CS412 - Maximum Flows

Source: CLRS

Think of a flow on graphs as a bandwidth problem, or a current flow (recall, Kirchoff's Laws) and so on.

The Flow Network: A flow network is a directed graph $G = (V, E)$ with each edge $(u,v) \in E$ having a non-negative capacity $c(u,v) \geq 0$.

Also, if $(u,v) \in E$, then there is no edge (v,u) in E, i.e., $(v,u) \notin E$. Self-loops are not allowed.

Identify two special nodes (or vertices) in a flow network: a source node, say 's' and a sink node 't'. (say).

For convenience, we assume that every vertex $v \in V \setminus \{s,t\}$ is on a path from 's' to 't', or $s \overset{v}{\rightsquigarrow} t$, $\forall v \in V \setminus \{s,t\}$ [or $s \rightsquigarrow v \rightsquigarrow t$]

G is a connected directed graph, i.e., $|E| \geq |V| - 1$

Defn: Flow in a Graph: A flow is a real-valued function $f: V \times V \to \mathbb{R}$ that satisfies the following two (02) properties :-

1. Capacity Constraint: $\forall u, v \in V$, $0 \leq f(u,v) \leq c(u,v)$

2. Flow Conservation :- $\forall u \in V \setminus \{s,t\}$,

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

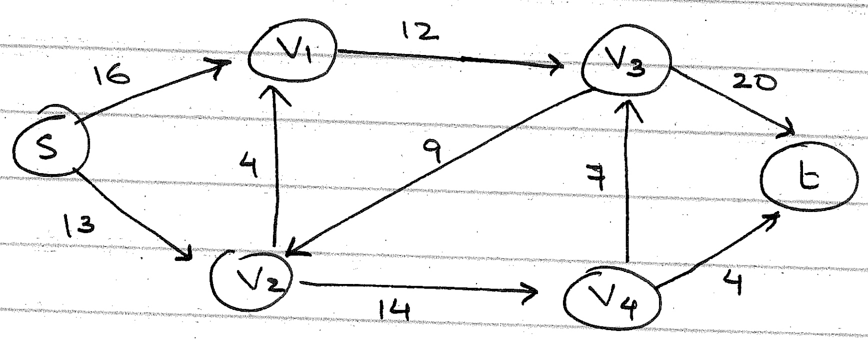
[Kirchoff's Law of Flow of Current]

When

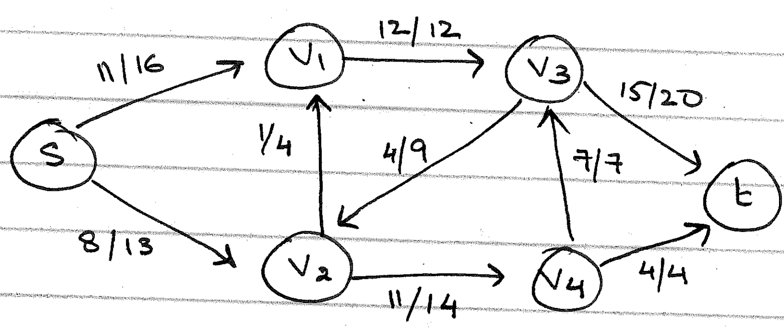
$(u,v) \notin E$, $c(u,v) = 0$ and hence, $f(u,v) = 0$.

Signature ______ UNIQUE No. ______

The following is a flow graph $G = (V, E, s, t)$ [with special nodes 's' and 't'] with capacity for each $(u,v) \in E$, ie.,

$G=$ 

The following is an example flow on G, with each edge labelled as $f(u,v) / c(u,v)$.



$\underbrace{|f| = 19}_{\text{Net flow on G.}}$

Note that for every vertex $v \in V \setminus \{s,t\}$ the flow is conserved, ie., flow-in = flow-out.

The total flow out of the source node minus the total flow coming into the source is

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

$\because$ No edge is coming into a source node

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

Where $|f|$ is the net flow on the graph.

Date

Maximum Flow is a problem to find the maximum possible value of $|f|$ on a graph.
[an optimization problem]

×―――――――――――――――――――――――――――――×

The Ford - Fulkerson Method

An iterative method that iteratively increases the value of flow $|f|$ starting with $f(u,v)=0, \forall u,v \in V$
$\{$ie, $|f|=0\}$

We introduce two concepts:-

I) An augmented path: A path (called an augmenting path) in G is found in the Residual Network G_f (to increase the flow).

II) A residual graph (network) G_f (we'll introduce it shortly!)

Scheme:- Once we know the edges of an augmenting path in the residual graph G_f, we can specify edges in G, for which we can alter flows to increase the total flow on the graph.

Our goal is to maximize $|f|$, the total flow, which may require us to increase or decrease a flow on a particular edge, at each successive iteration.

(Next page...)

Outline of the Ford - Fulkerson Method

1) Initialize the total flow $|f|$ to 0.
2) While there exists an augmenting path p in the residual graph G_f

   a. Augment flow $|f|$ along p.
3) Return $|f|$.

$\underbrace{\text{Return } |f|}_{\text{maximum flow.}}$

So, what's the residual graph? But, first define the augmenting path!

Given a flow network $G=(V,E,s,t)$ and a flow $|f|$, an augmenting path p is a simple path (use BFS or DFS) from s to t

$$s \overset{p}{\leadsto} t \text{ in the residual graph } G_f.$$

Residual Graph:- How much can be sent back from a directed edge? The flow over an edge is now the net flow (both ways!)

1) Admit all edges in G into G_f that have a residual capacity, i.e., $c_f(u,v) = c(u,v) - f(u,v)$ and $c_f(u,v) > 0$.

2) A residual graph G_f may contain additional edges that are not in G. The Ford-Fulkerson method increases the total flow in G by sometimes decreasing a flow at a particular edge (introduces a reverse edge) with a residual capacity of $f(u,v)$

Notice, an edge in a reversed direction can admit as much as, the flow on (u,v) to cancel out.

$\underbrace{\hspace{6cm}}$

Sending back the flow is equivalent to decreasing flow in the actual graph.

<u>Residual Capacity</u> :- $c_f(u,v)$ where $(u,v) \in E$ in $G = (V, E, s, b)$

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & , \text{ if } (u,v) \in E \\ f(u,v) & , \text{ if } (v,u) \in E \\ 0 & , \text{ otherwise} \end{cases} \Big]^{*}$$

* We assumed earlier on that only one of the two cases apply to each ordered pair.

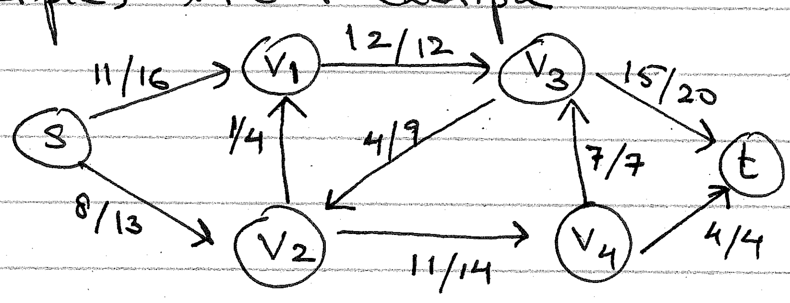
<u>Residual Graph (G_f)</u> :- Given a flow network $G = (V, E, s, b)$, and a flow (total) $|f|$, the residual graph (network) G_f is given as $G_f(V, E_f)$, s.b.

$$E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$$

↳ residual edges with positive residual capacity c_f.

$|E_f| \leq 2|E|$, ie in E_f, edges are either in G or <u>their reversals</u>.

For example, in our example



$|f| = 19$

Now,

Let's check the residual capacity (how much more can be pushed) of edges in E, and the residual capacity of reversed edges (how much can be sent back?)

$c_f(s, v_1) = c(s, v_1) - f(s, v_1) = 16 - 11 = 5\ (>0)$

$[c_f(v_1, s) = 11]$

$c_f(s, v_2) = c(s, v_2) - f(s, v_2) = 13 - 8 = 5\ (>0)$

$[c_f(v_2, s) = 8]$

$$\underbrace{c_f(v_1, v_3) = c(v_1, v_3) - f(v_1, v_3) = 12 - 12 = 0}_{\text{Not admitted in } E_f}$$

but the reversed will be admitted, i.e., $c_f(v_3, v_1) = 12$

only the reversed will be admitted! Why?

$c_f(v_2, v_1) = 4 - 1 = 3\ (>0)$ $[c_f(v_1, v_2) = 1]$

$c_f(v_2, v_4) = 14 - 11 = 3\ (>0)$ $[c_f(v_4, v_2) = 11]$

$c_f(v_3, v_2) = 9 - 4 = 5\ (>0)$ $[c_f(v_2, v_3) = 4]$

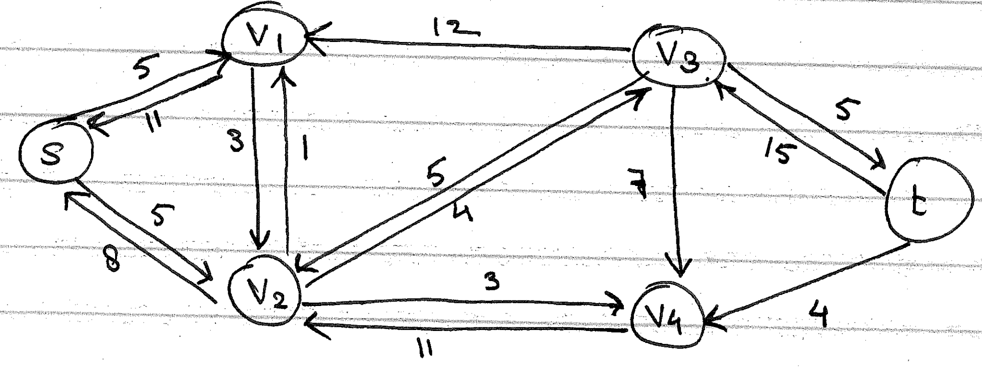
$c_f(v_3, t) = 20 - 15 = 5\ (>0)$ $[c_f(t, v_3) = 15]$

$c_f(v_4, v_3) = 7 - 7 = 0$ {not admitted in E_f} $[c_f(v_3, v_4) = 7]$

$c_f(v_4, t) = 4 - 4 = 0$ {not admitted in E_f} $[c_f(t, v_4) = 4]$

We can now construct the residual graph $G_f = (V, E_f)$ for the given flow network G, with $|f| = 19$.

$G_f =$
(V, E_f)



Each edge in E_f shows the residual capacity c_f.

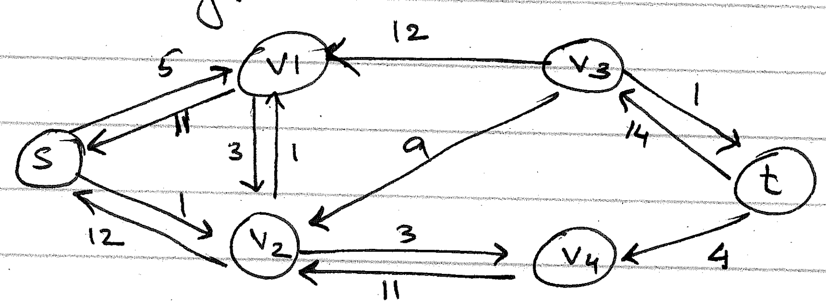
Now, consider an augmenting path p (recall that an augmenting path is a path from s to t in G_f).

eg., p: $s - v_2 - v_3 - t$

If we want to push (add/augment) further flow on this augmenting path, then where is the <u>bottleneck</u>?

$c_f(v_2, v_3) = 4$ [the min]

Adding 4 along p: $s - v_2 - v_3 - t$ results in the residual graph becoming:



With total flow on the network G to be $|f| = 23$.

<u>Is it the maximum flow on G?</u>

The Ford-Fulkerson method continues until there are no augmenting paths left in G_f

$\underbrace{\hspace{10em}}$

Does that guarantee a max-flow? [the optimal solution?]

The residual capacity $C_f(p)$ for an augmenting path p is

$$c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$$

The Min-cut - Max-flow Problem [Theorem]

Defn:-

A (s,t) cut is a partition of a set of vertices V of the given flow graph $G = (V, E, s, t)$ in two sets $\underbrace{S \text{ and } T}_{\text{Disjoint sets}}$,

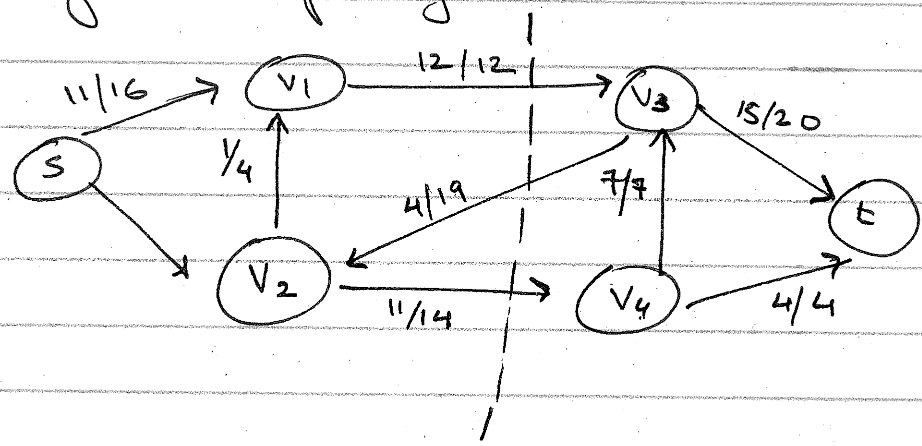
s.t. $(S \cap T = \phi)$ and $s \in S$ and $t \in T$.

If f is a flow, then the net flow $f(S,T)$ across a (s,t) cut is defined as

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

flow in/out across the cut (s,t).

Revisiting our example again:

$G =$ 

The dotted line is a cut (s,t). There are three cross edges; two from S to T and from T to S.

Then, the net flow across the cut is: $12 + 11 - 4 = 19$

$f(S,T) = 19$

More importantly, the capacity of a cut (s,t) is defined as

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

[Notice the asymmetry: we're only concerned with a flow from S to t]

<u>Minimum Cut:</u> is a (s,t) cut whose capacity is minimum over all cuts in a flow network!

So, from our example,

the capacity of the cut (dotted line) was

$$c(S,T) = \underbrace{c(v_1,v_2) + c(v_2,v_4)}_{\text{forward cross-edges}} = 12+14 = 26$$

not optimal!

Not a min-cut!

<u>Lemma</u>: Let f be a flow on a flow network $G=(V,E,s,t)$.
Let (S,T) be any (s,t) cut in G.
Then, the net flow across (S,T) is $|f|$
[Conservation of flow]

<u>Corollary</u>: The value of any flow f in a flow network G is bounded from above by the capacity of any cut in G.

Finding a min-cut (minimum capacity cut) is equivalent to finding max-flow (maximum flow)!

The Min-cut-Max-flow Theorem

If the total flow is $|f|$ in a flow network $G = (V, E, s, t)$, then the following three statements are equivalent:

1) $|f|$ is a maximum flow in G.

2) The residual network G_f contains no augmenting path ($s \not\rightarrow t$)

3) The total flow $|f| = c(u, v)$, $u \in S$, $v \in T$, for some cut (minimum capacity cut) (S, T) of G

The min-cut is the smallest capacity cut (duality property of optimizing problems)

Proof:-

1=>2 Suppose, f is the maximum flow in a flow graph G but an augmenting path 'p' exists in G_f, the residual graph

Then the flow found by augmenting f by some f_p is a flow in G but with a value, which is strictly greater than $|f|$

ie,

$$\underbrace{|f| + |f_p'| > f_{max}}_{\text{a contradiction!}}$$

and

'2' ⇒ '3' Suppose G_f has no augmenting path

i.e. $s \not\xrightarrow{p} t$

Define a S-T cut s.t. $S: \{v \in V, s \rightsquigarrow v \text{ in } G_f\}$
$T: V \setminus S$

Then (S,T) is a cut where $s \in S$ and $t \in T$.

Let u and v be a pair of distinct vertices, s.t., $u \in S$ and $v \in T$.

If $f(u,v) \in E$ then we must have $f(u,v) = c(u,v)$

Otherwise, suppose

$$\underbrace{f(u,v) < c(u,v)}_{\text{a positive residual capacity } c_f \text{ in } G_f}$$

$\because$ $c_f(u,v) > 0$, $(u,v) \in E_f$, which would place v in S.

If $(v,u) \in E$, then $f(v,u) = 0$ because otherwise

if $f(v,u) > 0$
$= c_f(v,u)$

and then,

$(u,v) \in E$, thus again placing v in S

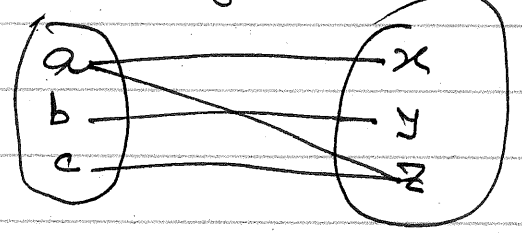
Neither is possible!

(Let's take a detour to Bipartite Matching!)

Slides on Canvas!

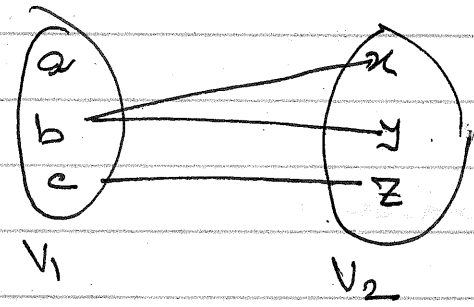
A Matching Problem in a Bipartite Graph

Let's take a very simple example:



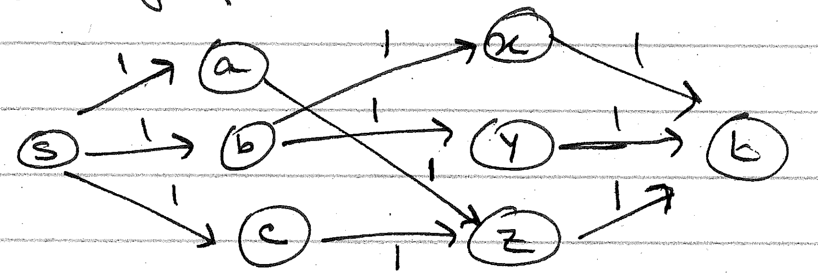
What is the maximum matching in the above graph?

Let's take another simple example:



The maximum possible matching is $|M| = 2 < |V_1|$

Given a flow graph

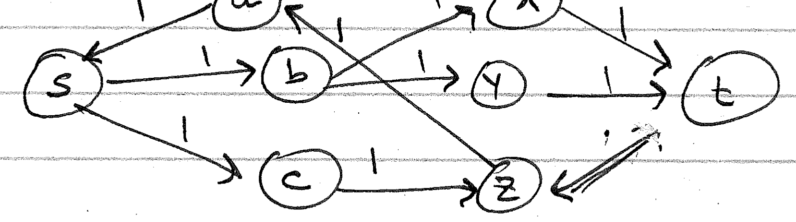
$G =$ 

$|f| = 0$

What is the maximum flow on G?

Consider an augmenting path $s \to a \to z \to t$

Augment flow 1 on this path. The residual graph G_f becomes:



$|f| = 1$