COMP 761: Lecture 27 – Max Flow

David Rolnick

November 6, 2020

Problem

A *cut* in a flow network is a partition of the vertices into S and T, where $s \in S$ and $t \in T$. The *capacity* of the cut (S, T) is:

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

Prove that $|f| \leq c(S, T)$ for any flow f and cut (S, T).

(Please don't post your ideas in the chat just yet, we'll discuss the problem soon in class.)

Course Announcements

Course Announcements

- On Problem 1, reminder not just to cite that polynomials are Θ(leading term) unless you prove that. May want to provide a constant c for Ω notation explicitly. If this was not clear to you, feel free to take an additional 24 hours before submitting (you can edit an existing submission). Slack message with any questions.

Review: Flow networks

Review: Flow networks

- We have seen problems (shortest path and spanning tree) where weights on edges in a graph are used to represent lengths.

Review: Flow networks

- We have seen problems (shortest path and spanning tree) where weights on edges in a graph are used to represent lengths.
- They can also represent the maximum capacity on edges.

Review: Flow networks

- We have seen problems (shortest path and spanning tree) where weights on edges in a graph are used to represent lengths.
- They can also represent the maximum capacity on edges.
- A *flow network* is a directed graph where, for each edge (u, v), there is an associated capacity $c(u, v) > 0$.

Review: Flow networks

- We have seen problems (shortest path and spanning tree) where weights on edges in a graph are used to represent lengths.
- They can also represent the maximum capacity on edges.
- A *flow network* is a directed graph where, for each edge (u, v), there is an associated capacity $c(u, v) > 0$.
- We assume there are no self-loops (u, u).

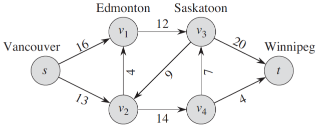
Review: Flow networks

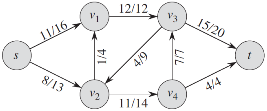
- We have seen problems (shortest path and spanning tree) where weights on edges in a graph are used to represent lengths.
- They can also represent the maximum capacity on edges.
- A *flow network* is a directed graph where, for each edge (u, v), there is an associated capacity $c(u, v) > 0$.
- We assume there are no self-loops (u, u).
- A *flow* from s to t is a function f where if (u, v) is an edge, then

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

and $f(u, v) = 0$ if there is no edge (u, v), and where for all u different from s and t,

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





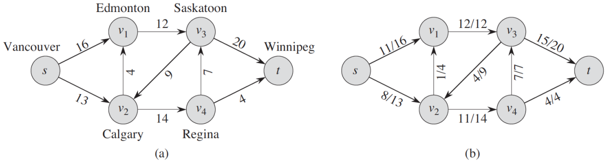
Review: Flow networks

- A *flow* from s to t is a function f where if (u, v) is an edge, then

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

and $f(u,v) = 0$ if there is no edge (u, v), and where for all u different from s and t,

$$\sum_v f(v,u) = \sum_v f(u,v).$$



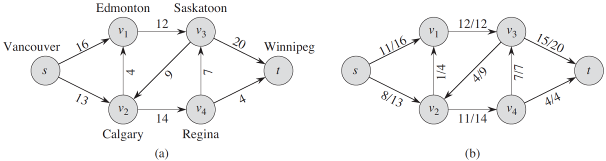
Review: Flow networks

- A *flow* from s to t is a function f where if (u, v) is an edge, then

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

and $f(u, v) = 0$ if there is no edge (u, v), and where for all u different from s and t,

$$\sum_v f(v, u) = \sum_v f(u, v).$$



- We say that the *value* $|f|$ of a flow f is defined by:

$$|f| = \sum_v f(s, v) - \sum_v f(v, s).$$

Maximum flow

Maximum flow

- The problem of *maximum flow* is, given a flow network, to find a flow f with maximum value $|f|$.

Maximum flow

- The problem of *maximum flow* is, given a flow network, to find a flow f with maximum value $|f|$.
- (There might be several with this maximum value.)
- Applications include routing shipments, calculating flow of water or electricity, airplane scheduling, image segmentation, etc.

Antiparallel edges

Antiparallel edges

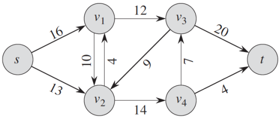
- We will assume that flow networks don't have any antiparallel edges – i.e. never have both (u, v) and (v, u).

Antiparallel edges

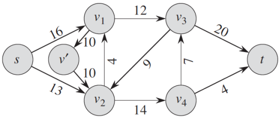
- We will assume that flow networks don't have any antiparallel edges – i.e. never have both (u, v) and (v, u).
- Why can we make this assumption?

Antiparallel edges

- We will assume that flow networks don't have any antiparallel edges – i.e. never have both (u, v) and (v, u).
- Why can we make this assumption?
- We can transform a graph with antiparallel edges into one without:



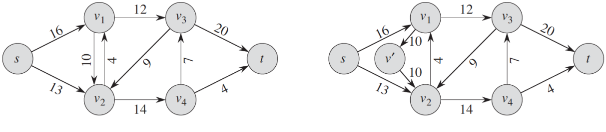
(a)



(b)

Antiparallel edges

- We will assume that flow networks don't have any antiparallel edges – i.e. never have both (u, v) and (v, u).
- Why can we make this assumption?
- We can transform a graph with antiparallel edges into one without:



(a) (b)

- Note that we can't just "cancel" the antiparallel capacities, since that would prevent us traveling in one direction along that edge.

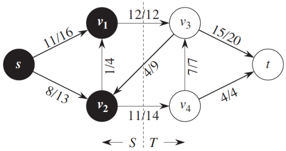
Cuts

Cuts

- Suppose we have a flow network with flow f from s to t.

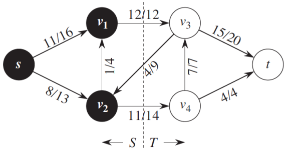
Cuts

- Suppose we have a flow network with flow f from s to t.
- A *cut* is a partition of the vertices into S and T, where $s \in S$ and $t \in T$.



Cuts

- Suppose we have a flow network with flow f from s to t.
- A *cut* is a partition of the vertices into S and T, where $s \in S$ and $t \in T$.

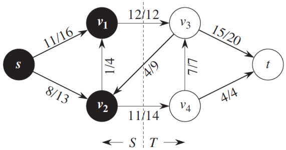


- The *net flow* of the cut (S, T) is:

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

Cuts

- Suppose we have a flow network with flow f from s to t.
- A *cut* is a partition of the vertices into S and T, where $s \in S$ and $t \in T$.



- The *net flow* of the cut (S, T) is:

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

- **Claim:** The net flow $f(S, T)$ equals the flow value $|f|$.

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- How can we prove this?

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- Similar to showing flow into t = flow out of s.

$$|f| = \sum_v f(s, v) - \sum_v f(v, s)$$

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- Similar to showing flow into t = flow out of s.

$$|f| = \sum_v f(s, v) - \sum_v f(v, s)$$

$$= \sum_v f(s, v) - \sum_v f(v, s) + \sum_{u \in S - \{s\}} \left(\sum_v (f(u, v) - f(v, u)) \right)$$

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- Similar to showing flow into t = flow out of s.

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

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- Similar to showing flow into t = flow out of s.

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

Cuts

Claim: The net flow $f(S, T)$ equals the flow value $|f|$.

- Similar to showing flow into t = flow out of s.

$$
\begin{aligned}
|f| &= \sum_{v} f(s, v) - \sum_{v} f(v, s) \\
&= \sum_{v} f(s, v) - \sum_{v} f(v, s) + \sum_{u \in S - \{s\}} \left(\sum_{v} (f(u, v) - f(v, u)) \right) \\
&= \sum_{u \in S} \sum_{v} (f(u, v) - f(v, u)) \\
&= \sum_{u \in S} \sum_{v \in S} (f(u, v) - f(v, u)) + \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u)) \\
&= 0 + \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u))
\end{aligned}
$$

Cuts

Claim: The net flow $f(S,T)$ equals the flow value $|f|$.

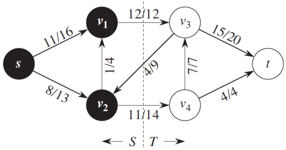
- Similar to showing flow into t = flow out of s.

$$
\begin{aligned}
|f| &= \sum_v f(s,v) - \sum_v f(v,s) \\
&= \sum_v f(s,v) - \sum_v f(v,s) + \sum_{u \in S - \{s\}} \left(\sum_v (f(u,v) - f(v,u)) \right) \\
&= \sum_{u \in S} \sum_v (f(u,v) - f(v,u)) \\
&= \sum_{u \in S} \sum_{v \in S} (f(u,v) - f(v,u)) + \sum_{u \in S} \sum_{v \in T} (f(u,v) - f(v,u)) \\
&= 0 + \sum_{u \in S} \sum_{v \in T} (f(u,v) - f(v,u)) \\
&= f(S,T)
\end{aligned}
$$

Min cut

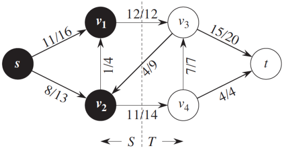
Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.



Min cut

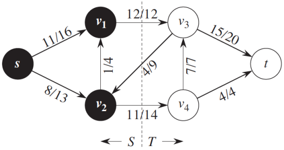
- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.



Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.

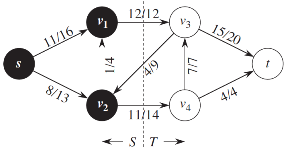


Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

- Why is this true?

Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.

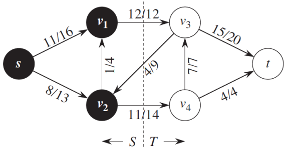


Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

- Let f be a flow from s to t, and let (S, T) be a cut.

Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.



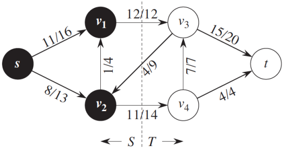
Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

- Let f be a flow from s to t, and let (S, T) be a cut.
- We have:

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

Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.



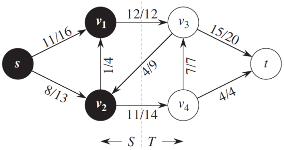
Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

- Let f be a flow from s to t, and let (S, T) be a cut.
- We have:

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

Min cut

- The *capacity* of the cut (S, T) is $c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$.



Claim: The maximum flow from s to t is at most the minimum capacity of any cut (S, T).

- Let f be a flow from s to t, and let (S, T) be a cut.
- We have:

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

Residual network

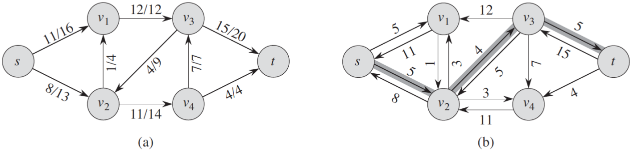
Residual network

- Given a flow network G and a flow f in it, we can define the *residual network* G_f as follows:

Residual network

- Given a flow network G and a flow f in it, we can define the *residual network* G_f as follows:
- If (u, v) is an edge in G, then both (u, v) and (v, u) are edges in G_f, with *residual capacities*

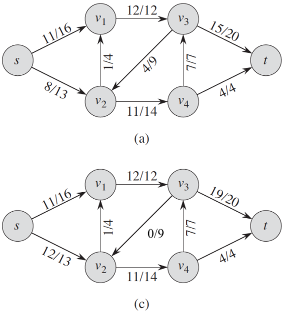
$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$



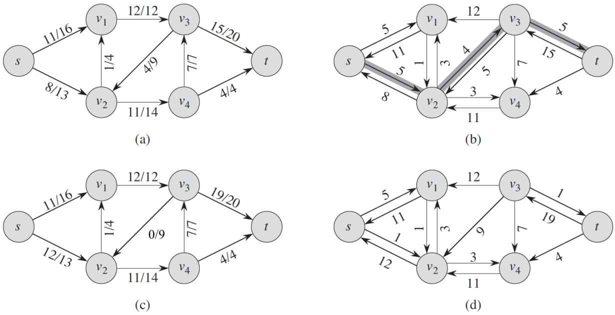
Augmenting paths

Augmenting paths

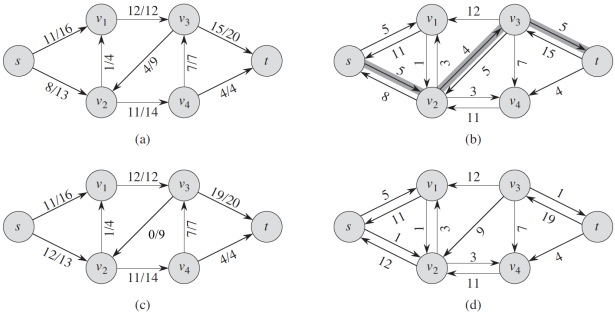
- Given a flow network G and a flow f from s to t, an *augmenting path* is a path from s to t in the residual network with positive capacity along all edges.



Ford-Fulkerson method

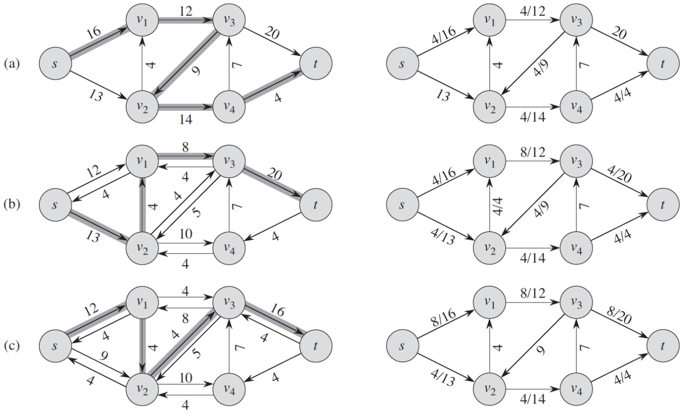


Ford-Fulkerson method

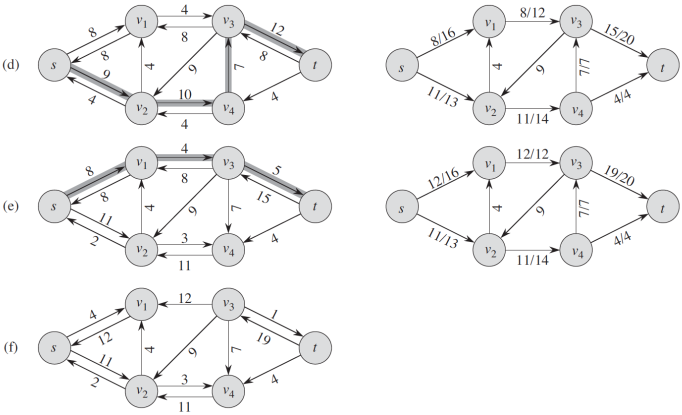


- The *Ford-Fulkerson method* for solving max flow consists of:
  1. Start out with $f(u, v) = 0$ everywhere.
  2. Find an augmenting path in the residual network G_f.
  3. Update f by adding flow along that path.
  4. Repeat Steps 2 and 3 until there is no augmenting path.

Ford-Fulkerson method



Ford-Fulkerson method



Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
- So there are three things we need to prove.

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
- So there are three things we need to prove.
- Why does (1) imply (2)?

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
- So there are three things we need to prove.
- Why does (1) imply (2)?
- If we have an augmenting path in G_f, we can add it to f to get a bigger flow, which means f wasn't maximum.

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
- So there are three things we need to prove.
- Why does (1) imply (2)?
- If we have an augmenting path in G_f, we can add it to f to get a bigger flow, which means f wasn't maximum.
- Why does (3) imply (1)?

Max-flow min-cut theorem

Will prove Ford-Fulkerson by showing the following are all 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.

This is called the **max-flow min-cut theorem**.

- We will prove this by showing that (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1).
- So there are three things we need to prove.
- Why does (1) imply (2)?
- If we have an augmenting path in G_f, we can add it to f to get a bigger flow, which means f wasn't maximum.
- Why does (3) imply (1)?
- We know $|f| \leq c(S, T)$ for every flow f, so if $|f| = c(S, T)$, then f must be maximum.

Proof

Proof

- Let us now prove that (2) implies (3).

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).
- What might be a good cut to consider?

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).
- What might be a good cut to consider?
- Let S be the set of vertices u where there's a path s to u in G_f.

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).
- What might be a good cut to consider?
- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).
- What might be a good cut to consider?
- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

Proof

- Let us now prove that (2) implies (3).
- Suppose the residual network G_f contains no augmenting paths. Then, we need to prove $|f| = c(S, T)$ for some cut (S, T).
- There is no path from s to t in G_f (with all positive capacity edges).
- What might be a good cut to consider?
- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.
- We know $c_f(u, v) = 0$ by definition of T.

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$
\begin{aligned}
c_f(u, v) &= c(u, v) - f(u, v) \\
c_f(v, u) &= f(u, v).
\end{aligned}
$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.
- We know $c_f(u, v) = 0$ by definition of T.
- So $c(u, v) = f(u, v)$ and $f(v, u) = 0$.

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.
- We know $c_f(u, v) = 0$ by definition of T.
- So $c(u, v) = f(u, v)$ and $f(v, u) = 0$.
- Then:

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

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u, v) = c(u, v) - f(u, v)$$
$$c_f(v, u) = f(u, v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.
- We know $c_f(u, v) = 0$ by definition of T.
- So $c(u, v) = f(u, v)$ and $f(v, u) = 0$.
- Then:

$$|f| = f(S, T) = \sum_{u \in S} \sum_{v \in T} (f(u, v) - f(v, u))$$
$$= \sum_{u \in S} \sum_{v \in T} (c(u, v) - 0)$$

Proof

- Let S be the set of vertices u where there's a path s to u in G_f.
- Let T be all the other vertices (unreachable from s).
- Recall the definition of the residual graph: If (u, v) is an edge in G, then G_f has the residual capacities

$$c_f(u,v) = c(u,v) - f(u,v)$$
$$c_f(v,u) = f(u,v).$$

- Suppose (u, v) is an edge across the cut, with $u \in S, v \in T$.
- We know $c_f(u, v) = 0$ by definition of T.
- So $c(u, v) = f(u, v)$ and $f(v, u) = 0$.
- Then:

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

Max-flow min-cut theorem

Max-flow min-cut theorem

- We have thus proved the following are all 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.

Max-flow min-cut theorem

- We have thus proved the following are all 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.
- It also shows that the max flow is equal to the min cut.

Max-flow min-cut theorem

- We have thus proved the following are all 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.
- It also shows that the max flow is equal to the min cut.
- This shows that the Ford-Fulkerson method will eventually give us a max flow.

Running time

Running time

- Let's assume that all the weights are integers.

Running time

- Let's assume that all the weights are integers.
- How long might this take?

Running time

- Let's assume that all the weights are integers.
- How long might this take?
- Each one of the augmenting paths increases the flow by at least 1 since each flow value is a positive integer.

Running time

- Let's assume that all the weights are integers.
- How long might this take?
- Each one of the augmenting paths increases the flow by at least 1 since each flow value is a positive integer.
- So the number of iterations is at most the value $|f^*|$ of the maximum flow.

Running time

- Let's assume that all the weights are integers.
- How long might this take?
- Each one of the augmenting paths increases the flow by at least 1 since each flow value is a positive integer.
- So the number of iterations is at most the value $|f^*|$ of the maximum flow.
- Can do a *depth-first* or *breadth-first* search in a graph to find a path in time $O(|E|)$ (won't go into details here), so overall time is

$$O(|E| \cdot |f^*|).$$

Running time

- Let's assume that all the weights are integers.
- How long might this take?
- Each one of the augmenting paths increases the flow by at least 1 since each flow value is a positive integer.
- So the number of iterations is at most the value $|f^*|$ of the maximum flow.
- Can do a *depth-first* or *breadth-first* search in a graph to find a path in time $O(|E|)$ (won't go into details here), so overall time is

$$O(|E| \cdot |f^*|).$$

- An improvement in the algorithm (the Edmonds-Karp algorithm) can actually make it:

$$O(|V| \cdot |E|^2).$$