CS 218 Design and Analysis of Algorithms

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Module 2: Flow networks, Max-flow, Min-cut and applications

Ford-Fulkerson Algorithm

What all do we need to argue about this algorithm?

For each $e \in E$, set $f(e) \leftarrow 0$
Compute G_f
while There is an s to t path π in G_f **do**
  $f' \leftarrow \texttt{Aug}(\pi, f)$
  $f \leftarrow f'$
  Compute G_f
end while
Output f

Termination: Why does the algorithm terminate?

Time analysis: What is the running time of the algorithm?

Correctness: Why does it output the maximum flow?

Ford-Fulkerson Algorithm

What all do we need to argue about this algorithm?

```
For each e ∈ E, set f(e) ← 0
Compute G_f
while There is an s to t path π in G_f do
    f′ ← Aug(π, f)
    f ← f′
    Compute G_f
end while
Output f
```

Termination: Why does the algorithm terminate?

Time analysis: What is the running time of the algorithm?

Correctness: Why does it output the maximum flow?

Other issues worth discussing.

Is it important to choose the right s to t paths? Or would any path be okay? Any path is "ok" but TC might ↑

What is the relationship between Max-Flow and Min-Cut that we have been talking about all along?

Can the capacities be non-integral?

Termination of Ford-Fulkerson Algorithm

Recall that we have assumed that the capacities are integers.

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{\max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{\max}$.

The above three will suffice to prove the termination of the algorithm.

Termination of Ford Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Termination of Ford Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. *Then maximum flow* $\leq C_{max}$.

Proof.

Before the algorithm starts, the capacities are integral and flows are 0. Say it is true after j-th iteration.

As residual capacities are integral, the value of θ is also integral.

Thus the next flow f' is also integral.

Termination of Ford-Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Termination of Ford-Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Proof.

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

It must be a forward edge.

Its flow increases by $\theta(\pi, f)$. Flow on no other edge out of s changes.

Hence $|f'| = |f| + \theta(\pi, f)$.

Termination of Ford-Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Termination of Ford-Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Proof.

If all the edges out of s are saturated by flow f then $f = C_{\max}$.

No flow can exceed this value anyway.

Hence $C_{\max}$ is an upper bound on the maximum flow.

Termination of Ford-Fulkerson Algorithm

Lemma

At every intermediate stage in the algorithm, the flow values and the residual capacities stay integral.

If f is a flow and π is an s to t path in G_f, then $|f'| = |f| + \theta(\pi, f)$.

Let $C_{max} = \sum_{v \in V, (s,v) \in E} c(s, v)$. Then maximum flow $\leq C_{max}$.

Termination assuming the Lemma.

The flow only increases in each iteration of the algorithm.

It increases by at least 1 every time.

It cannot increase beyond $C_{\max}$.

Ford-Fulkerson Algorithm

What all do we need to argue about this algorithm?

For each $e \in E$, set $f(e) \leftarrow 0$
Compute G_f
while There is an s to t path π in G_f **do**
  $f' \leftarrow \texttt{Aug}(\pi, f)$
  $f \leftarrow f'$
  Compute G_f
end while
Output f

- ✓ Termination: Why does the algorithm terminate?
- Time analysis: What is the running time of the algorithm?
- Correctness: Why does it output the maximum flow?

Other issues worth discussing.

- Is it important to choose the right s to t paths? Or would any path be okay?
- What is the relationship between Max-Flow and Min-Cut that we have been talking about all along?
- Can the capacities be non-integral?

Time analysis of Ford Fulkerson Algorithm

Lemma

The running time of the algorithm is bounded by $O(C_{max} \cdot |E|)$.

The while loop runs for at most $C_{\max}$ iterations.

In each loop, to maintain residual graph we need $O(m)$ time.

Finding an s to t path π in this graph will need $O(m + n)$ times.

Augmenting takes time $O(n)$. (As $n - 1$ vertices along π.)

Ford-Fulkerson Algorithm

What all do we need to argue about this algorithm?

For each $e \in E$, set $f(e) \leftarrow 0$
Compute G_f
while There is an s to t path π in G_f **do**
  $f' \leftarrow \texttt{Aug}(\pi, f)$
  $f \leftarrow f'$
  Compute G_f
end while
Output f

✓ Termination: Why does the algorithm terminate?

✓ Time analysis: What is the running time of the algorithm?

Correctness: Why does it output the maximum flow?

Other issues worth discussing.

Is it important to choose the right s to t paths? Or would any path be okay?

What is the relationship between Max-Flow and Min-Cut that we have been talking about all along?

Can the capacities be non-integral?

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that

- A, B partition V.
- $s \in A$. Also $t \in B$. Why?
- When the algorithm terminates, no path from s to t in $G_{\mathfrak{f}}$.
- Therefore (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

- It saturates every edge from A to B.
- It avoids every edge from B to A.

Lemma

Let f be any flow in the flow network G. Let (S, T) be any (s, t)-Cut in G. Then $|f| \leq cap(S, T)$. Moreover, $|f| = cap(S, T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid$ there is a path to v from s in $G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that

- A, B partition V.
- $s \in A$. Also $t \in B$. Why?
- When the algorithm terminates, no path from s to t in $G_{\mathfrak{f}}$.
- Therefore (A, B) is a cut.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that

- A, B partition V.
- $s \in A$. Also $t \in B$. Why?
- When the algorithm terminates, no path from s to t in $G_{\mathfrak{f}}$.
- Therefore (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

It saturates every edge from A to B.

It avoids every edge from B to A.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that

- A, B partition V.
- $s \in A$. Also $t \in B$. Why?
- When the algorithm terminates, no path from s to t in $G_{\mathfrak{f}}$.
- Therefore (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

- It saturates every edge from A to B.
- It avoids every edge from B to A.

Lemma

Let f be any flow in the flow network G. Let (S, T) be any (s, t)-Cut in G. Then $|f| \leq cap(S, T)$. Moreover, $|f| = cap(S, T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

It saturates every edge from A to B.

It avoids every edge from B to A.

Lemma

Let f be any flow in the flow network G. Let (S, T) be any (s, t)-Cut in G. Then $|f| \leq cap(S, T)$. Moreover, $|f| = cap(S, T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Correctness of Ford Fulkerson Algorithm

Lemma

Let $\mathfrak{f}, A, B$ be as defined on the previous slide. Then $\mathfrak{f}$ saturates every edge from A to B.

Proof.

Suppose there is an edge $(u, v) \in E$ such that $u \in A$ and $v \in B$.

Suppose $\mathfrak{f}(u, v) < c(u, v)$.

Then (u, v) will be a forward edge in $G_{\mathfrak{f}}$.

But then $v \in A$ as per the construction of A. This is a contradiction.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

✓ It saturates every edge from A to B.

It avoids every edge from B to A.

Lemma

Let f be any flow in the flow network G. Let (S, T) be any (s, t)-Cut in G. Then $|f| \leq cap(S, T)$. Moreover, $|f| = cap(S, T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Correctness of Ford Fulkerson Algorithm

Lemma

Let $\mathfrak{f}, A, B$ be as defined on the previous slide. Then $\mathfrak{f}$ avoids every edge from B to A.

Proof.

Suppose there is an edge $(u, v) \in E$ such that $u \in B$ and $v \in A$.

Suppose $\mathfrak{f}(u, v) > 0$.

Then (v, u) will be a backward edge in $G_{\mathfrak{f}}$.

But then $u \in A$ as per the construction of A. This is a contradiction.

Correctness of Ford Fulkerson Algorithm

Notation.

Let $\mathfrak{f}$ be the flow returned by Ford-Fulkerson algorithm.

Let $A \leftarrow \{v \in V \mid \text{ there is a path to } v \text{ from } s \text{ in } G_{\mathfrak{f}}\}$.

Let $B \leftarrow V \setminus A$.

Note that (A, B) is a cut.

To show optimality, we need to show two things about $\mathfrak{f}$.

- ✓ It saturates every edge from A to B.
- ✓ It avoids every edge from B to A.

Lemma

Let f be any flow in the flow network G. Let (S, T) be any (s, t)-Cut in G. Then $|f| \leq \mathit{cap}(S, T)$. Moreover, $|f| = \mathit{cap}(S, T)$ if and only if f saturates every edge from S to T and avoids every edge from T to S.

Correctness of Ford Fulkerson Algorithm

Why is $\mathfrak{f}$ the maximum flow?

The algorithm terminates when there is no s to t path left in $G_{\mathfrak{f}}$.

This and all our analysis show that (A, B) cut has the same capacity as $|\mathfrak{f}|$.

As a result, $|\mathfrak{f}|$ is the maximum possible flow in G and $\mathrm{cap}(A, B)$ is the minimum capacity cut in G.

Ford-Fulkerson Algorithm

What all do we need to argue about this algorithm?

For each $e \in E$, set $f(e) \leftarrow 0$
Compute G_f
while There is an s to t path π in G_f **do**
  $f' \leftarrow \texttt{Aug}(\pi, f)$
  $f \leftarrow f'$
  Compute G_f
end while
Output f

- ✓ Termination: Why does the algorithm terminate?
- ✓ Time analysis: What is the running time of the algorithm?
- ✓ Correctness: Why does it output the maximum flow?

Other issues worth discussing.

Is it important to choose the right s to t paths? Or would any path be okay?

What is the relationship between Max-Flow and Min-Cut that we have been talking about all along?

Can the capacities be non-integral?

Which s to t path?

Consider the following example.