CSE 100: Algorithm Design and Analysis
Chapter 26: Maximum Flow

Sungjin Im

University of California, Merced

Last Update: 4-27-2023

Flow networks and flows

Flow network: a directed graph $G = (V, E)$ with each edge (u, v) having capacity $c(u, v) \geq 0$. (for simplicity, assume that all capacities are integers.)

- has two special vertices: source $s \in V$ and sink $t \in V$.

Flow: a function $f : V \times V \to R$ satisfying

- Capacity constraint: for all $u, v \in V$, we have $0 \leq f(u, v) \leq c(u, v)$.
- Flow conservation: for all $u \in V \setminus \{s, t\}$, $\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$.

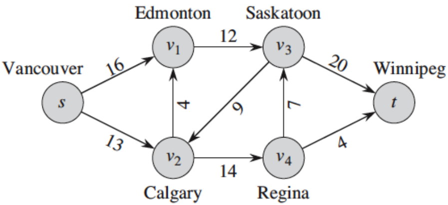
The value $|f|$ of flow f is defined as

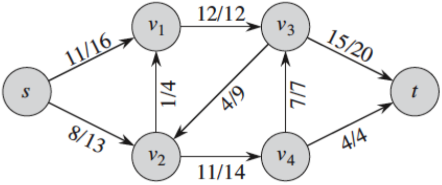
$$\sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

The maximum-flow problem asks to find a flow of the maximum value.

Flow networks and flows

Example

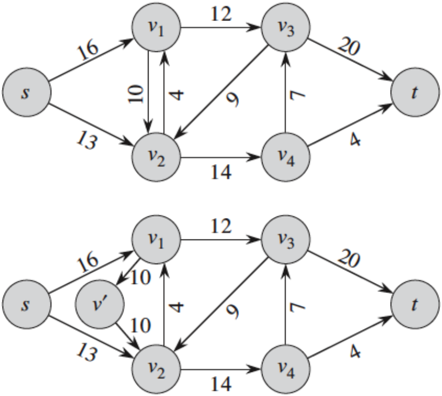




What is the flow value?

Flow networks

No antiparallel edges wlog



One can assume wlog that the given flow network has no antiparallel edges

Antiparallel edges: (u, v) and (v, u) for some $u \neq v \in V$.

The Ford-Fulkerson Method

FORD-FULKERSON-METHOD(G, s, t)

```
initialize flow f to 0
while there exists an augmenting path p in the residual network G_f
    augment flow f along p
return f
```

Encompasses many max-flow algorithms based on residual networks, augmenting paths, and cuts.

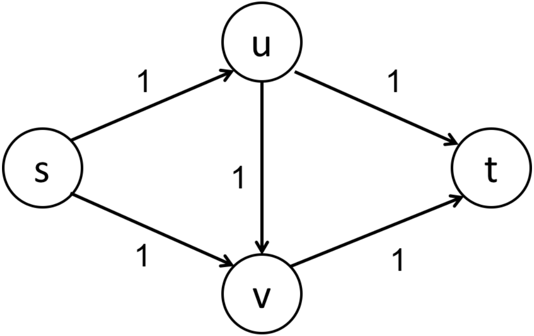
The Ford-Fulkerson Method

We use residual networks and augmenting paths to increase the flow value.

We use (min) cut to verify whether the current flow is a flow of the maximum value.

Minimum Cut

First question: How do we know whether the current flow is a maximum flow?



Minimum Cut

First question: How do we know whether the current flow is a maximum flow?

A cut of flow network $G = (V, E)$ is $(S, T = V \setminus S)$ such that $s \in S$ and $t \in T$
(In the literature, this is often called an s-t cut, distinguished from the cut definition in Ch 23).
The capacity of cut: $c(S, T) := \sum_{u \in S} \sum_{v \in T} c(u, v)$.
A minimum cut of a network is a cut of the min capacity over all cuts.

The net flow across cut (S, T):
$f(S, T) := \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)$.

Lemma (Lemma 26.4 + Corollary 26.5)
$|f| = f(S, T) \leq c(S, T)$ *for any cut* (S, T).

Minimum Cut

First question: How do we know whether the current flow is a maximum flow?

Lemma (Lemma 26.4 + Corollary 26.5)

$|f| = f(S, T) \leq c(S, T)$ *for any cut* (S, T).

$(\Rightarrow)$ If $f(S, T) = c(S, T)$ for some cut (S, T), then f is max flow.
$(\Rightarrow)$ Max flow (value) $\leq$ min cut (capacity).

Minimum Cut

First question: How do we know whether the current flow is a maximum flow?

Theorem (a simpler version of Theorem 26.6: Max-flow Min-cut Theorem)

Max flow (value) = min cut (capacity).

We will see why "=" holds true by finding a flow f and cut (S, T) such that $f(S, T) = c(S, T)$.

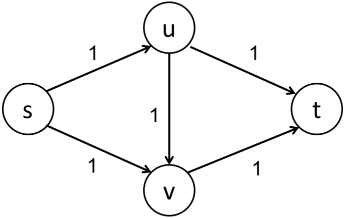
Finding a max flow

Second Question: How do we find a flow f and cut (S, T) such that $f(S, T) = c(S, T)$.

By using residual networks and augmenting paths!

Finding a max flow

Example



Finding a max flow

Residual Network

Given a flow network G and a flow f, the residual network G_f has the following edge capacities:

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \,, \\ f(v,u) & \text{if } (v,u) \in E \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

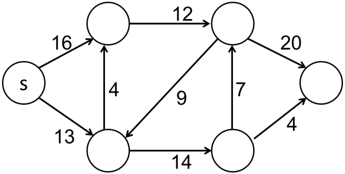
Finding a max flow

Augmenting Paths

Given a flow network G and a flow f, an augmenting path p is a simple path from s to t in the residual network G_f (such that every edge (u, v) on the path p has a positive residual capacity $c_f(u, v)$).

We can send along path p an additional flow of value up to $c_f(p) := \min\{c_f(u, v) \ : \ (u, v) \text{ is on } p\}$, which is called p's residual capacity.

Illustration of FF



Finding a max flow

Correctness of FF

Observation

Suppose all capacities are integers and we use the FF method. Then, whenever we find an augmenting path, we send an additional flow of value at least one, thus increasing the flow value by at least one.

Observation

The FF method terminates. Further, when it does, if all capacities are integers, $f(u, v)$ is an integer for all $(u, v) \in E$, and so is the flow value, $|f|$.

Finding a max flow

Correctness of FF

Lemma

When the FF method terminates, the flow f is a max flow.

Finding a max flow

Correctness of FF

Lemma

When the FF method terminates, the flow f is a max flow.

Proof.

(sketch) Let S denote the vertices reachable from s in G_f using edges of positive residual capacities. Note that $t \notin S$. Then, we show that $|f| = c(S, V \setminus S)$. □

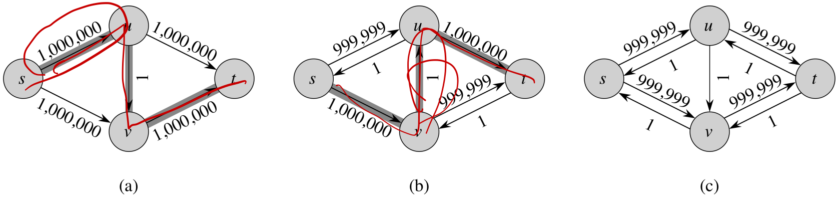
Running Time of FF

FORD-FULKERSON(G, s, t)

for each edge $(u, v) \in G.E$
    $(u, v).f = 0$
while there exists a path p from s to t in the residual network G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is in } p\}$
    for each edge (u, v) in p
        if $(u, v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$

RT: $O(E|f^*|)$
where f^* is a maximum flow.

Running Time of FF



Note: $E|f^*|$ is not polynomial in the input size.

The Edmonds-Karp algorithm

Speeding up the FF method

FORD-FULKERSON(G, s, t)

for each edge $(u, v) \in G.E$
    $(u, v).f = 0$
while there exists a path p from s to t in the residual network G_f
    $c_f(p) = \min\{c_f(u, v) : (u, v) \text{ is in } p\}$
    for each edge (u, v) in p
        if $(u, v) \in E$
            $(u, v).f = (u, v).f + c_f(p)$
        else $(v, u).f = (v, u).f - c_f(p)$

Change to FF: Find the augmenting path p with a BFS.
RT: $O(E^2 V)$; polynomial in the input size!

Applications of Maximum Flow

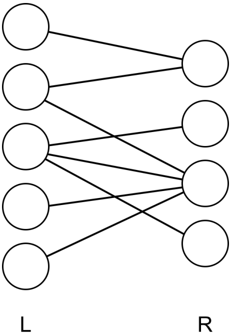
Maximum Bipartite Matching

Input: a bipartite (undirected) graph $G = (V = L \cup R, E)$.

- $L \cap R = \emptyset$; and no edges within L or R.

Output: a maximum matching.

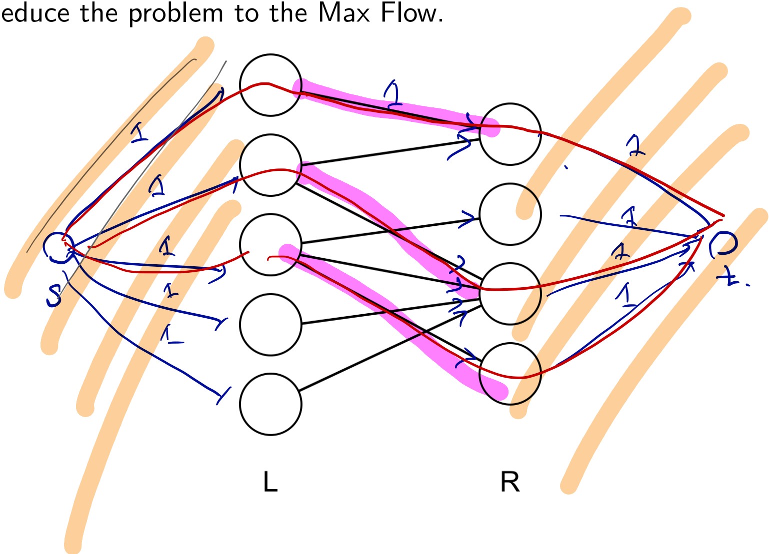
- $M \subseteq E$ is a matching if no two (different) edges in M are adjacent, i.e., have a common end point.
- Maximum matching: a matching of the maximum cardinality.



Applications of Maximum Flow

Maximum Bipartite Matching

Reduce the problem to the Max Flow.

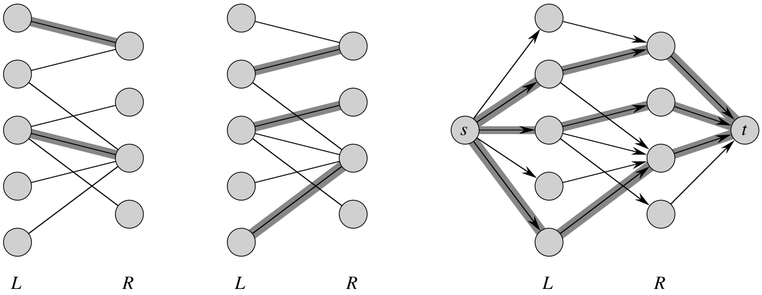


Do not forget to specify edge capacities.

Applications of Maximum Flow

Maximum Bipartite Matching

Reduce the problem to the Max Flow.



(a) (b) (c)