Net Flow Across Cut

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

Proof. Follows directly from the definition of the flow, see Lemma 26.4

16

Upper Bound on Flow Value

Let (S,T) be a cut in a flow network $G = (V,E)$, and let f be a flow in G. Then

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

This inequality in particular holds if (S,T) is a minimum capacity cut.

17

Max-Flow Min-Cut Theorem

Let f be a flow in a flow network $G = (V,E)$ with source s and sink t. The following statements are equivalent:

1. f is a maximum flow in G.
2. The residual network G_f has no augmenting paths.
3. $|f| = c(S,T)$ for some cut (S,T) in G.

18

(1) ===> (2)

Suppose that f is a maximum flow in G while G_f has an augmenting path P.

Hence there is a non-zero flow f' in G_f (along the edges of P), zero elsewhere.

Hence $f + f'$ is a flow in G with value strictly greater than $|f|$.

Contradiction.

19

(2) ===> (3)

Suppose that G_f has no augmenting path.

Let $S = \{ v\in V$: there is a path from s to v in $G_f \}$

Let $T = V\text{-}S$.

(S,T) is a cut.

Let $u \in S$, $v \in T$

If (u,v) is in E, then $f(u,v) = c(u,v)$; otherwise (u,v) would be an edge in G_f contradicting the definition of (S,T).

If (v,u) is in E, then f(v,u) = 0; otherwise (u,v) would be an edge in G_f contradicting the definition of (S,T).

Hence, $|f| = f(S,T) = c(S,T)$.

20

(3) ===> 1

We have shown that $|f| \le c(S,T)$ for all cuts (S,T).

$|f| = c(S,T)$ implies therefore that f is a maximum flow.

21

Ford-Fulkerson Algorithm

initialize flow f to 0-flow.

construct the residual network G_f (trivial for 0-flow).

while there is a flow augmenting path in G_f do

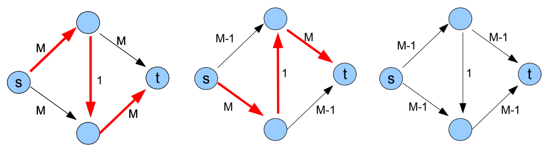
augment f by pushing as much as possible through the augmenting path.

construct the residual network for the increased flow.

Ford-Fulkerson - Complexity

Searching for an augmenting path: $O(V+E') = O(E)$, use for example depth-first search.

If capacities are integral, there can be as many as | f* | iterations

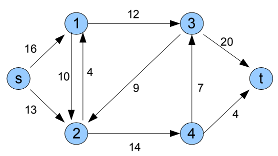


Edmonds-Karp Algorithm

Use breadth-first search!!!

This variant of Ford-Fulkerson algorithm runs in $O(nm^2)$.

Click to add title



Lemma 1

$\Delta_f(v)$ = minimum number of edges that have to be traversed from s to a vertex v in G_f.

Claim: $\Delta_f(v)$ increases monotonically with each flow augmentation for every v in G_f

Proof of Lemma 1

By contradiction.

Let f' denote the flow after the first Δ-decreasing flow augmentation. Let v denote the vertex with the smallest decreased $\Delta_{f'}$ value and let (u,v) be the edge on the edge-minimal path to v in $G_{f'}$. Let f denote the flow just before f'. We know that $\Delta_{f'}(v) < \Delta_f(v)$, and

$\Delta_{f'}(u) = \Delta_{f'}(v) - 1$

$\Delta_{f'}(u) \geq \Delta_f(u)$

Assume that (u,v) is in G_f

$\Delta_f(v) \leq \Delta_f(u) + 1 \leq \Delta_{f'}(u) + 1 = \Delta_{f'}(v)$

Proof of Lemma 1 - Continued

Hence (u,v) is in $G_{f'}$ but not in G_f

This is only possible if the augmentation of f increased the flow from v to u.

Edmonds-Karp algorithm augments along shortest paths. Therefore

$\Delta_f(v) = \Delta_f(u) - 1 \leq \Delta_{f'}(u) - 1 = \Delta_{f'}(v) - 2$

This contradicts our assumption that $\Delta_{f'}(v) < \Delta_f(v)$

28

Lemma 2

An edge (u,v) on the augmenting path P in G_f is critical if the residual capacity of P is equal to the residual capacity of (u,v).

Claim: An edge (u,v) can be critical at most $n/2 - 1$ times.

Proof: When (u,v) is critical on an augmenting path P, we must have $\Delta_f(v) = \Delta_f(u) + 1$.

When the flow is augmented along P, (u,v) disappears from the residual network.

It reapears when (v,u) is on the augmenting path for some flow f' and $\Delta_{f'}(u) = \Delta_{f'}(v) + 1$

$\Delta_{f'}(u) = \Delta_{f'}(v) + 1 \geq \Delta_f(v) + 1 = \Delta_f(u) + 2$

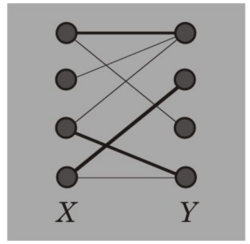
$\Delta_f(u)$ is at most n-2 ((u,v) being critical implies that $u \neq t$)

(u,v) can be critical at most $(n-2)/2$ times

29

Bipartite Graphs

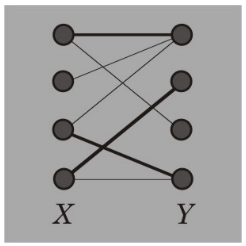
A graph $G = (V,E)$ is bipartite if its vertices can be partitioned into two subsets X and Y such that every edge connects a vertex in X with a vertex in Y.



30

Maximum Matching in Graphs

A matching is a subset of edges M in E such that each vertex v in V is incident with at most one edge of M. A maximum matching is a matching with the maximum number of edges.



31

Relating Flow to Matching in Bipartite Graphs

Add source vertex and connect it to all vertices in X

Add sink vertex and connect all vertices in Y to it.

Unit capacities for all edges.

32

Matching Defines Integral Flow

Bipartite graph $G = (V,E)$.

Flow network $G' = (V',E')$.

If M is a matching in G then there is an integral flow f in G' of value $|f| = |M|$.

Proof: For every edge (u,v) in M, let $f(s,u) = f(u,v) = f(v,t) = 1$. For all other edges (u,v) in E', let $f(u,v) = 0$.

Check that f satisfies capacity constraints and flow conservation.

The paths through the edges of matching are vertex disjoint (apart from s and t). It is obvious that $|f| = |M|$ and there is integer flow through each edge.

33

Integral Flow Defines Matching

Integral flow network $G' = (V', E')$.

Bipartite graph $G = (V, E)$.

If f is a flow in G' of value $|f|$ then M is a matching in G, $|M| = |f|$.

Proof. Unit capacities and integrality of flow ensures that only one unit of flow can enter a vertex of X. Hence this unit of flow must leave such a vertex through exactly one edge.

Similarly only one unit of flow can leave a vertex of Y. Hence this unit of flow can enter such a vertex through exactly one edge.

Let M be the edges from X to Y with unit flow.

M is a matching.

$$|M| = f(X,Y) = f(X,V') - f(X,X) - f(X,s) - f(X,t) = 0 - 0 + f(s,X) - 0 = f(s,V') = |f|$$

Max Matching Defines Max Flow
Max Flow Defines Max Matching

Follows immediately if we can show that max flow algorithm returns integral flow when capacities are integer.

Easy induction proof, see exercise 26.3-2