COMP 251

Algorithms & Data Structures (Winter 2022)

Graphs – Flow Network 2

School of Computer Science
McGill University

Slides of (Comp321 ,2021), Langer (2014), Kleinberg & Tardos, 2005 & Cormen et al., 2009, Jaehyun Park' slides CS 97SI, Top-coder tutorials, T-414-AFLV Course, Programming Challenges books, slides from D. Plaisted (UNC) and Comp251-Fall McGill.

Announcements

Outline

- Graphs.
  - Introduction.
  - Topological Sort. / Strong Connected Components
  - Network Flow 1.
    - Introduction
    - Ford-Fulkerson
  - Network Flow 2.
    - Min-cuts
  - Shortest Path.
  - Minimum Spanning Trees.
  - Bipartite Graphs.

Flow Network

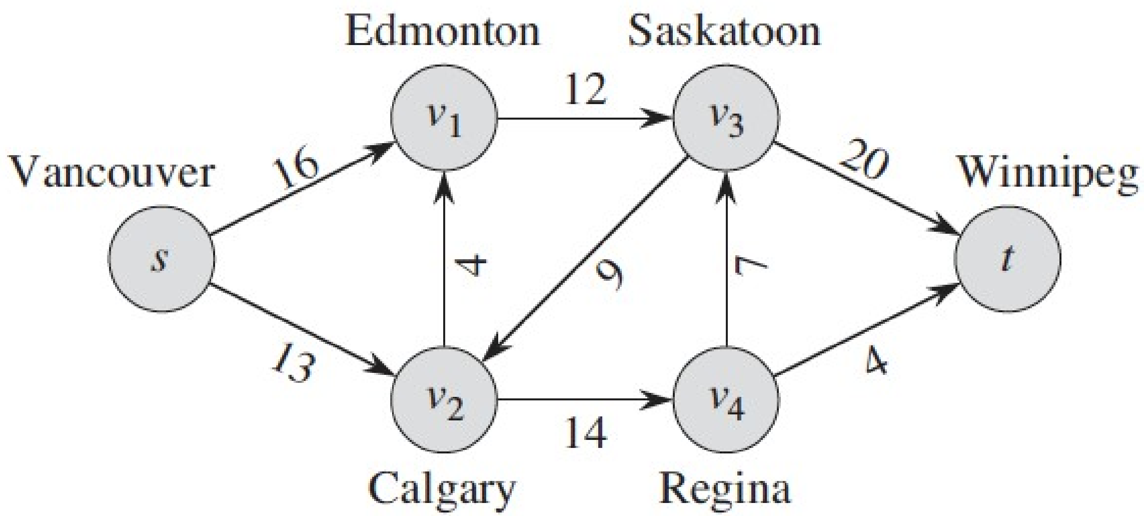
$G = (V, E)$ directed.

Each edge (u, v) has a ***capacity*** $c(u, v) \geq 0$.

If $(u,v) \notin E$, then $c(u,v) = 0$.

Source vertex s, ***sink*** vertex t, assume $s \leadsto v \leadsto t$ for all $v \in V$.

c(u,v) is a non-negative integer.
If $(u,v) \in E$, then $(u,v) \notin E$.
No incoming edges in source.
No outcoming edges in sink.



Ford-Fulkerson algorithm - correctness

Claim: The Ford-Fulkerson algorithm terminates. O($C \cdot |E|$)

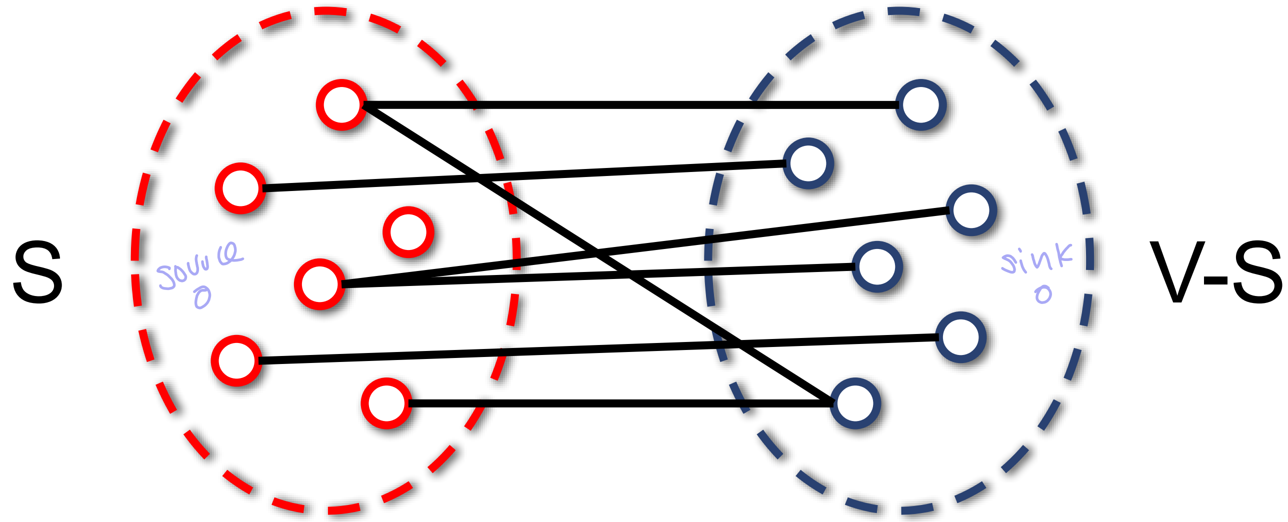
- The capacities and flows are strictly positive integers.
- The sum of capacities leaving s is finite.
- Bottleneck values β are strictly positive integers.
- The flow increase by β after each iteration of the loop.

```
f ←0
G_f←G
while (there is a s-t path in G_f) {
      f.augment(P)
      update G_f based on new f
}
```

- The flow is an increasing sequence of integers that is bounded.
- How do we know that when the algorithm terminates, we have actually found a maximum flow?
  - The max-flow min-cut theorem, tells us that a flow is maximum if and only if its residual network contains no augmenting path.
    - We must first explore the notion of a cut of a flow network.

Cuts

A graph cut is a partition of the graph vertices into two disjoint sets.



The crossing edges from S to V-S are { (u,v) | u ∈ S, v ∈ V-S }, also called the cut set.

Cuts in flow networks

Definition: An s-t cut of a flow network is a cut A, B such that:

- and $s \in A$ and $t \in B$
- $A \cup B = V$
- $A \cap B = \emptyset$

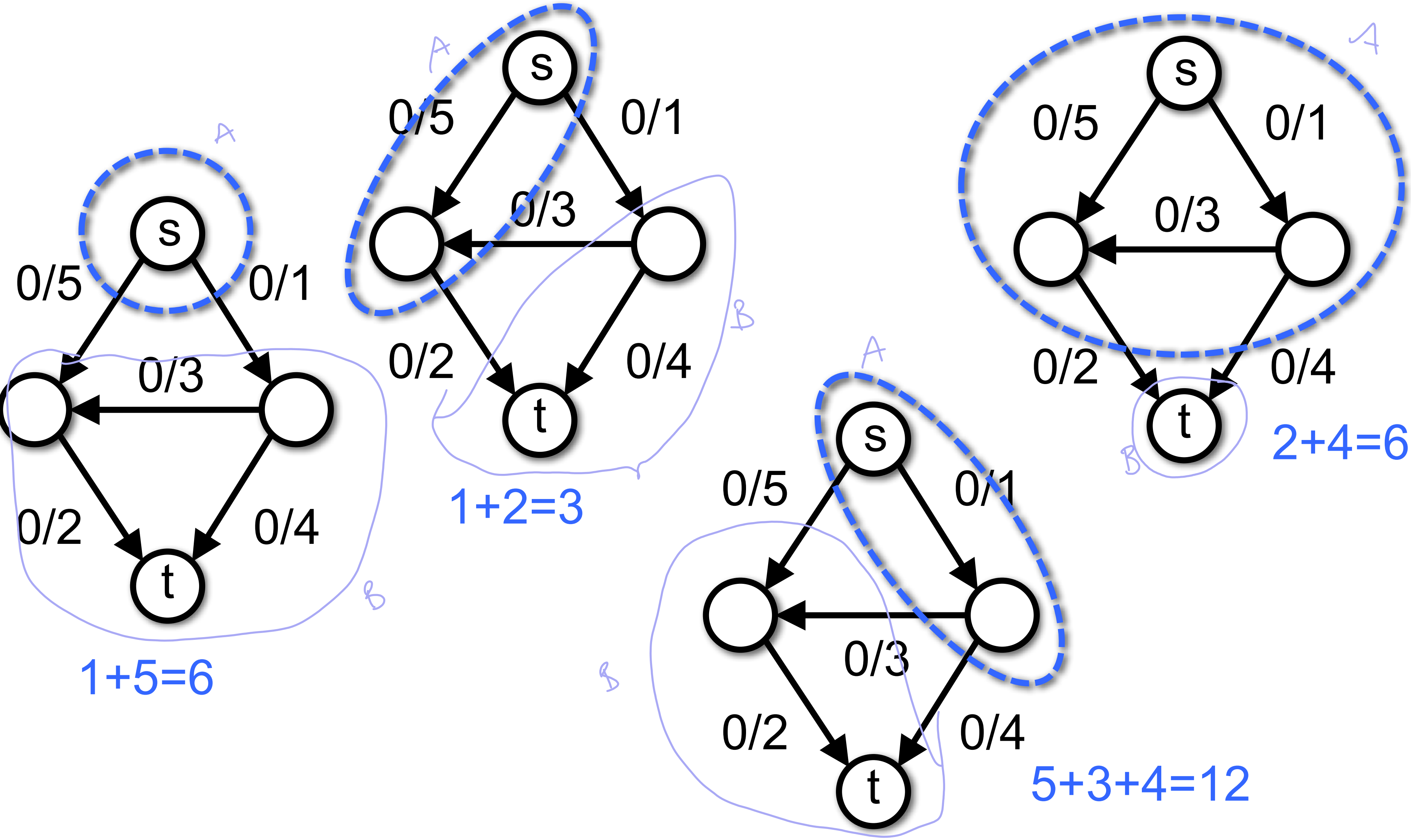
s = source
t = sink

Notation: We write **cut(A,B)** the set of edges from A to B.

Definition: The capacity of an s-t cut is $\sum_{e \in cut(A,B)} c(e)$

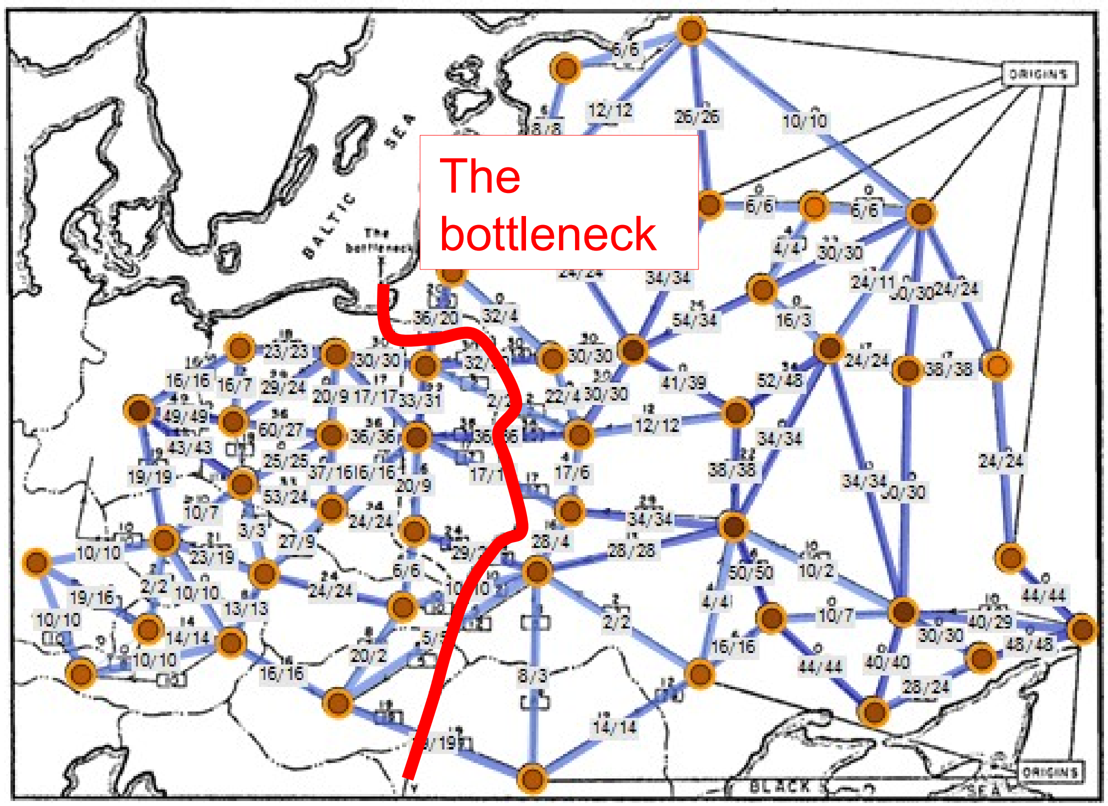
Sum of capacity of edges that go from $A \to B$

Cuts in flow networks



Minimum cut problem

- The minimum cut problem is to compute an (s, t)-cut whose capacity is as small as possible.
  - Intuitively, the minimum cut is the cheapest way to disrupt all flow from s to t.



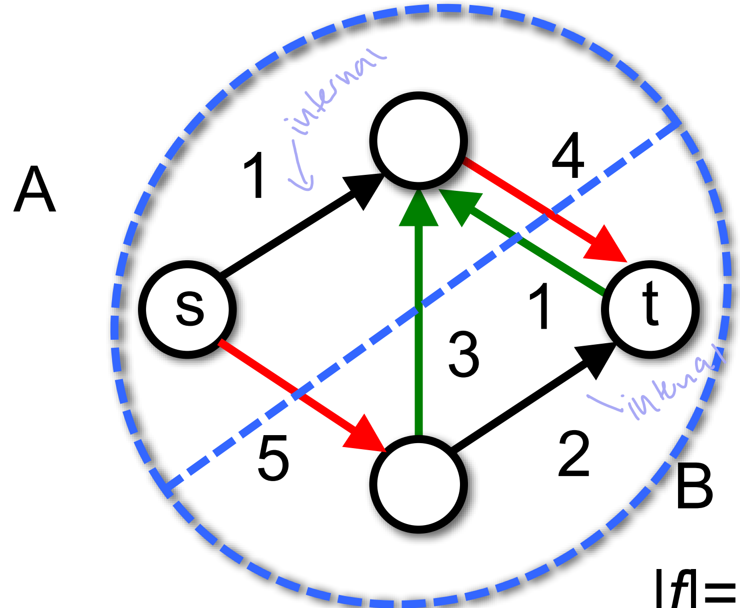
How to cut supplies!

Flow through a cut

Claim: Given a flow network. Let f be a flow and A, B be a s-t cut. Then, the net flow is defined to be:

$$|f| = \underbrace{\sum_{e \in cut(A,B)} f(e)}_{\text{sum flow } A \to B} - \underbrace{\sum_{e \in cut(B,A)} f(e)}_{\text{sum flow } B \to A}$$

Notation: $\boxed{|f| = f^{out}(A) - f^{in}(A)}$



$|f| = 9 - 4 = 5$

Flow through a cut – Upper bound

Claim: For any network flow *f*, and any s-t cut(A,B)

$$|f| \le \sum_{e \in cut(A,B)} c(e)$$

c(e)

$e \in cut(A,B)$

flow ≤ sum capacities A → B

upper bound

Proof:

$$|f| = f^{out}(A) - f^{in}(A)$$
$$\le f^{out}(A)$$
$$\le \sum_{e \in cut(A,B)} c(e)$$

- The asymmetry between the definitions of flow and capacity of a cut is intentional and important.
- Right-hand side is independent of any particular flow f
- The value of every flow is upper-bounded by the capacity of every cut
- The value of a maximum flow in a network is bounded from above by the capacity of a minimum cut of the network.

Flow through a cut – Observations

- Some cuts have greater capacities than others.
- Some flows are greater than others.
- **But every flow must be ≤ capacity of every s-t cut.**
  - if we exhibit any s-t cut in G of some value $c*$, we know that there cannot be an s-t flow in G of value greater than $c*$.
  - Conversely, if we exhibit any s-t flow in G of some value $v*$, we know that there cannot be an s-t cut in G of value less than $v*$.
- Thus, the value of the maximum flow is less than capacity of the minimum cut.
  - The max-flow min-cut theorem, says that the value of a maximum flow is in fact equal to the capacity of a minimum cut.

Value of flow in Ford-Fulkerson

Theorem (Max-flow min-cut theorem)

If f is a flow in a flow network $G = (V,E)$ with source s and sink t , then the following conditions are equivalent:

1. f is a maximum flow in G.

2. The residual network G_f contains no augmenting paths.

3. $|f| = c(S,T)$ for some cut (S, T) of G.

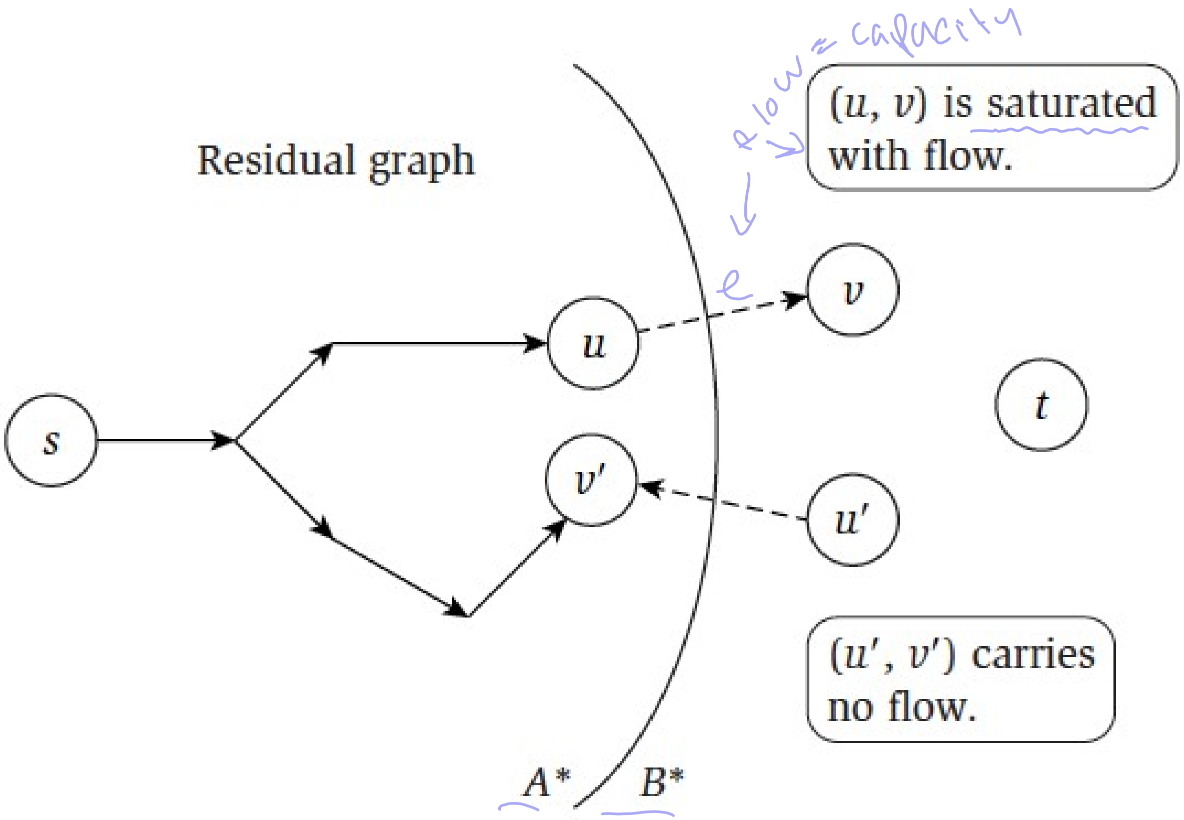
source sink

Value of flow in Ford-Fulkerson

- Ford-Fulkerson terminates when there is no augmenting path in the residual graph G_f. no path $s \to T$
- Let $\boxed{A}$ be the set of vertices reachable from s in G_f, and B=V-A.
  ↳ split vertices into two groups
- A,B is a s-t cut in G_f. disjoint set, $A \cup B = V$, $A \cap B = \emptyset$, $s \in A$, $t \in B$
- A,B is an s-t cut in G (G and G_f have the same vertices).
  max flow
- $|f|=f^{out}(A)-f^{in}(A)$
- We want to show: $$|f| = \sum_{e \in cut(A,B)} c(e)$$
- And in particular:

① $$f^{out}(A) = \sum_{e \in cut(A,B)} c(e)$$ ② $$f^{in}(A) = 0$$

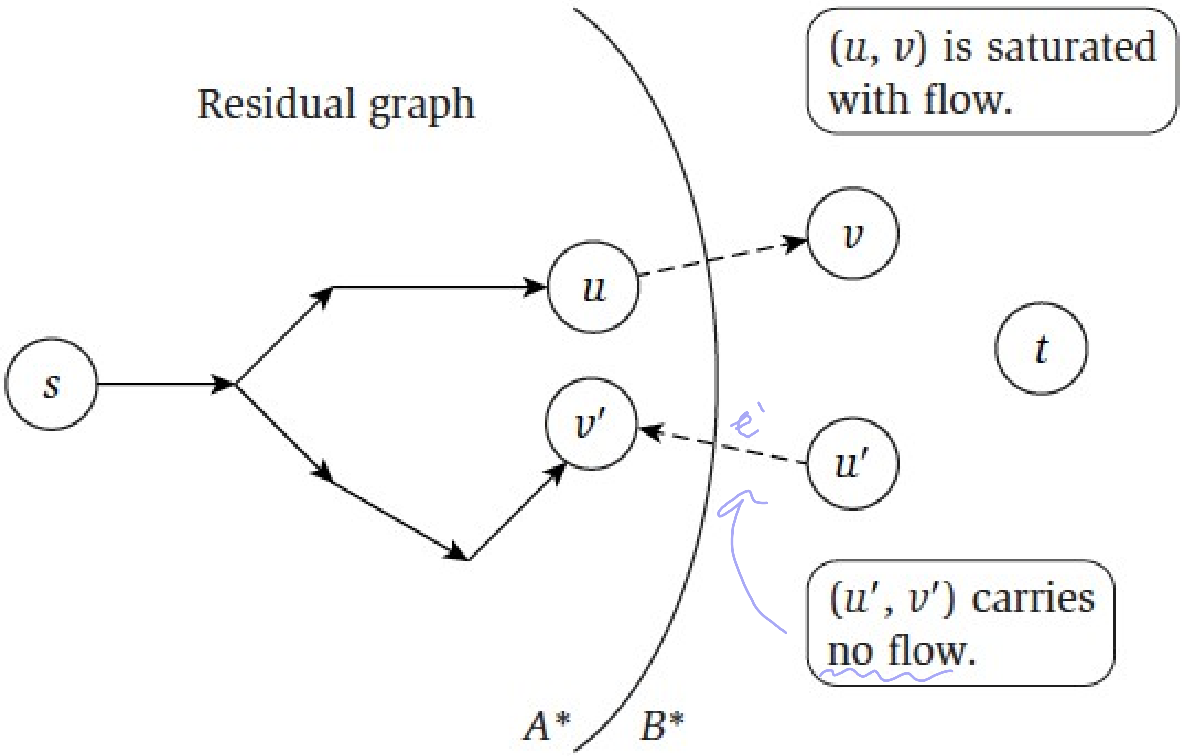
Value of flow in Ford-Fulkerson



Suppose that e = (u, v) is an edge in G for which u ∈ A∗ and v ∈ B∗. We claim that $f(e) = c_e$. For if not, e would be a forward edge in the residual graph G_f, and since u ∈ A∗, there is an s-u path in G_f; appending e to this path, we would obtain an s-v path in G_f, contradicting our assumption that v ∈ B∗.

(A∗, B∗) is indeed an s-t cut. It is clearly a partition of V. t A∗ by the assumption that there is no s-t path in the residual graph; hence t ∈ B∗ as desired.

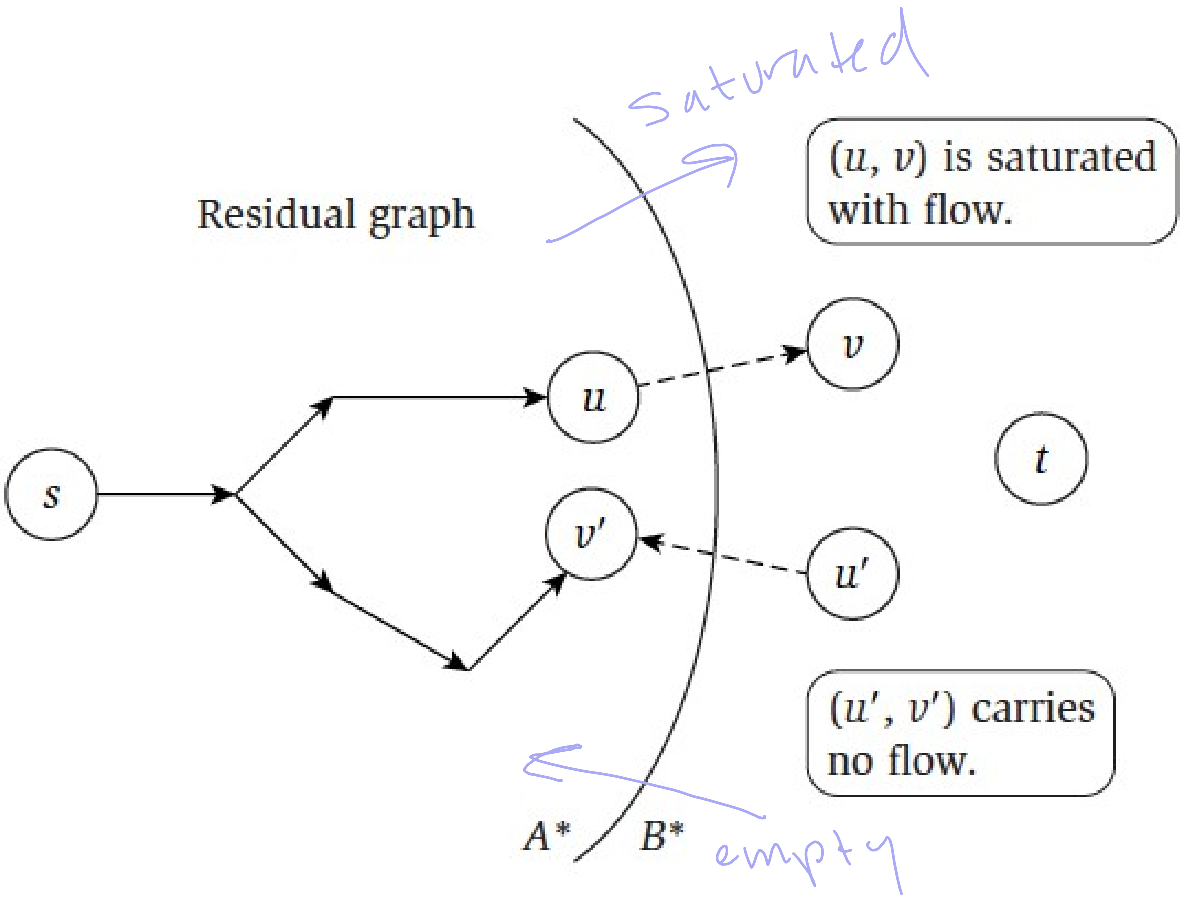
Value of flow in Ford-Fulkerson



Suppose that e’ = (u’, v’) is an edge in G for which u’ ∈ B∗ and v’ ∈ A∗. We claim that f(e’) = 0. For if not, e’ would give rise to a backward edge e’’ = (v’, u’) in the residual graph G_f , and since v’ ∈ A∗, there is an s-v’ path in G_f; appending e’’ to this path, we would obtain an s-u’ path in Gf , contradicting our assumption that u’ ∈ B∗.

(A∗, B∗) is indeed an s-t cut. It is clearly a partition of V. t A∗ by the assumption that there is no s-t path in the residual graph; hence t ∈ B∗ as desired.

Value of flow in Ford-Fulkerson



So all edges out of A∗ are completely saturated with flow, while all edges into A∗ are completely unused.

Max flow = Min cut

- Ford-Fulkerson terminates when there is no path s-t in the residual graph G_f
- This defines a cut in A,B in G (A = nodes reachable from s)
- $$|f| = f^{out}(A) - f^{in}(A)$$
  $$= \sum_{e \in cut(A,B)} f(e) - \sum_{e \in cut(B,A)} f(e)$$
- Ford-Fulkerson flow = $\sum_{e \in cut(A,B)} c(e) - 0$

  = capacity of cut(A,B)

Computing the min-cut

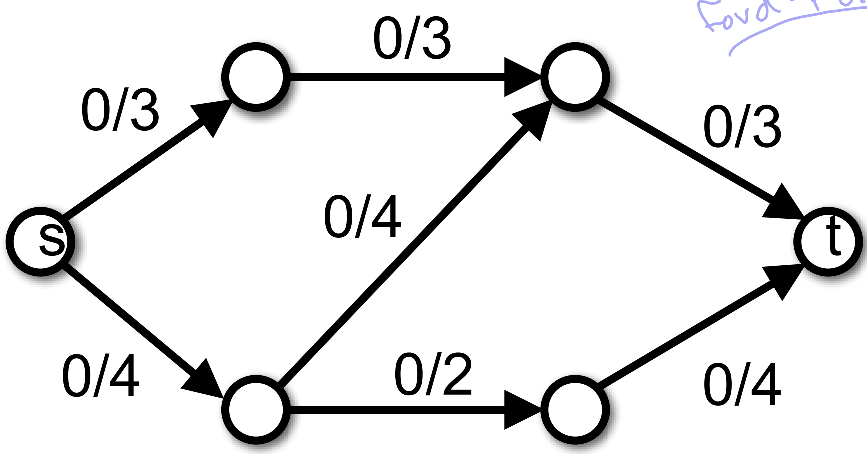
Q: Given a flow network, how can we compute a minimum cut?

Answer:

- Run Ford-Fulkerson to compute a maximum flow (it gives us G_f)
- Run BFS or DFS of s.
- The reachable vertices define the set A for the cut

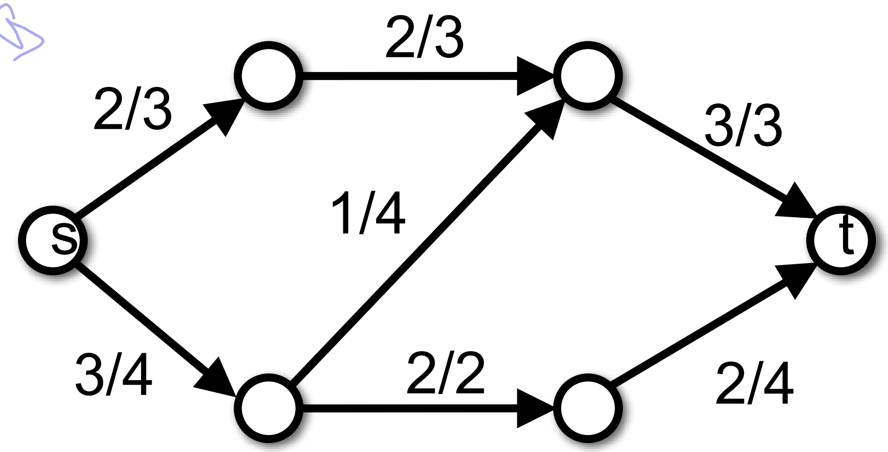
Computing the min-cut – Ford-fulkerson

(1) Initial flow net G



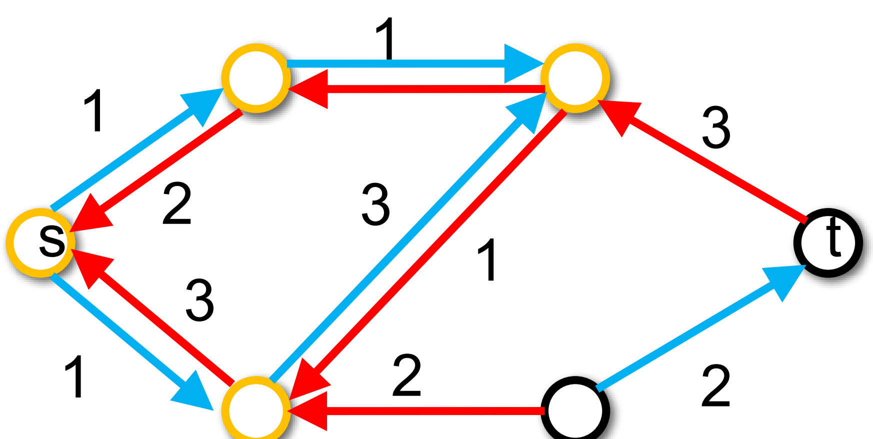
Ford-Fulkerson

(2) Compute max flow (FF)

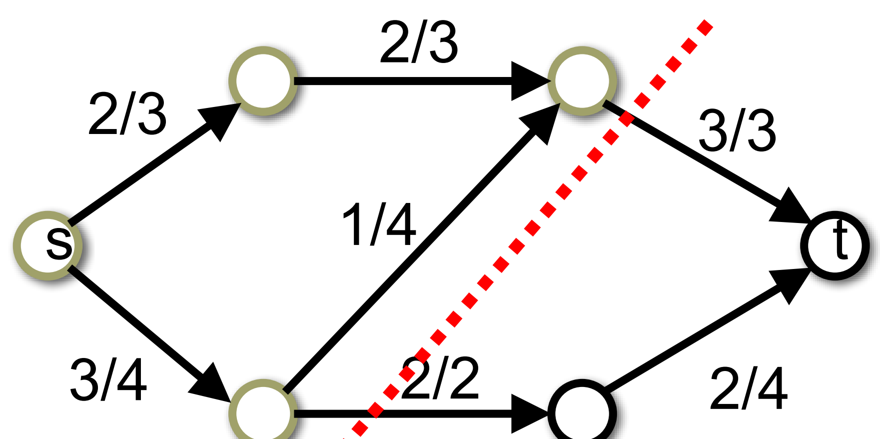


(3) Compute G_f and vertices accessible from s

DFS



(4) Vertices accessible from s in G_f determine the min cut



Outline

- Graphs.
  - Introduction.
  - Topological Sort. / Strong Connected Components
  - Network Flow 1.
    - Introduction
    - Ford-Fulkerson
  - Network Flow 2.
    - Min-cuts
  - Shortest Path.
  - Minimum Spanning Trees.
  - Bipartite Graphs.


time for questions