Lecture Notes for CS105 Algorithms

Network Flow – Edmonds-Karp

Thomas C. O'Connell

November 7, 2001

Edmonds-Karp Algorithm

So Ford-Fulkerson just says we should find repeatedly find augmenting paths and push flow along that path.

- So depending on how you go about finding augmenting paths, you'll get algorithms with potentially different running times (and may never terminate).
- The Edmonds-Karp algorithms says find the shortest augmenting path from s to t in the residual network using BFS. ↳ fewest # edges

Same algorithm just make sure you take a shortest path in G_f when **edge weights are all 1**:

Algorithm 0.0: Edmonds-Karp(G, s, t)**.**

```
for each (u, v) ∈ E
 do f[u, v] ← 0
    f[v, u] ← 0                          fewest # edges
while BFS finds a shortest path p from s to t in G_f
 do c_f(p) = min{c_f(u, v) : (u, v) ∈ p}
    for each (u, v) ∈ p
     do f[u, v] ← f[u, v] + c_f(p)
        f[v, u] ← −f[u, v]
return f.
```

Definition 0.1 *Let $\delta_f(s,v)$ be the number of edges on the shortest path from s to v in G_f.*

Lemma 0.2 (CLR 27.8) *Using Edmonds-Karp, $\delta_f(s,v)$ in G_f is nondecreasing with each flow augmentation.*

Proof.

- Suppose there is a flow augmentation that caused the shortest path from s to at least one vertex to decrease.
- Let f be the flow before this augmentation
- Let f' be the flow after this augmentation
- Let $X = \{x : \delta_{f'}(s,x) < \delta_f(s,x)\}$.
- Let v be a vertex in X such that $\delta_{f'}(s,v) \leq \delta_{f'}(s,x)$ for all $x \in X$. (i.e. v is the vertex in X with the shortest path from s after the augmentation)

G_f

$$s \rightarrow \cdots \rightarrow \cdots \rightarrow \cdots \rightarrow v$$

$$s \rightarrow \cdots \rightarrow \cdots \rightarrow \cdots \rightarrow x$$

$G_{f'}$

$$s \rightarrow \cdots \rightarrow v$$

$$s \rightarrow \cdots \rightarrow \cdots \rightarrow x$$

- Consider a shortest path from s to v in $G_{f'}$.
- Let u be the vertex preceeding v on this path.

  $G_{f'}$:

  $$s \rightarrow \cdots \rightarrow u \rightarrow v$$

  .

- $u \notin X$, since u is closer to s than v in $G_{f'}$.
- So $\delta_f(s,u)x \leq \delta_{f'}(s,u)$ since otherwise $u \in X$.

2 cases to consider:

1. $(u,v) \in E_f$
2. $(u,v) \notin E_f$

Case 1: $(u, v) \in E_f$

- v got closer to s but u didn't
- yet u now precedes v on the shortest path from s to v.

$$\begin{array}{c} G_f \\ s \to \cdots \to \cdots \to u \to v \end{array}$$

The last line follows since u preceeds v on the shortest path from s to v in $G_{f'}$.

So we must have $(u, v) \notin E_f$.

Case 2: $(u,v) \notin E_f$

- Let p be the augmenting path that produces $G_{f'}$.
- p is a shortest path from s to t.
- p must contain (v,u) (in the direction from v to u) since $(u,v) \in E_{f'}$ and $(u,v) \notin E_f$.

$$p : s \to \cdots \to v \to u \to \cdots \to t$$

.

- So we have:

G_f

$s \;\to\; \cdots \to \cdots \to v \to u$ if u is even reachable

$G_{f'}$

$s \;\to\; \cdots \to \cdots \to u \to v$ is now a shortest path

- this doesn't make sense.

Formally:

$$\begin{aligned} \delta_f(s,v) &\leq \delta_f(s,u) + 1 \\ &\leq \delta_{f'}(s,u) + 1 \\ &= \delta_{f'}(s,v) \qquad \text{contradiction} \end{aligned}$$

The last line follows since u preceeds v on the shortest path from s to v in $G_{f'}$.
So we must have $(u,v) \notin E_f$.

Case 2: $(u, v) \notin E_f$

- Let p be the augmenting path that produces $G_{f'}$.
- p is a shortest path from s to t.
- p must contain (v, u) (in the direction from v to u) since $(u, v) \in E_{f'}$ and $(u, v) \notin E_f$.

$$p : s \to \cdots \to v \to u \to \cdots \to t$$

.

- So we have:

G_f

$s \to \cdots \to \cdots \to v \to u$ is a shortest path inG_f

$G_{f'}$

$s \to \cdots \to u \to v$ is now a shortest path

- So v has moved closer, u has not, yet u moved in front of v on a shortest path.
- This is impossible:

  Formally:

$$\begin{aligned}
\delta_f(s, v) &= \delta_f(s, u) - 1 && (v, u) \text{ is on a shortest path in } G_f \\
&\leq \delta_{f'}(s, u) + 1 && \\
&= \delta_{f'}(s, v) - 2 && (u, v) \text{ is on a shortest path in } G_{f'} \\
&< \delta_{f'}(s, v) && \text{contradiction}
\end{aligned}$$

Since we have contradictions in either case, no flow augmentation using Edmonds-Karp can cause the shortest path from s to any vertex to decrease. ∎

Definition 0.3 *An edge* (u, v) *on an augmenting path* p *in* G_f *is called* **critical** *if* $c_f(p) = c_f(u, v)$.

Example: (y, z) is critical on this augmenting path:

$$s \xrightarrow{2} x \xrightarrow{3} y \xrightarrow{1} z \xrightarrow{2} t$$

How many times can an edge (u, v) be critical in the execution of Edmonds-Karp.

- Since augmenting paths are SP's:

$$\delta_f(s, v) = \delta_f(s, u) + 1$$

  when (u, v) is critical for the first time.

- After the flow is augmented, (u, v) disappears.
- It cannot reappear until the net flow from u to v decreases.
- This happens only when (v, u) appears on an augmenting path.
- Let f' be the flow when this occurs, then:

$$\delta_{f'}(s, u) = \delta_{f'}(s, v) + 1$$

- By the lemma, the distance from s is nondecreasing with each flow augmentation so:

$$\delta_f(s, v) \leq \delta_{f'}(s, v)$$

- But then we have:

G_f

$s \rightarrow \cdots \rightarrow u \rightarrow v$ is a shortest path in G_f

$G_{f'}$

$s \rightarrow \cdots \rightarrow .. \rightarrow v \rightarrow u$ is a shortest path in $G_{f'}$

  So u has moved a distance of at least 2 futher away from s.

  Formally:

$$\begin{aligned} \delta_{f'}(s, u) &= \delta_{f'}(s, v) + 1 \\ &\geq \delta_f(s, v) + 1 \\ &= \delta_f(s, u) + 2 \end{aligned}$$

- So from the time that (u, v) becomes critical to the next time it becomes critical the distance from s to u increases by at least 2.

- This distance starts at 1 and once it gets to $|V| - 2$, u becomes unreachable.
- Thus (u, v) can be critical at most $O(V)$ times.
- Since there are $O(E)$ pairs of vertices that can have an edge betwen them in a residual network, the total number of critical edges during the entire execution is $O(VE)$.
- Each augmenting path has at least one critical edge so there can be at most $O(VE)$ augmenting paths during the entire execution of the algorithm.

Running Time: $O(VE^2)$.

- Each BFS takes $O(V + E) = O(E)$ time since the graph is assumed to be connected.
- Inner loop take $O(E)$ time.
- We just proved that there are $O(VE)$ iterations of outer loop.