Edmonds-Karp algorithm

To improve the $O(mC)$ bound of the Ford-Fulkerson algorithm, we can be careful while choosing augmenting paths. The Edmonds-Karp algorithm chooses a *shortest* augmenting $s-t$ path where each edge has unit distance, ie an $s-t$ path with the fewest number of edges.

For the purpose of analysis, let $d_f[v]$ be the shortest distance(number of edges) of the vertex v from source s in the residual graph G_f.

Lemma 1. *During the course of the algorithm, $d_f[\cdot]$ value of any vertex does not decrease.*

Proof. We use proof by contradiction. Let G_f be the first residual graph in which the shortest distance from s of some vertex decreased and let v be such a vertex in G_f which is closest to s, ie the one with minimum $d_f[v]$. Let $G_{f'}$ be the flow network just before G_f. Then,

$$d_f[v] < d_{f'}[v] \tag{1}$$

Let $s \rightsquigarrow u \to v$ be the shortest $s-v$ path in G_f. Then,

$$d_f[v] = d_f[u] + 1 \tag{2}$$

Because v was the closest vertex to s in G_f whose shortest distance decreased, $d_f[u]$ did not decrease, ie,

$$d_f[u] \geq d_{f'}[u] \tag{3}$$

Now, $(u,v) \notin E_{f'}$. Because if (u,v) was an edge in $G_{f'}$,

$$\begin{aligned} d_{f'}[v] &\leq d_{f'}[u] + 1 \\ &\leq d_f[u] + 1 && \text{(from equation 3)} \\ &= d_f[v] && \text{(from equation 2)} \end{aligned}$$

But this contradicts our assumption made in equation 1. Thus $(u,v) \notin E_{f'}$. But since $(u,v) \in E_f$, it must have been that in $G_{f'}$, our algorithm chose an augmenting path that contained the edge (v,u). Since we always choose shortest paths,

$$\begin{aligned} d_{f'}[u] &= d_{f'}[v] + 1 \\ d_{f'}[v] &= d_{f'}[u] - 1 \\ &\leq d_f[u] - 1 && \text{(from equation 3)} \\ &= d_f[v] - 2 && \text{(from equation 2)} \\ d_{f'}[v] &< d_f[v] \end{aligned}$$

This contradicts our initial assumption(equation 1). Thus such a vertex v cannot exist. □

Lemma 2. *Let (u,v) be the bottleneck edge in G_f. Then the next time (u,v) becomes a bottleneck edge, $d[u]$ increases by at least 2.*

Proof. Since (u,v) is chosen as the bottleneck edge in G_f and we choose shortest paths,

$$d_f[v] = d_f[u] + 1 \tag{4}$$

In all subsequent residual graphs, the edge (u,v) will not be present unless the algorithm augments flow along the edge (v,u). Let $G_{f'}$ be the first such residual graph. Then,

$$d_{f'}[u] = d_{f'}[v] + 1 \tag{5}$$

By the previous lemma, we have,

$$d_{f'}[v] \geq d_f[v] \tag{6}$$

Using equation 6 in equation 5 we get,

$$\begin{aligned} d_{f'}[u] &\geq d_f[v] + 1 \\ &= d_f[u] + 2 && \text{(from equation 4)} \end{aligned}$$

□

Theorem 3. *Edmonds-Karp algorithm takes $O(m^2 n)$ time to compute max-flow in a network.*

Proof. Initially the distance of a node u from s is at least 0 and if distance of u from s becomes more than n, it is unreachable from s(since we always choose a simple path for augmentation). Also, between each time the edge (u, v) becomes a bottleneck edge, $d[u]$ increases by at least 2. Thus each edge can become bottleneck edge at most $n/2$ times. Since there are $O(m)$ edges, and each augmenting path has at least one bottleneck edge, it follows that the total number of augmenting paths is at most $O(mn)$. To find one augmenting path, it takes $O(m)$ time using breadth-first search. Thus the total running time is $O(m^2 n)$. □