Lecture22_NetworkFlows1

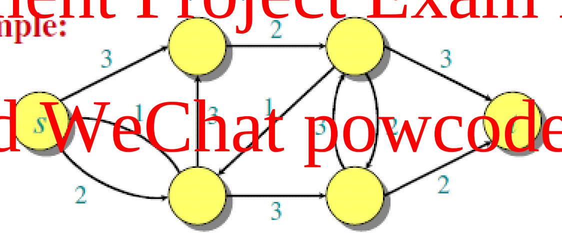
Wednesday, October 21, 2020 3:31 PM

Matching problem

Flow Networks

Definition. A flow network is a directed graph G = (V, E) with two distinguished vertices: a source s and a sink t. Each edge (u, v) ∈ E has a nonnegative capacity c(u, v). If (u, v) ∉ E, then c(u, v) = 0.

Example:



Definition. A *positive flow* on G is a function $p : V \times V \to \mathbb{R}$ satisfying the following:

- *Capacity constraint:* For all $u, v \in V$, $0 \le p(u, v) \le c(u, v)$.
- *Flow conservation:* For all $u \in V - \{s, t\}$,

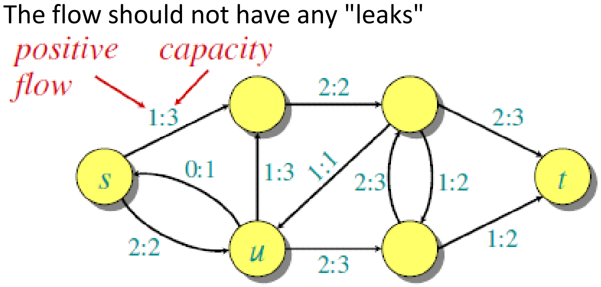
$$\sum_{v \in V} p(u, v) - \sum_{v \in V} p(v, u) = 0 .$$

The flow goes into the vertex should equal to the flow goes out from that vertex

The *value* of a flow is the net flow out of the source:

$$\sum_{v \in V} p(s, v) - \sum_{v \in V} p(v, s) .$$

- The flow should not have any "leaks"



Flow conservation (like Kirchoff's current law):

- Flow into u is $2 + 1 = 3$.
- Flow out of u is $0 + 1 + 2 = 3$.

The value of this flow is $1 - 0 + 2 = 3$.

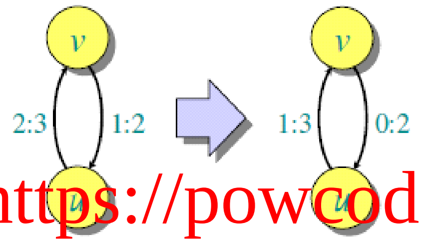
- The positive flow is not greater than the capacity
- Flow conservation: in == out
- Sink vertex doesn't need to have flow conservation
- The nature of the problem: at some edges we are wasting our capacity
- ➢ Find the maximum assignment

Maximum-flow problem: Given a flow network G, find a flow of maximum value on G.

➢ The value of the maximum flow of the above graph is 4.

Flow cancellation

Without loss of generality, positive flow goes either from u to v, or from v to u, but not both.



Net flow from u to v in both cases is 1.

The capacity constraint and flow conservation are preserved by this transformation.

INTUITION: View flow as a *rate*, not a *quantity*.

Notational simplification

IDEA: Work with the net flow between two vertices, rather than with the positive flow.

Definition. A ***net flow*** on G is a function $f : V \times V \to \mathbb{R}$ satisfying the following:

- ***Capacity constraint:*** For all $u, v \in V$, $f(u, v) \le c(u, v)$.

- ***Flow conservation:*** For all $u \in V - \{s, t\}$, $\boxed{\sum_{v \in V} f(u,v) = 0.}$ ← *One summation instead of two.*

- ***Skew symmetry:*** For all $u, v \in V$, $f(u, v) = -f(v, u)$.

Equivalence of definitions

Theorem. The two definitions are equivalent.

Proof. ($\Rightarrow$) Let $f(u, v) = p(u, v) - p(v, u)$.

- ***Capacity constraint:*** Since $p(u, v) \le c(u, v)$ and $p(v, u) \ge 0$, we have $f(u, v) \le c(u, v)$.
- ***Flow conservation:***

$$\sum_{v \in V} f(u,v) = \sum_{v \in V} (p(u,v) - p(v,u)) = \sum_{v \in V} p(u,v) - \sum_{v \in V} p(v,u)$$

- ***Skew symmetry:***

$$f(u, v) = p(u, v) - p(v, u) = -(p(v, u) - p(u, v)) = -f(v, u).$$

• If given a network flow, can we create a flow that satisfies its two properties

($\Leftarrow$) Let

$$p(u,v) = \begin{cases} f(u,v) & \text{if } f(u,v) > 0, \\ 0 & \text{if } f(u,v) \le 0. \end{cases}$$

- *Capacity constraint:* By definition, $p(u, v) \ge 0$. Since $f(u, v) \le c(u, v)$, it follows that $p(u, v) \le c(u, v)$.
- *Flow conservation:* If $f(u, v) > 0$, then $p(u, v) - p(v, u) = f(u, v)$. If $f(u, v) \le 0$, then $p(u, v) - p(v, u) = -f(v, u) = f(u, v)$ by skew symmetry. Therefore,

$$\sum_{v \in V} p(u,v) - \sum_{v \in V} p(v,u) = \sum_{v \in V} f(u,v). \quad \square$$

- From summation notation to **set notation**

Definition. The *value* of a flow f, denoted by $|f|$, is given by

$$|f| = \sum_{v \in V} f(s,v) = f(s,V).$$

set summation

Implicit summation notation: A set used in an arithmetic formula represents a sum over the elements of the set.

- **Example** — flow conservation: $f(u, V) = 0$ for all $u \in V - \{s, t\}$.

all vertices except source and sink

Simple properties of flow

Lemma.
- $f(X, X) = 0$,
- $f(X, Y) = -f(Y, X)$,
- $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$ if $X \cap Y = \varnothing$. $\square$

Theorem. $|f| = f(V, t)$.

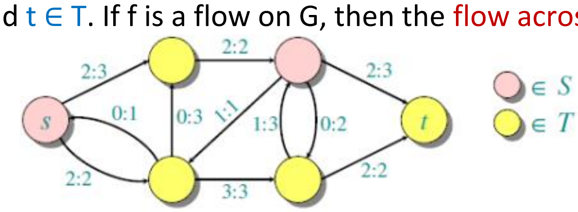
Proof.

$$\begin{aligned} |f| &= f(s, V) && \text{lemma 1} \\ &= f(V, V) - f(V-s, V) && \textit{Omit braces.} \\ &= f(V, V-s) && \text{lemma 2} \\ &= f(V, t) + \underline{f(V, V-s-t)} && \text{lemma 3} \\ &= f(V, t). \quad \square && \text{conservation of flow} \end{aligned}$$

- Whatever we send to the network should end up in the sink

Cuts

Definition. A cut (S, T) of a flow network $G = (V, E)$ is a partition of V such that $s \in S$ and $t \in T$. If f is a flow on G, then the flow across the cut is $f(S, T)$.



$$f(S,T) = (2+2) + (-2+1-1+2) = 4$$

ch 3, 2009 COT5405 Analysis of Algorithms 13

- Split the graph to two side: source to one side, and sink to another side

Another characterization of flow value

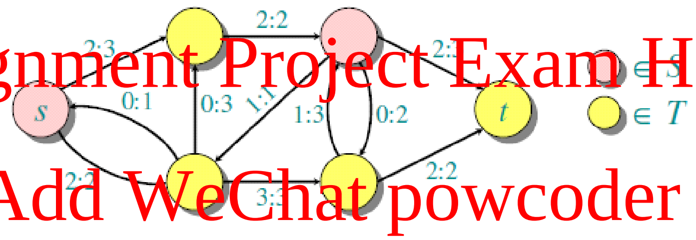
Lemma. For any flow f and any cut (S, T), we have | f | = f (S, T).

Proof.
$$\begin{aligned} f(S,T) &= f(S,V) - f(S,S) \\ &= f(S,V) \\ &= f(s,V) + f(S-s,V) \\ &= f(s,V) \\ &= |f|. \quad \square \end{aligned}$$

- T = v - s

Capacity of a cut

Definition. The capacity of a cut (S, T) is c(S, T).



$$\begin{aligned} c(S,T) &= (3+2) + (1+2+3) \\ &= 11 \end{aligned}$$

Upper bound on the maximum flow value

Theorem. The value of any flow is bounded above by the capacity of any cut.

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

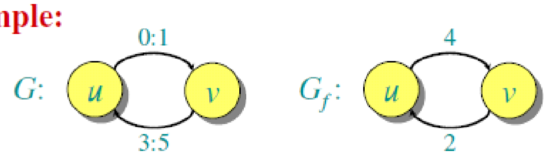
Residual network

Definition. Let f be a flow on G = (V, E). The residual network G_f (V, E_f) is the graph with strictly positive residual capacities

c_f (u, v) = c(u, v) – f (u, v) > 0.

Edges in E_f admit more flow.

Example:

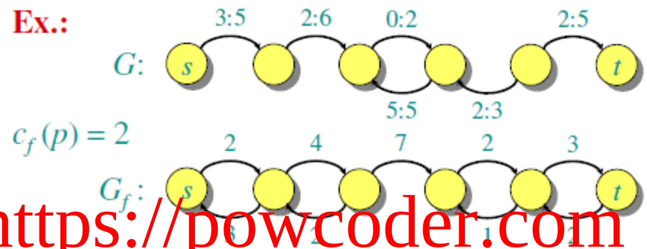


Lemma. $|E_f| \le 2|E|$. $\square$

Augmenting paths

Definition. Any path from s to t in G_f is an ***augmenting path*** in G with respect to f. The flow value can be increased along an augmenting path p by $c_f(p) = \min_{(u,v)\in p} \{c_f(u,v)\}$.

Ex.:



- Any path from source to sink found in the residual path.
- The graph above is not entire graph, therefore there are some node without conservation int the network flow

Max-flow, min-cut theorem

Theorem. The following are equivalent:

1. f is a maximum flow.
2. f admits no augmenting paths.
3. | f | = c(S, T) for some cut (S, T).