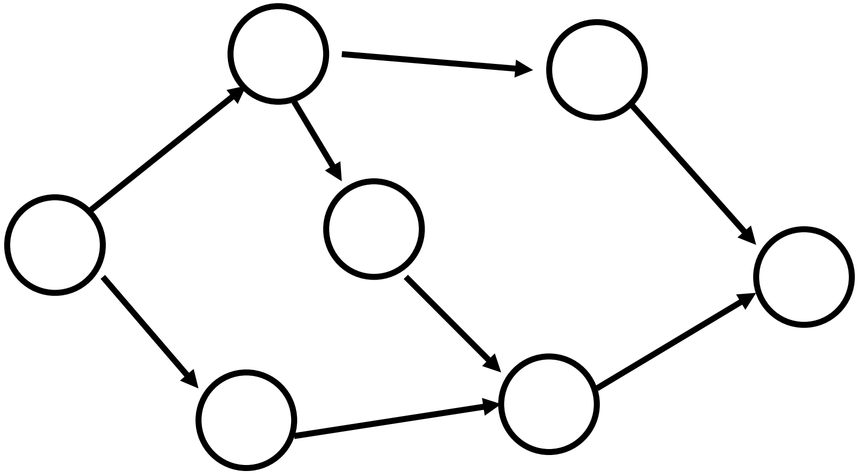


Find a finite value cut with at least two vertices on each side of the cut



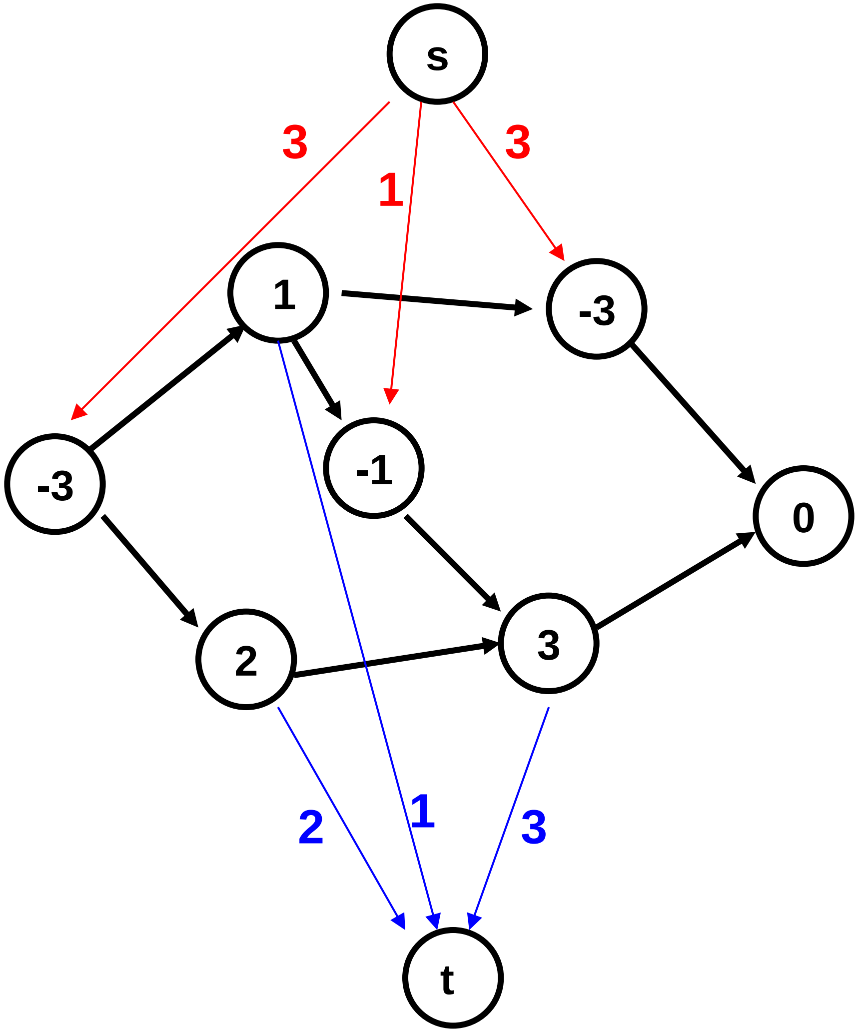
The sink side of a finite cut is a feasible set

- No edges permitted from S to T
- If a vertex is in T, all of its ancestors are in T

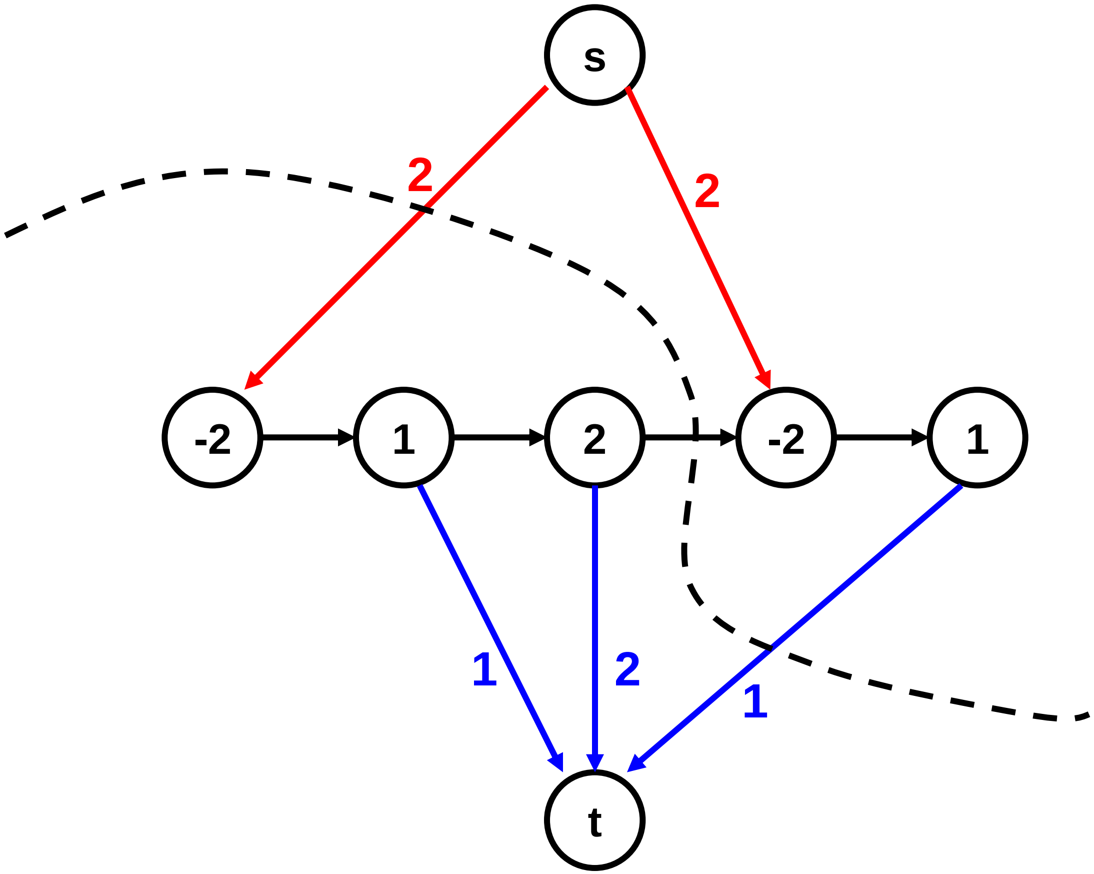


Setting the costs

- If p(v) > 0,
  - cap(v,t) = p(v)
  - cap(s,v) = 0
- If p(v) < 0
  - cap(s,v) = -p(v)
  - cap(v,t) = 0
- If p(v) = 0
  - cap(s,v) = 0
  - cap(v,t) = 0



Minimum cut gives optimal solution
Why?

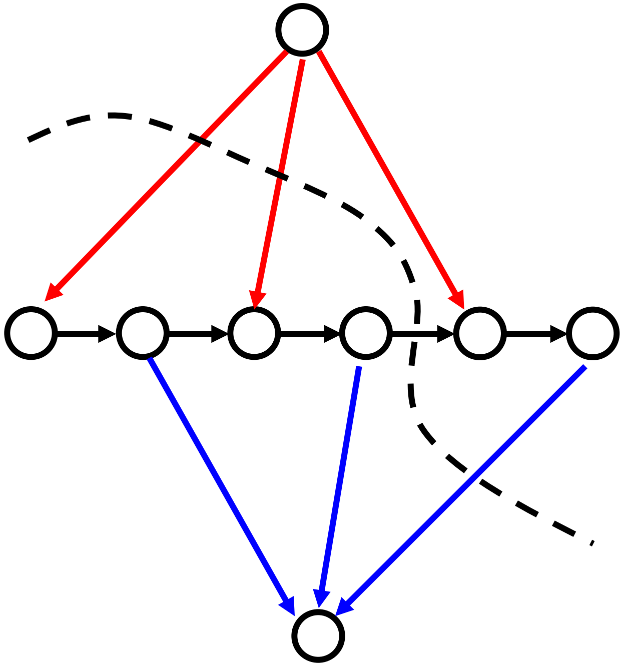


Computing the Profit

- $\text{Cost}(W) = \Sigma_{\{w \text{ in } W;\ p(w) < 0\}} -p(w)$
- $\text{Benefit}(W) = \Sigma_{\{w \text{ in } W;\ p(w) > 0\}}\ p(w)$
- Profit(W) = Benefit(W) – Cost(W)

- Maximum cost and benefit
  - C = Cost(V)
  - B = Benefit(V)

Express Cap(S,T) in terms of B, C, Cost(T), Benefit(T), and Profit(T)



Cap(S,T) = Cost(T) + Ben(S) = Cost(T) + Ben(S) + Ben(T) – Ben(T)

= B + Cost(T) – Ben(T) = B – Profit(T)