CS 6200: Algorithmics II	**Fall 2020**
Instructor: Avah Banerjee, S&T.	
Lectures: 10	**Dates: 10/16**

1 Edmonds-Karp algorithm

The Edmonds-Karp algorithm is an implementation of the Ford-Fulkerson algorithm for finding the maximum flow in a flow network. Rather than considering an arbitrary path as an augmented path, as done in the Ford-Fulkerson algorithm, the Edmonds-Karp algorithm considers the path with the minimal length (shortest path) as an augmented path in the residual graph. It identifies such a path using the breadth-first search technique.

Algorithm 1 Edmonds-Karp(G, s, t)

```
initialize flow f ← 0
while there exists an augmenting path p in the residual network G_f do
    choose augmented path in the flow using Breadth First Search technique
    update the flow along G_f
end while
return f
```

Let m=number of edges and n=number of vertices in a graph that is given as input to the above algorithm. The run time complexity of the Edmonds-Karp algorithm is $O(nm^2)$. In the later part of the document, we will be proving the same.

We know that for any residual graph the time complexity for finding the shortest path using the breadth-first technique is $O(m)$. Hence, our aim is reduced to prove that the total possible augmentations that can occur are $O(mn)$.

Definition 1.1 *A path (u, v) in a network flow G_f is called critical when $c_f(u, v)$ is minimal along all the edges in the augmented path containing (u, v).*

Lemma 1.1 *If the Edmonds-Karp algorithm is run on a graph $G = (V, E)$ with s, t as a source and sink then the shortest distance $\delta_f(s, v)$ monotonically increases with a sequence of augmentation.*

Proof: Let us prove by this by contradiction. Let us assume that for a vertex v in the graph,a augmentation has decreased the shortest-path distance from s to v to decrease. Let f be the initial flow and f' be the flow after augmentation. So, $\delta'_f(s, v) < \delta_f(s, v)$. Consider an vertex u in the flow which is adjacent to v in the shortest path (E'_f). Then in the flow f' we have,

$$\delta'_f(s, u) = \delta_f(s, v) - 1 \tag{1}$$

and we have,

$$\delta'_f(s, u) \geq \delta_f(s, u) \tag{2}$$

because of how we choose v,we know that distance between vertex u and source did not decrease. For $(u, v) \in E_f$, it should follow

$$\begin{aligned} \delta_f(s,v) \leq &\quad \delta_f(s,u)+1 \quad \text{(from triangular inequality)} \\ \delta_f(s,v) \leq &\quad \delta_{f'}(s,u)+1 \qquad \text{(from Equation (2))} \\ \delta_f(s,v) = &\qquad \delta_{f'}(s,v) \qquad\quad \text{(from Equation (1))} \end{aligned} \tag{3}$$

which contradicts our initial assumption. Hence $(u, v) \notin E_f$. As we can see $(u, v) \notin E_f$, but $(u, v) \in E_{f'}$ and we also know that Edmonds-Karp algorithm always has shortest paths in the augmented flow, this augmentation must have increased flow from v to u. And the shortest path from s to u had (v, u) as its last stage. Therefore,

$$\begin{aligned} \delta_f(s,v) &= \delta_f(s,u)-1 \\ &\leq \delta_{f'}(s,u)-1 \quad \text{(From Equation (2))} \\ &= \delta_{f'}(s,v)-2 \quad \text{(From Equation (1))} \end{aligned} \tag{4}$$

Therefore, from the time (u, v) becomes critical to the next time the distance between u and s will increase at least by 2. ■

As we know the maximum distance between any two vertices in a graph is bounded by $m-1$ (number of edges). By considering the previous statement, the number of maximum possible augmentations is less than $m/2$. As there are n vertices in a graph. The run time complexity of the algorithm is $O(mn)$. From the previous assertion, we concluded that the run-time of BFS is $O(m)$. When combining both the results for the Edmonds-Karp algorithm we can conclude the final run-time is $O(nm^2)$

2 Max-Flow Min-cut Theorem

Theorem 2.1 *Below three statements are equivalent:*

1. *f is a maximum flow in G*
2. *G_f does not have any shortest path (augmenting path)*
3. *$|f| = C^*(s,t)$ for some $s-t$ cut, where $C^*(s,t)$ is also a minimum $s-t$ cut.*

Proof: First let us show the equivalence of Statement (1) and (2). (1) $\implies$ (2): Proof by contradiction $\neg(2) \implies \neg(1)$

Suppose there is an augmenting path p in flow f then,

$$|f \uparrow f_p| = |f| + |f_p| > |f| \tag{5}$$

It implies that there exists a flow greater than f, which is contradiction to statement (1).

(2) $\implies$ (3) : Suppose G_f does not have an augmenting path, Nw let us divide the vertices in the flow into two sets $S = \{v|$ there exists a path from s to v in $G_f\}$ and

T=V - S that is the remaining vertices in the graph.

This partition can be a cut because the source s $\in S$ and $t \notin S$ as we know that there is no path between s and t(statement 2).

(u, v) such that $u \in S$ and $v \in T$. Hence $(u, v) \notin E_f$

1. If $(u, v) \in E$.then $f(u, v) = c(u, v)$ the reason being, if the edge (u, v) is removed E_f that means that it was a critical path in the original graph E.Hence, the flow along the path containing (u, v) is the equal to the capacity of the edge (u, v)

2. Similarly, if $(v, u) \in E$ then we know $c_f(u, v) = f(v, u)$.As we do not have $(u, v) \in E_f$ the capacity is 0 and hence $f(v, u) = 0$ The total flow of the cut

$$\begin{aligned} f(S,T) &= \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{v\in T}\sum_{u\in S} f(v,u) \\ &= \sum_{u\in S}\sum_{v\in T} c(u,v) - \sum_{u\in S}\sum_{v\in T} 0 \\ &= c(S,T) \end{aligned} \tag{6}$$

Therefore , $|f| = f(S,T) = c(S,T)$

(3) $\implies$ (1) : Assume $|f| \leq C(S,T)$.

Since, $|f| = C(S,T)$(statement 3) then it must be the maximum flow. ■

Thus,we conclude that the above 3 statements are equivalent.