Algorithm Design, Analysis & Complexity

Lecture 6 - Flow Networks

Koushik Pal

University of Toronto

June 8, 2021

Flow Network

Definition

A **flow network** $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a nonnegative **capacity** $c(u, v) \geq 0$.

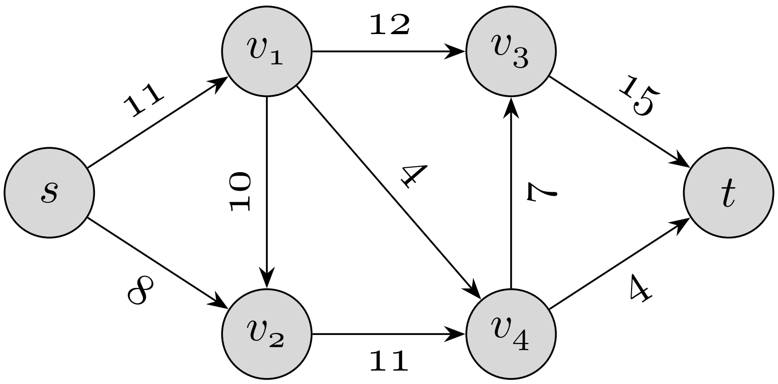
We further require that $(u, v) \in E \implies (v, u) \notin E$.

If $(u, v) \notin E$, for convenience, we define $c(u, v) = 0$.

There are two special vertices — source s and sink t. Every vertex $v \in V$ lies on some path from s to t.

The graph is therefore connected and $|E| \geq |V| - 1$ (since, every vertex other than s has at least one edge entering it).

Example



Flow

Definition

A **flow** in G is a real-valued function $f : V \times V \to \mathbb{R}$ satisfying the following two properties:

Capacity Constraint. For all $u, v \in V$, we require

$$0 \leq f(u,v) \leq c(u,v).$$

Flow Conservation. For all $u \in V \setminus \{s,t\}$, we require

$$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v).$$

When $(u,v) \notin E$, there is no flow from u to v, and $f(u,v) = 0$.

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

$$|f| := \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s).$$

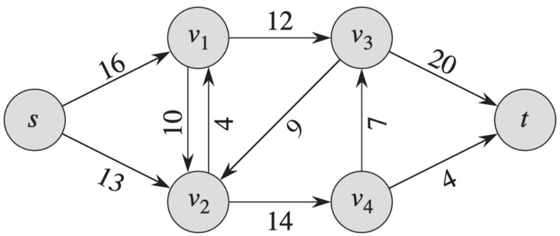
Maximum-Flow Problem

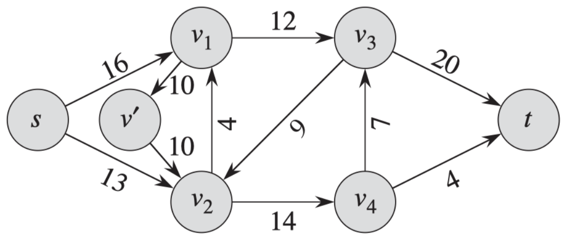
Problem (Maximum-Flow Problem)

Given a flow network $G = (V, E)$ with source s and sink t, find a flow of maximum value.

Variations

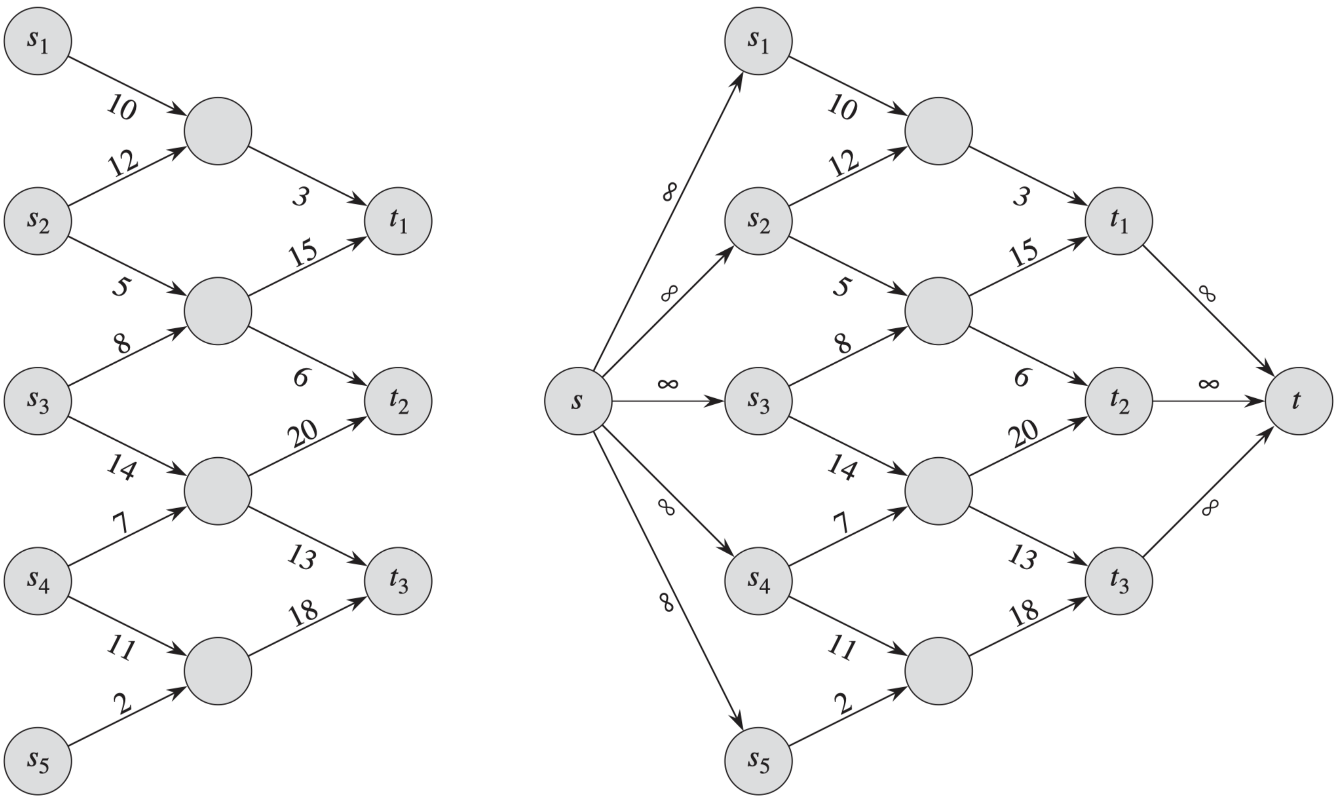
- Modeling anti-parallel edges





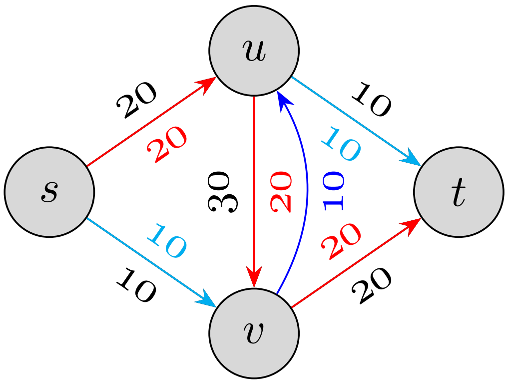
Variations

- Modeling multiple sources and/or multiple sinks



Designing the algorithm

Problem: Solve the maximum flow problem on this flow network.



First push 20 units along the edges (s, u), (u, v) and (v, t).

Now push 10 units along (s, v).

This results in too much flow coming into v. So, we "undo" 10 units of flow on (u, v).

That restores the conservation condition at v, but results in too little flow leaving u. So, finally we push 10 units along (u, t).

More general way: push forward on edges with leftover capacity and push backward on edges that are already carrying flow to divert it in a different direction.

Residual Capacity

Definition

Let f be a flow in a flow network $G = (V, E)$ with source s and sink t. Consider a pair of vertices $u, v \in V$.

We define the **residual capacity** $c_f(u, v)$ by

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

Given a flow network $G = (V, E)$ and a flow f, the **residual network of G induced by f** is $G_f = (V, E_f)$, where

$$E_f := \{(u,v) \in V \times V \mid c_f(u,v) > 0\}.$$

Note

Since each edge in E can give rise to at most two edges in G_f, we have $|E| \leq |E_f| \leq 2|E|$.

Example

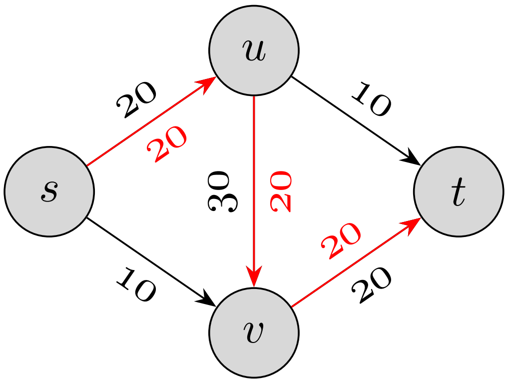


Figure: A flow f through G in red

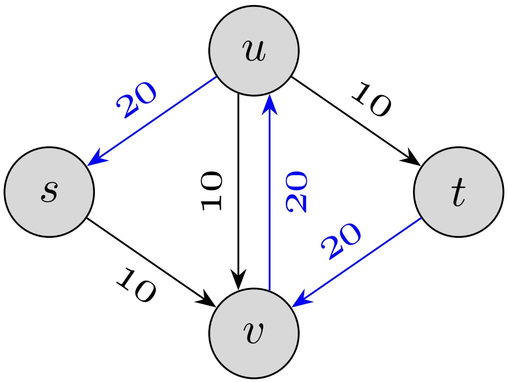


Figure: Residual graph G_f for the above flow f with black forward edges and blue backward edges

Augmenting Path in a Residual Graph

An **augmenting path** P is a simple $s-t$ path in G_f, i.e., P is a path from s to t in G_f and does not visit any node more than once.

Definition

$bottleneck(P, f) :=$ min$\{c_f(u, v) \mid (u, v) \in P\}$.

Define a new flow f' in G as follows:

---

procedure $\textsc{Augment}(f, P)$
    $b := bottleneck(P, f)$
    for each edge $(u, v) \in P$ **do**
        if (u, v) is a forward edge **then**
            $f'(u, v) = f(u, v) + b$
        else $\#(u, v)$ is a backward edge
            $f'(v, u) = f(v, u) - b$
    return f'

New flow is a flow on the original graph

Claim 1

Let $f' = \textsc{Augment}(f, P)$. Then f' is a new flow in G.

Proof.

We must verify that f' satisfies capacity constraint and flow conservation.

For capacity constraint, note that f' differs from f only on the edges of P. So, we need to verify the capacity constraints only on the edges of P.

Observe that

$$bottleneck(P, f) \leq c_f(u, v)$$

for any edge $(u, v) \in P$.

Proof continued...

Now, if (u, v) is a forward edge, then

$$\begin{aligned} 0 \leq f(u,v) \leq f'(u,v) &= f(u,v) + bottleneck(P,f) \\ &\leq f(u,v) + c_f(u,v) \\ &= f(u,v) + c(u,v) - f(u,v) \\ &= c(u,v). \end{aligned}$$

And, if (u, v) is a backward edge, then $c_f(u,v) = f(v,u)$ and

$$\begin{aligned} c(v,u) \geq f(v,u) \geq f'(v,u) &= f(v,u) - bottleneck(P,f) \\ &\geq f(v,u) - c_f(u,v) \\ &= f(v,u) - f(v,u) \\ &= 0. \end{aligned}$$

Thus, the capacity constraints hold in either case.

Proof continued...

Now, we have to verify that the conservation condition holds at each internal node v that lies on the path P.

Let $u \to v \to w$ be the two edges coming into and leaving v, respectively, on path P. Then there are four cases to consider.

- Both $u \to v$ and $v \to w$ are forward edges
- Both $u \to v$ and $v \to w$ are backward edges
- $u \to v$ is a forward edge while $v \to w$ is a backward edge
- $u \to v$ is a backward edge while $v \to w$ is a forward edge.

Exercise: Check that conservation condition holds at v in all cases.

$\square$

New flow is better than the original flow

Claim 2
$|f'| > |f|$.

Proof.
The first edge of P must be an edge out of s in G_f.

Since the path is simple, it does not visit s again.

Since G has no edge entering s, the edge must be a forward edge.

We increase the flow on this edge by $bottleneck(P, f)$, and we do not change the flow on any other edge incident to s.

Hence, $|f'| = |f| + bottleneck(P, f) > |f|$. $\square$

Ford-Fulkerson Algorithm

procedure $\textsc{Ford-Fulkerson}(G, s, t)$
    Define a flow variable f
    for each edge $(u, v) \in E$ **do**
        $f(u, v) = 0$
    $G_f := G$
    while there exists a path from s to t in G_f **do**
        Let P be one such path
        $f' = \textsc{Augment}(f, P)$
        Compute $G_{f'}$ # residual network of G induced by f'
        $f = f'$
        $G_f = G_{f'}$
    return f

Cuts and Flows

Definition

A **cut** $C = (S, T)$ of a flow network $G = (V, E)$ is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.

The **net flow** $f(S, T)$ for a flow f across the cut (S, T) is defined as

$$f(S, T) := \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u).$$

The **capacity** $c(S, T)$ of the cut (S, T) is defined as

$$c(S, T) := \sum_{u \in S} \sum_{v \in T} c(u, v).$$

A **minimum cut** of a network is a cut whose capacity is minimum over all cuts of the network.

Example

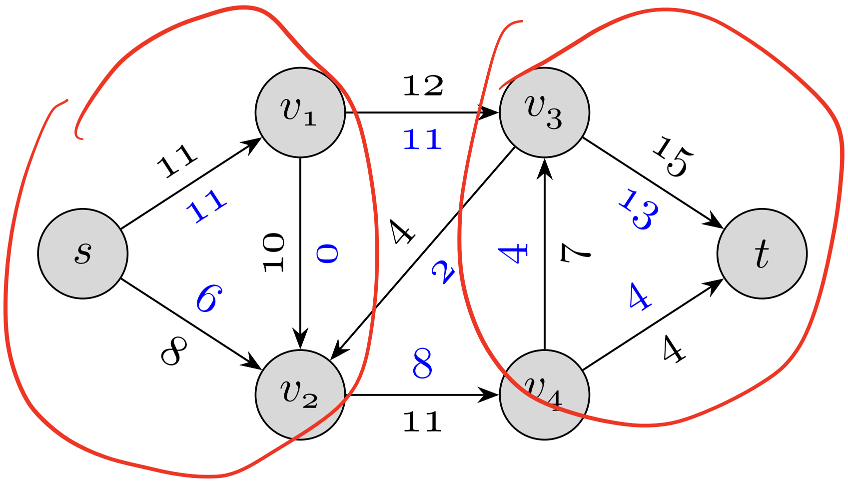


Figure: A network flow with flow in blue and capacities in black.

Let $S = \{s, v_1, v_2\}$ and $T = \{v_3, v_4, t\}$. Then, $C = (S, T)$ is a cut.

The net flow for f across the cut is $f(S, T) = 11 + 8 - 2 = 17$.

The capacity of the cut is $c(S, T) = 12 + 11 = 23$.

Fact

Lemma

Let f be a flow in a flow network G and let $C = (S, T)$ be any cut of G. Then $f(S, T) = |f|$.

Proof.

Observe that

$$
\begin{aligned}
|f| &= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) \\
&= \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) + \sum_{u \in S \setminus \{s\}} \left(\sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u) \right) \\
&= \sum_{v \in V} \sum_{u \in S} f(u, v) - \sum_{v \in V} \sum_{u \in S} f(v, u)
\end{aligned}
$$

The terms on the right in the second line above are all zeros due to flow conservation.

Proof continued...

Because $V = S \cup T$ and $S \cap T = \emptyset$, we obtain

$$\begin{aligned}
|f| &= \sum_{v \in T} \sum_{u \in S} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) + \\
&\quad \Big(\sum_{v \in S} \sum_{u \in S} f(u,v) - \sum_{v \in S} \sum_{u \in S} f(v,u) \Big) \\
&= f(S,T) + 0 \\
&= f(S,T).
\end{aligned}$$

$\square$

Fact

Corollary

The value of any flow f in a flow network G is bounded above by the capacity of any cut of G.

Proof.

$$\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u) \\
&\le \sum_{u\in S}\sum_{v\in T} f(u,v) \\
&\le \sum_{u\in S}\sum_{v\in T} c(u,v) \\
&= c(S,T).
\end{aligned}$$

$\square$

Max-Flow-Min-Cut Theorem

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.*

Proof.

$1 \implies 2$. If there is an augmenting path P in G_f, then $|\textsc{Augment}(f, P)| > |f|$, contradicting the maximality of f.

$3 \implies 1$. $|f| \leq c(S, T)$ for all cuts (S, T) by previous Corollary. The condition $|f| = c(S, T)$ thus implies that f is a maximum flow.

Proof continued...

$2 \implies 3$. Suppose G_f has no augmenting path, i.e., no path from s to t.

Define $S = \{v \in V \mid \text{ there is a path from } s \text{ to } v \text{ in } G_f\}$ and $T = V \setminus S$.

Clearly, (S, T) is a cut: we have $s \in S$ and $t \in T$.

Now, consider $u \in S$ and $v \in T$.

If $(u, v) \in E$, we must have $f(u, v) = c(u, v)$, since otherwise $(u, v) \in E_f$, which would place $v \in S$.

If $(v, u) \in E$, we must have $f(v, u) = 0$, because otherwise $c_f(u, v) = f(v, u) > 0$, and we would have $(u, v) \in E_f$, which would place $v \in S$.

If neither (u, v) nor (v, u) is in E, then $f(u, v) = f(v, u) = 0$.

Proof continued...

Thus,

$$
\begin{aligned}
f(S,T) &= \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u) \\
&= \sum_{u\in S}\sum_{v\in T} c(u,v) - \sum_{u\in S}\sum_{v\in T} 0 \\
&= c(S,T).
\end{aligned}
$$

By the lemma, we obtain $|f| = f(S,T) = c(S,T)$.

$\square$

Consequences

Thus, if f is an $s-t$ flow such that there is no $s-t$ path in the residual graph G_f, then there is an $s-t$ cut (A^*, B^*) in G for which $|f| = c(A^*, B^*)$.

Consequently, f has the maximum value of any flow in G, and (A^*, B^*) has the minimum capacity of any $s-t$ cut in G.

Corollary

1. *The flow $\bar{f}$ returned by* Ford-Fulkerson *is a maximum flow.*
2. *Given a flow f of maximum value, we can compute an $s-t$ cut of minimum capacity in $\mathcal{O}(m)$ time.*
3. *In every flow network, the maximum value of an $s-t$ flow is equal to the minimum capacity of an $s-t$ cut.*

Complexity

procedure FORD-FULKERSON(G, s, t)
    Define a flow variable f
    for each edge $(u, v) \in E$ **do**
        $f(u, v) = 0$
    $G_f := G$
    while there exists a path from s to t in G_f **do**
        Let P be one such path
        $f' = \text{AUGMENT}(f, P)$
        Compute $G_{f'}$ # residual network of G induced by f'
        $f = f'$
        $G_f = G_{f'}$
    return f

Complexity: $\mathcal{O}(|E||f^*|)$, where $|f^*|$ is the value of a maximum flow in G (provided all capacities are integral/rational) - pseudo-polynomial.

Worst Case Example

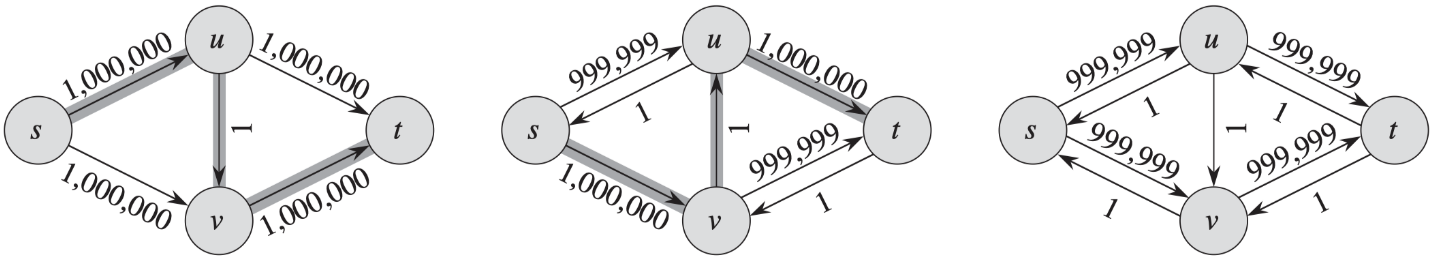


Figure: Poor choice of augmenting paths when capacities are integral

- What happens if the capacities are rational numbers?
- What happens if the capacities are real numbers?

Edmonds-Karp Algorithm

To do a better job, one needs to find a better way of computing an augmented path.

There are many algorithms that achieve that.

One such algorithm is **Edmonds-Karp algorithm**. It finds the augmenting paths by a breadth-first search in G_f, i.e., it chooses an augmenting path as a shortest path in G_f from s to t where each edge has weight 1.

Edmonds-Karp Algorithm

procedure $\text{Edmonds-Karp}(G, s, t)$
    Define a flow variable f
    for each edge $(u, v) \in E$ **do**
        $f(u, v) = 0$
    $G_f := G$
    while there exists a path from s to t in G_f **do**
        Let P be an $s - t$ path in G_f with minimum number of edges
        $f' = \text{Augment}(f, P)$
        Compute $G_{f'}$
        $f = f'$
        $G_f = G_{f'}$
    return f

Complexity: $\mathcal{O}(|V||E|^2)$.