Design and Analysis of Algorithms II: Flow Network Lecture 5

Varun Aravapalli

October 5th, 2023

Contents

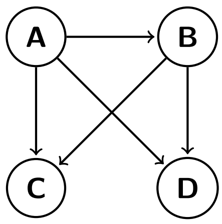
1 Directed Acyclic Graph (DAG)

1.1 Recap:

A Directed Graph $G = (V, E)$ where $V = |V|$ number of vertices (finite), $E = |E|$ number of edges (finite)

Example:

- $V = \{A, B, C, D\}$
- $E = \{AB, AC, AD, BC, BD\}$



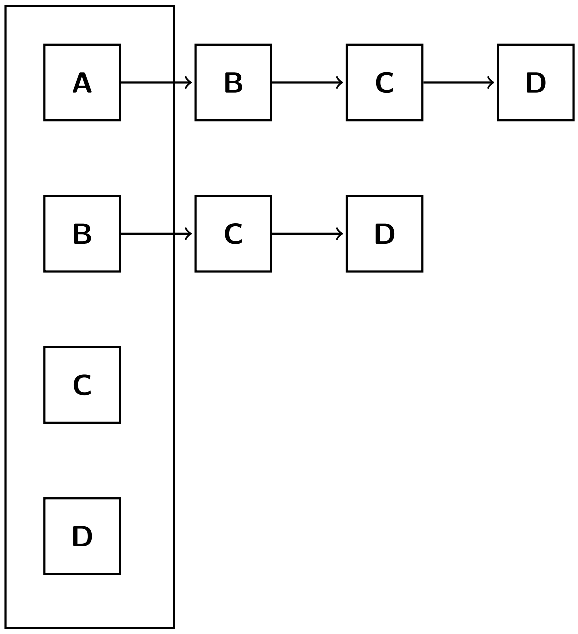
1.2 Representation

We can represent a DAG in two ways

1. Adjacency Matrix:

	A	B	C	D
A	0	1	1	1
B	0	0	1	1
C	0	0	0	0
D	0	0	0	0

2. Adjacency List:



2 Flow Network

2.1 Definition

Given a Directed Graph (G) (V,E), such that **s,t** $\in$ V

- **s** = source node (No Incoming Edges)
- **t** = sink node (No Outgoing Edges)

2.1.1 Assume Strongly Connected

Assume $\forall v \in V$ is reachable from S and can reach t.

- $\forall v \in V : s \rightsquigarrow v \rightsquigarrow t$.
- $v \rightsquigarrow w$: $\exists$ a directed path from v to w in G.

2.1.2 Assume No Cycles

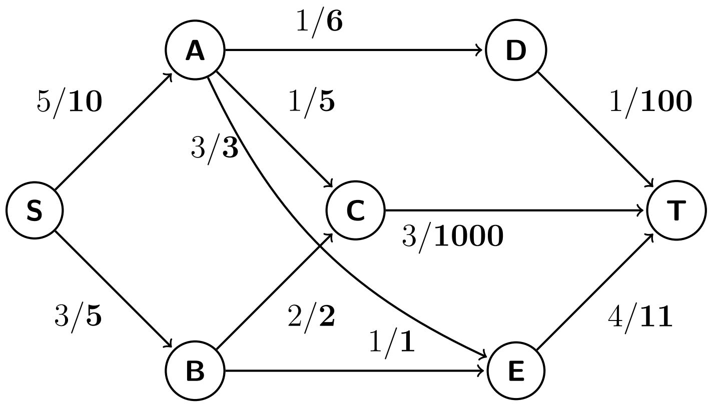
Assume between any two vertices there exists at most one edge, such that **there are no cycles**

2.1.3 Capacity Law

The capacity $\forall e \in E$: is a positive real number greater than 0

$$c : E \rightarrow \mathbb{R}^+ \quad [c(e) > 0, \forall e \in E]$$

2.2 Flow Diagram



BOLD = *Capacity*
NORMAL = *Flow*

2.2.1 Flow Laws

Flow: $f : E \rightarrow \mathbb{R}^+$
Value of Flow: $\sum$ f(s,v) = Sum of Flow of Each Edge [(s,v) $\exists$ E]

1. For any edge, the flow has to be ≥ 0 and less than capacity

$$0 \leq f(e) \leq c(e) \tag{1}$$

2. For any $V \neq s, t$ all incoming flow = outgoing flow

$$\sum f(u,v) = \sum f(v,u) \tag{2}$$

3. Flow = 0 is always valid, so we are never working with an empty set

3 Problem Max Flow

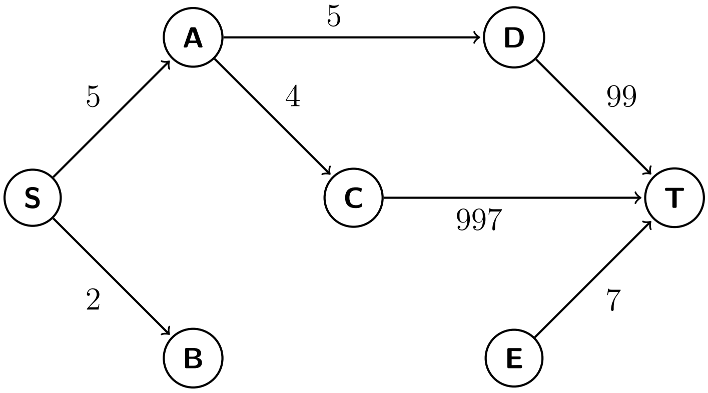
Given **G** (Graph), **C** (capacity), find flow **f** that maximizes $|f|$

3.1 Ford-Fulkerson (FF) Method

Note: This is not considered an *"Algorithm"* because if it doesn't work, it may never stop and an *"Algorithm"*must terminate on all inputs

3.1.1 Example Residual Network

Given a **G** (graph), **s**, **t**, **c** and a ***Valid Flow*** $G_f \Rightarrow$ "Residual Network".



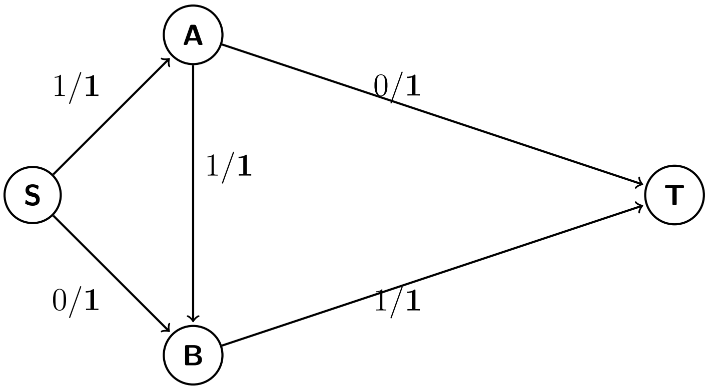
For each capacity that isn't at max, we write down the remainder in G_f
c_f(u,v) = c(u,v) - f(u,v) **if** f(u,v) < c(u,v)

3.1.2 Solving Ford-Fulkersons

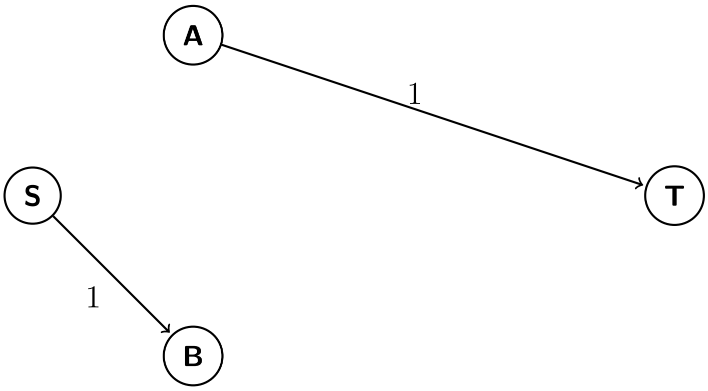
1. Start f = 0
2. Build G_f (residual network)
3. Find augmenting path (We stop if we don't find an augmenting path)
4. Update f, options:
   (a) Fat Pipe: "Max Improvement"
   (b) Short Pipe: "Minimum Edges"
5. Repeat

3.1.3 Solving Ford-Fulkersons Part 2.

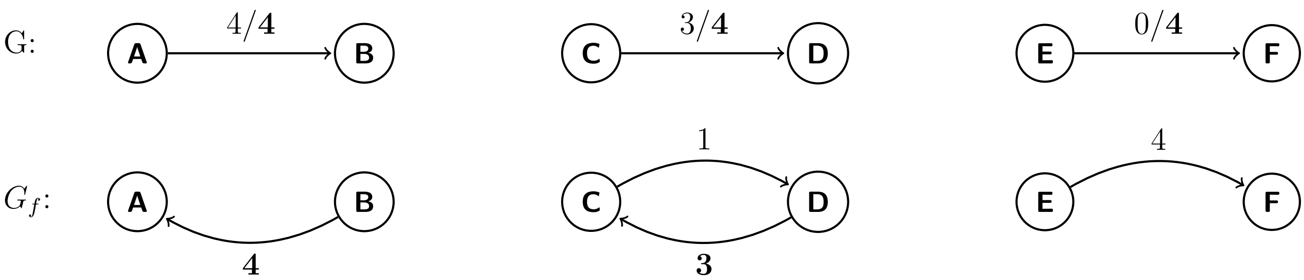
In this example, there are **no more augmented paths**, but the max flow isn't achieved
Flow Network:



Residual Network:

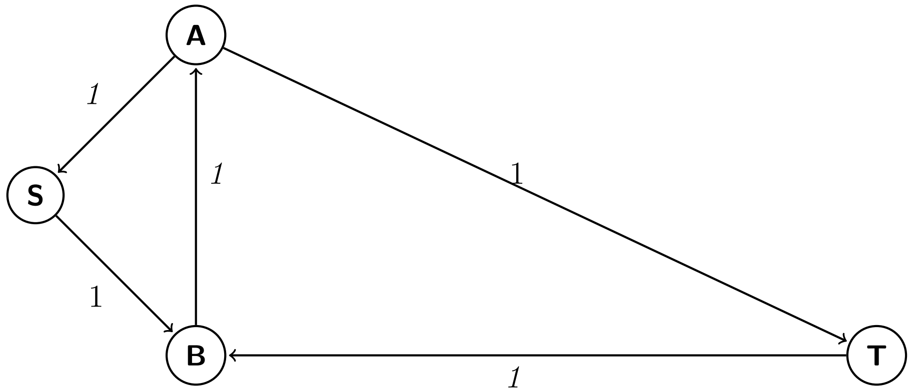


1. c_f = c(u,v) - f(u,v) **if** f(u,v) < c(u,v)
2. c_f = f(u,v) **if** f(u,v) > 0



Reverse Edges

3.1.4 New Residual Network:



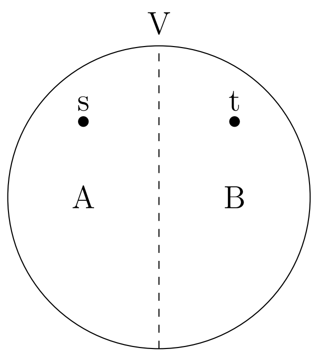
GOAL: If FF terminates, the flow is Maximum

3.2 Proof of FF

3.2.1 s-t cut in G:

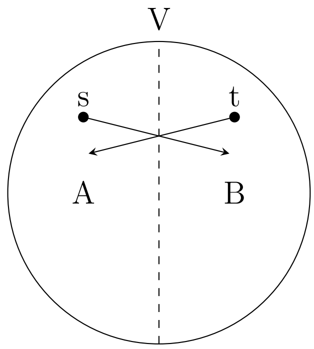
Let $A, B \in \mathcal{P}(V)$ such that,

1. $A \cup B = V$
2. $A \cap B = \emptyset$
3. $s \in A, \quad t \in B$



3.2.2 Flow Across cut (A,B) : f(A,B)

Flow across cut(A,B) : f(A,B) = Net flow, aggregate of positive and negative flow

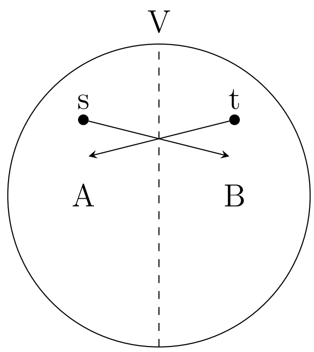


$$\sum f(u,v) - \sum f(u,v)$$

u ∈ A u ∈ B
v ∈ B v ∈ A
(u,v) ∈ E

3.2.3 Capacity Across cut (A,B) : c(A,B)

All positive, no negative and backwards measured



$$\sum c(A,B) = \sum c(u,v)$$

u ∈ A
v ∈ B
(u,v) ∈ E

Fact: For any f, c, $s - t$ cut (A, B):

1. $f(A,B) = |f|$
2. $f(A,B) \leq c(A,B)$

3.3 Max-Flow, Min-Cut Theorem:

For any flow f, the following hold:

$$|f| = f(A, B) \leq c(A, B)$$

$$\max |f| \leq \min c(A, B)$$

$$f = (A, B).$$

Basically, if you find the min cut, you have the max flow.

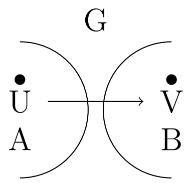
3.4 Claim: If FF stops you can find a cut (A,B) such that c(A,B) = max flow

1. Build G_f
2. Let A = All vertices reachable from S in G_f
3. B = V -A

There is no way to reach t from s (Augmented Path) because we claimed FF stopped

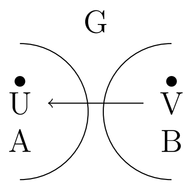
3.4.1 Full forward capacity

Case 1: (u,v) $\in$ E(G)

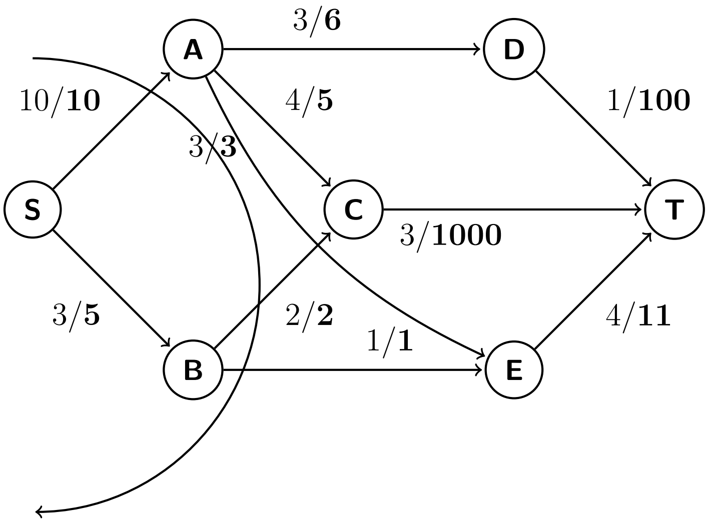


3.4.2 Backward Edge

Case 2: (v,u) $\in$ E(G)



3.4.3 Example of Cut



Maximum Capacity Cut because 1/1, 2/2, 10/10 exist so no more flow can go through the network