Analysis of Algorithms I: Maximum Flow

Xi Chen

Columbia University
In this class, we start by introducing the maximum flow problem. We then present the Ford-Fulkerson method based on the Max-Flow Min-Cut Theorem. In the next class, we will discuss one particular implementation of the Ford-Fulkerson method: the Edmonds-Karp algorithm for the maximum flow problem.
Let $G = (V, E)$ be a directed graph with positive edge weights $c : E \rightarrow \mathbb{R}_{>0}$. For now we assume all the weights are positive integers. We also assume $G$ satisfies the following condition:

$$G \text{ is reduced: If } (u, v) \in E \text{ then } (v, u) \notin E.$$ 

We make this assumption mainly to simplify the analysis. We will see shortly how to work around this restriction. There are two distinguished vertices in $G$: a vertex $s$ called the source and a vertex $t$ called the sink. (Note that here $s$ may have incoming edges and $t$ may have outgoing edges.)
Consider $G$ as a computer network and we want to send data from $s$ to $t$. An edge from $u$ to $v$ with weight $c(u, v) > 0$ means we can send data from $u$ to $v$ at a maximum rate of $c(u, v)$ Mbps. Given $G$, what is the maximum rate of sending data from $s$ to $t$? This is what we call the maximum flow problem. Since we want to send data from $s$ to $t$, we may well assume that every $v \in V$ lies on a path from $s$ to $t$. Otherwise, we can remove $v$ from the graph.
Given $G$, $c$ (positive edge weights), $s$ (source) and $t$ (sink), a flow $f$ in $G$ is a nonnegative function $f : E \rightarrow \mathbb{R}_{\geq 0}$ such that

1. **Capacity constraint:** For every $(u, v) \in E$,
   
   $$0 \leq f(u, v) \leq c(u, v)$$

2. **Flow conservation constraints:** For every $u \in V - \{s, t\}$,
   
   $$\sum_{v: (u, v) \in E} f(u, v) = \sum_{v: (v, u) \in E} f(v, u)$$

For the second cond., consider a router $u$ in a network. Its out-flow (sum on the left) should be equal to its in-flow (sum on the right).
The value of a flow $f$, denoted by $|f|$, is then defined as

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

This is what we call the net-out-flow of $s$. It is not surprising that the net-out-flow of $s$ is always the same as the net-in-flow of $t$. Intuitively this is because all other vertices have in-flow equals out-flow. So all the packages that $s$ sends out must end up at $t$. Formally, we have the following equation (try to prove it):

$$|f| = \sum_{v:(v,t)\in E} f(v,t) - \sum_{v:(t,v)\in E} f(t,v)$$
Here is a proof: There are two ways to write $\sum_{(u,v) \in E} f(u, v)$:

$$
\sum_{u \in V} \sum_{v:(u,v) \in E} f(u, v) = \sum_{u \in V} \sum_{w:(w,u) \in E} f(w, u)
$$

This implies that

$$
\sum_{u \in V} \text{out-flow} (u) = \sum_{u \in V} \text{in-flow} (u)
$$

As $\text{out-flow} (u) = \text{in-flow} (u)$ for all $u \in V - \{s, t\}$, we have

$$
\text{out-flow} (s) + \text{out-flow} (t) = \text{in-flow} (s) + \text{in-flow} (t)
$$

So the net-out-flow of $s$ is the same as the net-in-flow of $t$. 

Introduction
In the maximum flow problem, we are asked to find a flow $f$ that maximizes $|f|$. Before we present the Ford-Fulkerson method, it is worth pointing out that the restriction of $G$ being reduced (i.e., $(u, v) \in E$ implies $(v, u) \in E$) is without loss of generality.

Notation: Given a graph $G = (V, E)$, if both $(u, v) \in E$ and $(v, u) \in E$ then we call them two antiparallel edges.
To see this, let $G$ be a graph with antiparallel edges. We modify $G$ to get $G'$ as follows: For every two antiparallel edges $(u, v)$ and $(v, u) \in E$, add a new vertex $w$ and replace $(u, v)$ with $(u, w)$ and $(w, v)$. Also set $c(u, w) = c(w, v)$ to be the capacity $c(u, v)$ of the original edge $(u, v)$. It is clear that the new graph $G'$ has no antiparallel edges and thus, is reduced. Also $G'$ is essentially equivalent to $G$: a maximum flow in $G'$ has the same value as a maximum flow in $G$. (Actually, there is clearly a one-to-one correspondence between flows in $G'$ and flows in $G$.) This implies that any algorithm for finding a maximum flow in a reduced graph can be used to solve the same problem over general graphs.
We now describe the Ford-Fulkerson method. It is in some sense a greedy algorithm: Start with the zero flow: \( f(u, v) = 0 \) for all \((u, v) \in E\). Repeatedly increase the value of \( f \) by finding an “augmenting path” from \( s \) to \( t \) in the “residual graph” \( G_f \), until no such path exists. We will see that in each round, the value of \( f \) strictly increases. But the flow on a particular edge of \( G \) may increase or decrease! To describe the Ford-Fulkerson method, we need to define “residual graph” and “augmenting path”.
Let $f$ be a flow in $G$. The key idea is the following. Let

$$\langle v_0v_1\cdots v_k \rangle$$

be a sequence of vertices (not necessarily a path in $G$!) starting from $v_0 = s$ and ending at $v_k = t$. We call it a “good” sequence if it is simple (no vertex appears twice) and for each $i \in [0 : k - 1]$, one of the following two holds:

1. Either $(v_i, v_{i+1}) \in E$ and is not saturated:

   $$f(v_i, v_{i+1}) < c(v_i, v_{i+1})$$

2. Or $(v_{i+1}, v_i) \in E$ and $f(v_{i+1}, v_i)$ is positive

Introduction
Given a good $\langle v_0 v_1 \cdots v_k \rangle$, we can modify $f$ as follows: Let

$$\delta = \min_{i \in [0:k-1]} \left\{ \begin{array}{ll}
    c(v_i, v_{i+1}) - f(v_i, v_{i+1}) & \text{if } (v_i, v_{i+1}) \in E \\
    f(v_{i+1}, v_i) & \text{if } (v_{i+1}, v_i) \in E
  \end{array} \right.$$ 

Then 1) increase the flow $f(v_i, v_{i+1})$ of each $(v_i, v_{i+1}) \in E$ by $\delta$; and 2) decrease the flow $f(v_{i+1}, v_i)$ of each $(v_{i+1}, v_i) \in E$ by $\delta$. Denote the new flow by $f'$. We now show that the new flow $f'$ is still feasible and its value increases by $\delta$. To see this, first of all it is easy to check that $f'$ satisfies the capacity constraint:

$$0 \leq f'(u, v) \leq c(u, v), \quad \text{for all } (u, v) \in E$$
Also $f'$ satisfies the flow conservation property. For each $v_i$ in the sequence, where $i \in [1 : k - 1]$, we have the following four cases:

1. If $(v_{i-1}, v_i) \in E$ and $(v_i, v_{i+1}) \in E$, then both the in-flow and out-flow of $v_i$ increase by $\delta$

2. If $(v_{i-1}, v_i) \in E$ and $(v_{i+1}, v_i) \in E$, then both the in-flow and out-flow of $v_i$ remain the same

3. If $(v_i, v_{i-1}) \in E$ and $(v_i, v_{i+1}) \in E$, then both the in-flow and out-flow of $v_i$ remain the same

4. If $(v_i, v_{i-1}) \in E$ and $(v_{i+1}, v_i) \in E$, then both the in-flow and out-flow of $v_i$ decrease by $\delta$

Finally, it is easy to verify that $|f'| = |f| + \delta$. 
The message here is that to improve the value of $f$, sometimes we need to decrease the flow along an edge $(u, v) \in E$. This is kind of anti-intuitive so make sure to think it through before moving on. Now we can informally describe Ford-Fulkerson: Start with the zero flow; Repeatedly find a good sequence and use it to improve $f$, until no such sequence exists. To better describe this method, we introduce the concept of residual graphs.
Let $f$ be a flow in $G$. The residual graph $G_f = (V, E_f)$ with respect to $f$ has the following directed edges. Each edge in $E_f$ also has a positive residual capacity $c_f$ defined as follows:

1. **Forward edges:** $(u, v) \in E_f$ if $(u, v) \in E$ and is not saturated in $f$: $f(u, v) < c(u, v)$. The residual capacity of $(u, v) \in E_f$ is set to be $c_f(u, v) = c(u, v) - f(u, v)$. The residual capacity tells us how much we can increase the flow along $(u, v) \in E$.

2. **Reverse edges:** $(v, u) \in E_f$ if $(u, v) \in E$ and $f(u, v) > 0$. The residual capacity of $(v, u) \in E_f$ is $c_f(v, u) = f(u, v)$. The residual capacity tells us how much we can decrease the flow along the original edge $(u, v) \in E$. 

Introduction
It is clear that $G_f$ in general is not reduced, and has a lot of anti-parallel edges. Key observation: $\langle v_0 v_1 \cdots v_k \rangle$ is a good sequence if and only if it is a simple path from $s$ to $t$ in $G_f$. We will from now on refer to a simple path $p = \langle v_0 v_1 \cdots v_k \rangle$ from $s$ to $t$ in $G_f$ as an augmenting path. Let the residual capacity of $p$ be

$$c_f(p) = \min \left\{ c_f(u, v) : (u, v) \text{ is on } p \right\} > 0$$

Then we can modify $f$ to improve its value by $c_f(p)$, in the same way we did using a good sequence (again, an augmenting path is essentially a good sequence defined earlier, with a fancy name).
More exactly, for each edge \((v_i, v_{i+1}) \in E_f\) in \(p\), two cases:

1. If \((v_i, v_{i+1})\) is a forward edge, increase \(f(v_i, v_{i+1})\) by \(c_f(p)\)
2. If \((v_i, v_{i+1})\) is a reverse edge, decrease \(f(v_{i+1}, v_i)\) by \(c_f(p)\)

By the end we get a new flow \(f\) with its value increased by \(c_f(p)\). This gives us a round-by-round method to increase the value of the current flow \(f\). The million-dollar question is then the following: When Ford-Fulkerson stops, meaning there exists no augmenting path in the current residual graph \(G_f\), is \(f\) optimal? The answer is yes! The Ford-Fulkerson method always returns a maximum flow upon termination.
To prove it, recall that an $s$-$t$ cut of $G = (V, E)$ is a partition of $V$ into two sets $S$ and $T = V - S$ such that $s \in S$ and $t \in T$. Given a cut $(S, T)$, we define the capacity of $(S, T)$ to be

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

Minimum cut: an $s$-$t$ cut $(S, T)$ of minimum capacity. The first lemma we prove is simple:
Lemma

Max flow is \( \leq \) Min cut: \( \max f |f| \leq \min_{(S, T)} c(S, T) \).

Proof: Let \( f \) be a maximum flow in \( G \), and let \((S, T)\) be “any” \( s-t \) cut. Then it is easy to show (Prove it by yourself) that

\[
|f| = \sum_{(u,v) \in E: u \in S, v \in T} f(u, v) - \sum_{(u,v) \in E: u \in T, v \in S} f(u, v)
\]

\[
\leq \sum_{(u,v) \in E: u \in S, v \in T} c(u, v) = c(S, T)
\]

It follows that max flow is \( \leq \) min cut.
Also note that given $f$ and $(S, T)$, we have $|f| = c(S, T)$ if and only if $f(u, v) = c(u, v)$ for all $(u, v) \in E : u \in S, v \in T$ and $f(u, v) = 0$ for all $(u, v) \in E : u \in T, v \in S$. Now consider a flow $f$ in $G$ such that there is no augmenting path in $G_f$. This means $t$ is not reachable from $s$. Let $S$ denote the set of all vertices reachable from $s$, and $T = V - S$. It is clear that $(S, T)$ is an $s$-$t$ cut because $t \in T$. 

---

Introduction
As vertices in $T$ are not reachable from $S$, none of the edges in $E_f$ goes from a vertex in $S$ to a vertex in $T$. This implies that

1. For every $(u, v) \in E$ such that $u \in S$ and $v \in T$, $(u, v)$ must be saturated in $f$: $f(u, v) = c(u, v)$. Otherwise $(u, v) \in E_f$.

2. For every $(u, v) \in E$ such that $u \in T$ and $v \in S$, we must have $f(u, v) = 0$. Otherwise we have $(v, u) \in E_f$.

This implies that $|f| = c(S, T)$ and thus,

$$|f| = c(S, T) \geq \min_{(S', T')} c(S', T')$$

and $f$ is a max flow because $\max_f |f| \leq \min_{(S', T')} c(S', T')$. 

**Introduction**
We summarize it in the following Max-Flow Min-Cut theorem:

**Theorem**

**Max flow equals min cut:**

\[
\max_{f} |f| = \min_{(S, T)} c(S, T)
\]

Moreover, if \(f\) is a flow in \(G\) such that \(G_f\) has no augmenting path, then \(f\) must be a maximum flow.
Now we can describe the Ford-Fulkerson method formally:

1. set $f$ to be the zero flow
2. while there exists a simple path $p$ from $s$ to $t$ in $G_f$ do
3. use $p$ to modify $f$ and increase its value by $c_f(p)$
It stops within a finite number of rounds because each while loop, the value of $f$ increases by at least 1 (since we assumed that all the capacities are positive integers). If $f^*$ is a maximum flow in $G$, then Ford-Fulkerson executes the while loop at most $|f^*|$ times. So the total running time is $O((n + m) \cdot |f^*|)$ if we use BFS or DFS to find a path from $s$ to $t$ in the residual graph $G_f$ each round. As we assumed that all vertices are reachable from $s$,

$$m = |E| \geq |V| - 1 = n - 1$$

and thus, $O(n + m) = O(m)$ so the running time is $O(m \cdot |f^*|)$. 
It turns out that there are bad examples for which Ford-Fulkerson does need to execute the while loop for $\Omega(m \cdot |f^*|)$ many times. See one such example in Figure 26.7 on page 728. A more efficient implementation of Ford-Fulkerson, as we will see in the next class, is the Edmonds-Karp algorithm. The only difference is that in each while loop, we do not just pick an arbitrary augmenting path in $G_f$. Instead, we always pick one that minimizes the number of edges. We will show that by doing this, the while loop is executed at most $O(nm)$ times so the total running time is $O(nm^2)$. 