CSC 373 - Algorithm Design, Analysis, and Complexity **Summer 2016**

Lalla Mouatadid

Network Flows: The Max Flow/Min Cut Theorem

In this lecture, we prove optimality of the Ford-Fulkerson theorem, which is an immediate corollary of a well known theorem: The Max-Flow/Min-Cut theorem, which says:

The Max-Flow/Min-Cut Theorem:

Let (G, s, t, c) be a flow network and left f be a flow on the network. The following is equivalent:

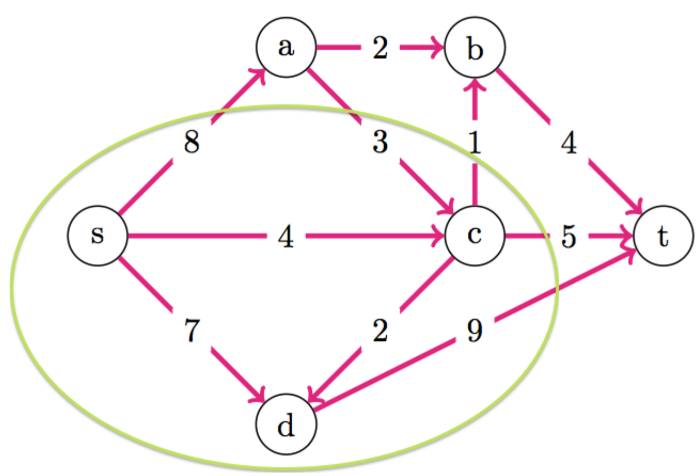
1. f is maximized.
2. G_f has no augmenting paths.
3. There exists a cut $C(S, T)$ such that $c(S, T) = val(f)$.

What is a cut?

Definition : Let (G, s, t, c) be a flow network, an **s-t cut** in G is a partition of V into two sets S and T such that:

1. $S \cup T = V$
2. $S \cap T = \emptyset$
3. $s \in S$ and $t \in T$

Below is an example of a cut $C(S, T)$ where $S = \{s, c, d\}$ and $T = \{a, b, t\}$.



The **capacity** of a cut $C(S, T)$, denoted $c(S, T)$, is the sum of the capacities of the edges (u, v)[1] with $u \in S$

[1]Recall that edges are directed, and thus (u, v) means the edge from u to v

and $v \in T$. That is:

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v) \tag{1}$$

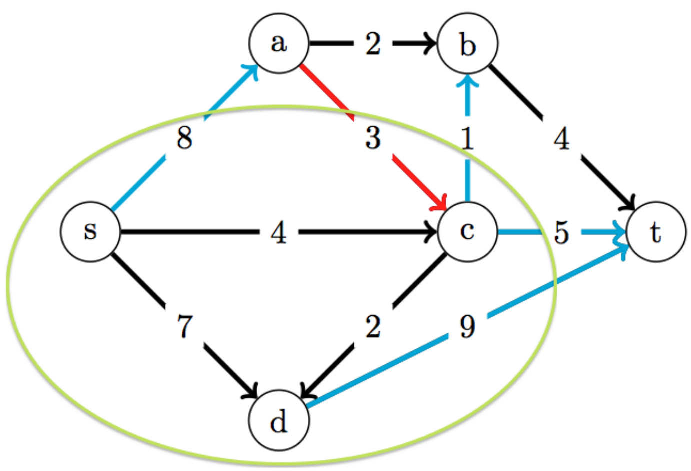
In the example above, $c(S,T) = 23$, we don't count the edge (a,c) since $a \in T, c \in S$.

This definition of capacity of a cut is very natural, and it suggests we can define the flow of a cut in a similar manner. That is, given a cut $C(S,T)$ with capacity $c(S,T)$, and a flow f, how much of f crosses from S to T? Intuitively this would be the flow going from S to T minus whatever flow was sent back. And this is exactly it.

The flow of a cut $C(S,T)$, denoted $f(C,T)$ is defined as:

$$f(S,T) = \Big(\sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u) \Big) \tag{2}$$

Again, if we consider the previous example, redrawn below: $f(S,T)$ would equal the sum of the flow sent across the blue edges minus whatever flow sent down the red edges.



By the capacity constraint, we know that $f(u,v) \leq c(u,v), \forall (u,v) \in E$. Using this fact and (1) and (2) above, we get the following lemma:

Lemma 1. *Let (G,s,t,c) be a flow network, $C(S,T)$ an $s-t$ cut and f a flow, then:*

$$f(S,T) \leq c(S,T)$$

Proof. For every (u,v) in the network, we know that

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

Therefore:

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

□

What this lemma says is that **any** cut you choose in the network, the flow going through $C(S,T)$ is always bounded by its capacity. Any cut. Now, consider the following lemma:

Lemma 2. *Let (G,s,t,c) be a flow network, $C(S,T)$ an $s-t$ cut, f a flow and v an element of T:*

$$f(S,T) = f(S\cup\{v\}, T\backslash\{v\})$$

Hmm what? If the flow is upper bounded by the cut, can't we pick a vertex $v \in T$ with an incoming neighbour $u \in S$ such that $c(u,v)$ is very high. Intuitively moving v to S would reduce the capacity of this new cut (its total capacity would lose $c(u,v)$), and so its corresponding flow can easily be less than $f(S,T)$ that we started with. No?

Well, no. Here's why:

Proof. Let $C(S,T)$ be any $s-t$ cut and v an element in T. Remove v from T and place it in S, and now let's evaluate the flow of this new cut $C'(S',T')$ where $S' = S\cup\{v\}, T' = T\backslash\{v\}$.

Let $In(v) = \{(u,v) \in E | u \in V\}$ denote the set of incoming edges to v, and $Out(v) = \{(v,w) \in E | w \in W\}$ the set of outgoing edges from v. We know that by conservation of flow:

$$\sum_{(u,v)\in In(v)} f(u,v) = \sum_{(v,w)\in Out(v)} f(v,w)$$

We partition $In(v)$ and $Out(v)$ based on where the end points of the edges fall as follows:

$$\begin{aligned}
In_S(v) &= \{(u,v) \in E | u \in S\} \\
In_T(v) &= \{(u,v) \in E | u \in T\} \\
Out_S(v) &= \{(v,w) \in E | w \in S\} \\
Out_T(v) &= \{(v,w) \in E | w \in T\}
\end{aligned}$$

Let's evaluate the flow of this new cut now. Moving v into S will result in loosing v's contribution to the original cut capacity, but also in gaining the capacity of the outgoing edges from v thus:

$$\begin{aligned}
f(S\cup\{v\}, T\backslash\{v\}) &= f(S,T) - \sum_{(u,v)\in In_S(v)} f(u,v) - \sum_{(u,v)\in In_T(v)} f(u,v) + \sum_{(v,w)\in Out_S(v)} f(v,w) + \sum_{(v,w)\in Out_T(v)} f(v,w) \\
&= f(S,T) - \Big(\sum_{(u,v)\in In(v)} f(u,v)\Big) + \Big(\sum_{(v,w)\in Out(v)} f(v,w)\Big) \\
&= f(S,T)
\end{aligned}$$

The second equality follows from $In_S(v) \cup In_T(v) = In(v)$ and $Out_S(v) \cup Out_T(v) = Out(v)$, and the last equality follows from conservation of flow. □

This is nice. Really! Consider applying this lemma to the following cut $C(S,T) : S = \{s\}, T = V\backslash\{s\}$, then we get:

$$f(\{s\}, V\backslash\{s\}) = f(\{s\} \cup S', V\backslash(\{s\} \cup S')) = f(S', T') \tag{3}$$

For any $S' \subset V\backslash\{s\}$!! This says that the flow of **any** cut equals the flow of the first cut we started with, namely $(\{s\}, V\backslash\{s\})$.

But what do we know about the flow of the cut $C(\{s\}, V\backslash\{s\})$? This is just the flow leaving the source s, and we know the this flow is just the flow of the network[2]:

$$val(f) = \sum_{(s,u)\in E} f(s,u) \tag{4}$$
$$= f(\{s\}, V\backslash\{s\}) \tag{5}$$

Therefore using (3), (4) and (5), we get the following lemma:

Lemma 3. *Let (G, s, t, c) be a flow network, $C(S,T)$ any s-t cut and f a flow on G, then:*

$$f(S,T) = val(f)$$

Now combining Lemma 1 and Lemma 3, we get the following corollary:

Corollary 1. *Let (G, s, t, c) be a flow network, $C(S,T)$ any s-t cut and f a flow on G, then:*

$$val(f) \leq c(S,T)$$

Contemplate this corollary :) then scroll back up to the statement of the Max-Flow/Min-Cut Theorem. And before reading further, can you see why Max-Flow/Min-Cut is the name of this theorem?

Corollary 1 says that the capacity of *any* cut is at least the value of a flow on G. The Max-Flow/Min-Cut Theorem says that there exists a cut whose capacity is *minimized* (i.e. $c(S,T) = val(f)$) but this only happens when f itself is the *maximum* flow of the network! Therefore, in any flow network (G, s, t, c), the value of the maximum flow equals the capacity of the minimum cut in the network.

Now, finally:

Proof of the Max-Flow/Min-Cut Theorem. We want to show that the 3 points of the theorem are equivalent, so we'll prove that $1 \implies 2, 2 \implies 3$ and $3 \implies 1$.

$1 \implies 2$: Let f be a max flow and suppose G_f still has an augmenting path $\mathcal{P}$. Then we can increase $val(f)$ by augmenting along $\mathcal{P}$, thus contradicting the maximality of f.

$2 \implies 3$: Suppose G_f has no augmenting paths. We will construct a cut $C(S,T)$ such that $c(S,T) = val(f)$. Let S denote the set of vertices that are reachable from s: If there is an augmenting path from s to a vertex v then $v \in S$. Since G_f has no augmenting s,t path, we know that $t \notin S$. Let $T = V\backslash S$, thus $t \in T$. And so $C(S,T)$ is a valid s-t cut. What's the flow across $C(S,T)$?

Consider an edge (u,v) crossing the cut, i.e. $u \in S, v \in T$. Recall that $c_f(u,v)$ denotes the capacity of the edge (u,v) in the residual graph G_f. We know that $c_f(u,v) = 0$ since otherwise there would be an augmenting s,v path and v would've been placed in S. Now we just need to know where the edge (u,v) comes from.

[2]See the previous lecture

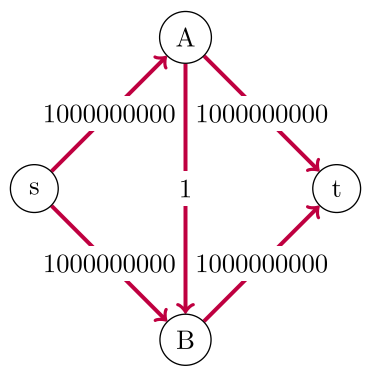
If (u,v) is an edge of G, then since $c_f(u,v) = 0$, it follows that $c(u,v) - f(u,v) = 0$ and thus $c(u,v) = f(u,v)$. If (u,v) is not an edge of G (i.e. it was added in the residual graph), then $f(v,u) = 0$. Why?[3]. We therefore have the following:

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

$3 \implies 1$: Let $C(S,T)$ be a cut with capacity $c(S,T) = val(f)$, and let f' be a maximum flow in G, so $val(f) \leq val(f')$. Since $val(f) = c(S,T)$, $c(S,T) \leq val(f')$. And by Corollary 1, $val(f') \leq c(S,T)$. And thus $val(f') = c(S,T) = val(f)$. $\square$

Et voilà! Now the optimality of Ford-Fulkerson is a one line proof (not even): $2 \implies 1$:) It should be clear now that Ford-Fulkerson also solves the minimum capacity cut problem: Given a network, compute its minimum capacity s,t cut.

Consider the network below:



How long does it take to compute the max flow using Ford-Fulkerson? Well, the algorithm could be sending 1 unit of flow at every iteration: Do a BFS, find an augmenting path, augment the flow by 1. So in total, it could take $\mathcal{O}((m+n) * val(f))$! Eeek! Thankfully, there is a better implementation, due to Edmonds and Karp where they show that if we carefully select which augmenting path to take at every iteration (namely the one with the fewest edges amongst all possible augmenting paths), then we can implement Ford-Fulkerson in $\mathcal{O}(nm^2)$.

[3]Notice that it's $f(v,u) = 0$ and not $f(u,v) = 0$.