DAA-Notes

220001077, 220001071, 220001070

4th April 2024

1 Max Flow - Min Cut Theorem

The theorem states that, If f is the flow in the Flow network G = *(V,E)* having a source vertex s and sink vertex t, then the following conditions are equivalent:
1. f is a maximum flow in G.
2. The residual network G_f contains no augmenting paths.
3. $|f|$ = *c(S,T)* for some cut *(S,T)* of G.

Proof.

1 $\implies$ 2. Suppose for the sake of contradiction that f is a maximum flow in G but that G_f has an augmenting path p. Then, from $|f \uparrow f_p| = |f| + |f_p| > |f|$, the flow found by augmenting f by f_p, where f_p is given by

$$f_p(u,v) = \begin{cases} c_f(p) & \text{if } (u,v) \text{ is on } p, \\ b & \text{otherwise} \end{cases}$$

is a flow in G with value strictly greater than $|f|$, contradicting the assumption that f is a maximum flow.

2 $\implies$ 3. Suppose that G_f has no augmenting path, that is, that G_f contains no path from s to t. Define

$$S = \{v \in V : \textbf{There exists a path from s to v in } G_f\}$$

$$T = \{v \in V : \textbf{There does not exists a path from s to v in } G_f\}$$

The partition (S,T) is a cut: we have $s \in S$ trivially and $t \in T$ because there is no path from s to t in G_f. Now consider a pair of vertices $u \in S$ and $v \in T$. If $(u,v) \in E$, we must have $f(u,v) = c(u,v)$, since otherwise $(u,v) \in E_f$, which would place v in set S. If $(v,u) \in E$, we must have $f(v,u) = 0$, because otherwise $c_f(u,v) = f(v,u)$ would be positive and we would have $(u,v) \in E_f$, which again would place v in S. Of course, if neither (u,v) nor (v,u) belongs to E, then $f(u,v) = f(v,u) = 0$. We thus have

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) \leq \sum_{u \in S} \sum_{v \in T} c(u,v) = c(S,T).$$

From previous classes we that the net flow across *(S,T)* is *f(S,T)* $= |\ f\ |$, therefore, $|f| = f(S,T) \leq c(S,T)$.

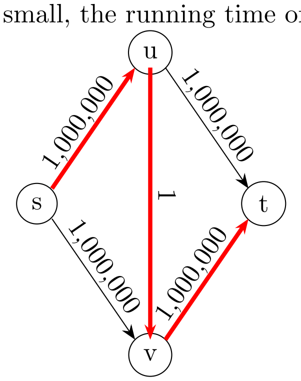
3 $\Longrightarrow$ **1** From previous classes we also know that the value of any flow f in a flow network G is bounded from above by the capacity of any cut of G so, $|f| \leq c(S,T)$ for all cuts (S,T). The condition $|f| = c(S,T)$ thus implies that f is a maximum flow.

The Ford-Fulkerson algorithm keeps on finding augmenting paths in each previously modified flow network, and if it is able to find augmenting path then updates the flow.It continues this until it is not able to find another augmenting path of non-zero flow.

2 Analysis of Ford- Fulkerson Algorithm

In the worst case, if the edge capacities are irrational numbers, it's possible to choose the augmenting path so that the algorithm never terminates:

However, When the capacities are integers and the optimal flow value $|f^*|$ is small, the running time of the Ford-Fulkerson algorithm is good.



The above example shows a flow network for which $|f^*|$ is large. A maximum flow in this network has a value of 2,000,000: 1,000,000 units of flow traverse the path $s \to u \to t$, and another 1,000,000 units traverse the path $s \to v \to t$. If the first augmenting path found by FORD-FULKERSON is $s \to u \to v \to t$, as shown above, The flow has value 1 after the first iteration. If the second iteration finds the augmenting path $s \to v \to u \to t$, the flow then has value 2. If the algorithm continues alternately choosing the augmenting paths $s \to u \to v \to t$ and $s \to v \to u \to t$, it performs a total of 2,000,000 augmentations, increasing the flow value by only 1 unit in each.