MAXIMUM FLOW

Sample Networks

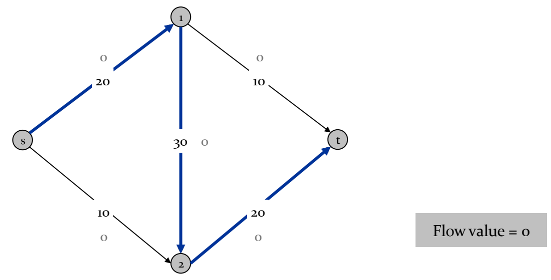
Network	Nodes	Arcs	Flow
communication	telephone exchanges, computers, satellites	cables, fiber optics, microwave relays	voice, video, packets
circuits	gates, registers, processors	wires	current
mechanical	joints	rods, beams, springs	heat, energy
hydraulic	reservoirs, pumping stations, lakes	pipelines	fluid, oil
financial	stocks, companies	transactions	money
transportation	airports, rail yards, street intersections	highways, railbeds, airway routes	freight, vehicles, passengers
chemical	sites	bonds	energy

- MaX-FLOW:
- The greatest rate at which we can ship material from source to the sink/target without violating any capacity constraints.

Towards a Max Flow Algorithm

Greedy algorithm.

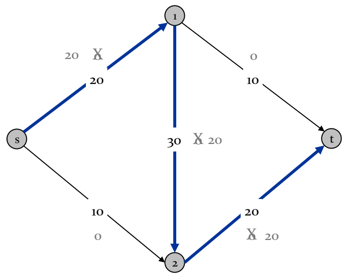
- Start with f(e) = 0 for all edge e ∈ E.
- Find an s-t path P where each edge has f(e) < c(e).
- Augment flow along path P.
- Repeat until you get stuck.



Towards a Max Flow Algorithm

Greedy algorithm.

- Start with f(e) = 0 for all edge e ∈ E.
- Find an s-t path P where each edge has f(e) < c(e).
- Augment flow along path P.
- Repeat until you get stuck.



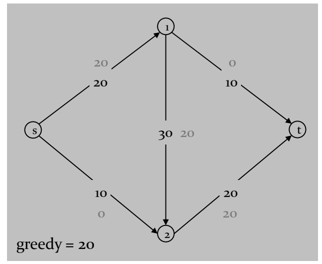
Flow value = 20

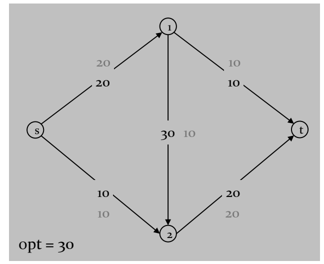
Towards a Max Flow Algorithm

Greedy algorithm.

- Start with f(e) = 0 for all edge e ∈ E.
- Find an s-t path P where each edge has f(e) < c(e).
- Augment flow along path P.
- Repeat until you get stuck.

locally optimality $\nRightarrow$ global optimality





Residual networks

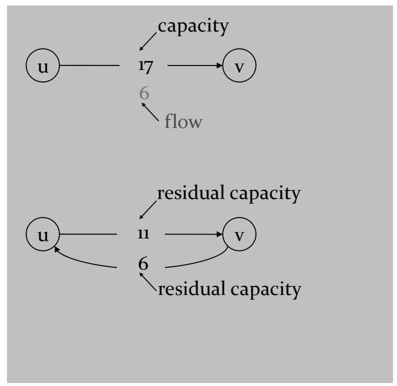
- Given a flow network and a flow, the **residual network** consists of edges that can admit more net flow.
- G=(V,E)
- f: a flow network with source s and sink t,
- The amount of additional net flow from u to v before exceeding the capacity c(u,v) is the residual capacity of (u,v), given by:

$$c_f(u,v)=c(u,v)-f(u,v)$$

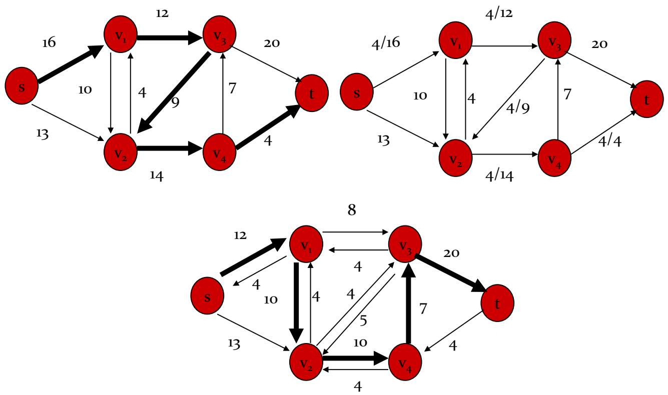
in the other direction:

$$c_f(v, u)=c(v, u)+f(u, v).$$

Residual Graph



Example of residual network



Augmenting paths

- Given a flow network G=(V,E) and a flow f, an augmenting path is a simple path from s to t in the residual network G_f.
- Residual capacity of p : the maximum amount of net flow that we can ship along the edges of an augmenting path p, i.e., $c_f(p)=\min\{c_f(u,v):(u,v) \text{ is on } p\}$.

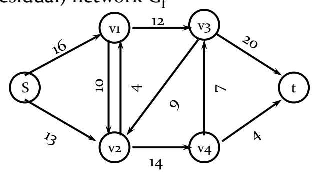
The residual capacity is 1.

The basic Ford Fulkerson algorithm

for *each edge* $(u, v) \in E[G]$
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v)$ is in p}
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + c_f (p)
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm

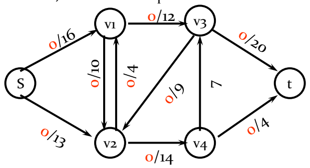
(residual) network G_f



for *each edge* $(u, v) \in E[G]$
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

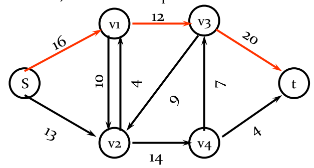
(residual) network G_f



for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

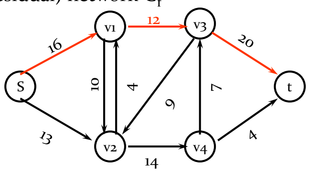
(residual) network G_f



for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f

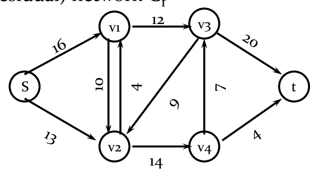


for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

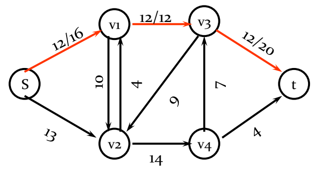
temporary variable:
$c_f(p) = 12$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network G

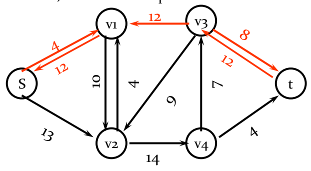


for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

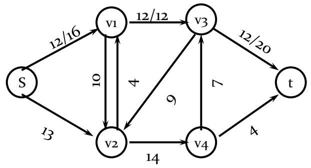
temporary variable:
$c_f(p) = 12$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



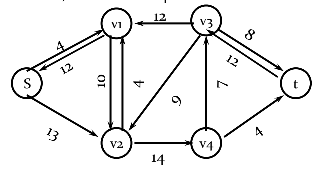
new flow network G



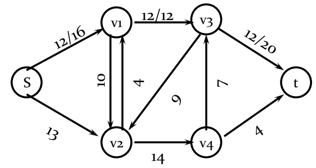
for *each edge (u, v) ∈ E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network G_f*
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



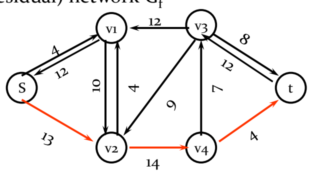
new flow network G



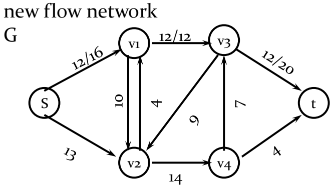
for *each edge (u, v) ∈ E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network G_f*
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



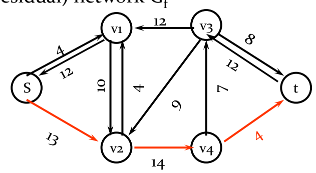
new flow network G



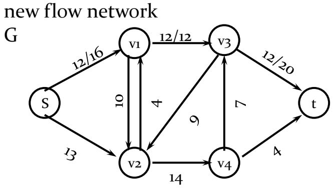
for *each edge (u, v) ∈ E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network G_f*
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network G

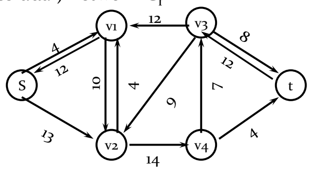


for *each edge (u, v) ∈ E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network G_f*
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

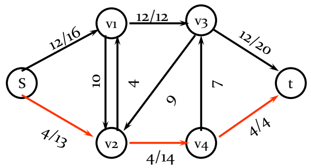
temporary variable:
$c_f(p) = 4$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network
G

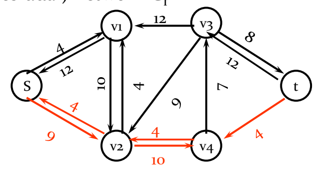



```
for each edge (u, v) ∈ E [G]
   do f [u, v] = 0
      f [v, u] = 0
while there exists a path p from s to t
in the residual network Gf
   do cf(p) = min{cf(u, v) | (u, v) ∈ p}
      for each edge (u, v) in p
          do f [u, v] = f [u, v] + cf(p)
             f [v, u] = - f [u, v]
```

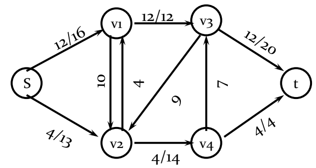
temporary variable:
$c_f(p) = 4$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



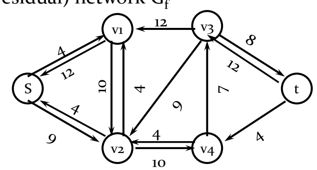
new flow network
G



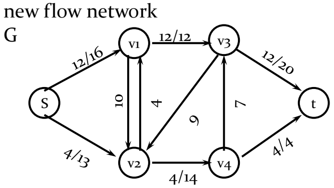

```
for each edge (u, v) ∈ E [G]
   do f [u, v] = 0
      f [v, u] = 0
while there exists a path p from s to t
in the residual network Gf
   do cf(p) = min{cf(u, v) | (u, v) ∈ p}
      for each edge (u, v) in p
          do f [u, v] = f [u, v] + cf(p)
             f [v, u] = - f [u, v]
```

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



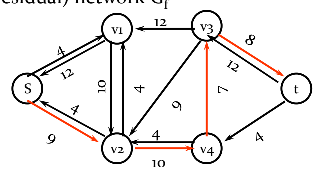
new flow network
G



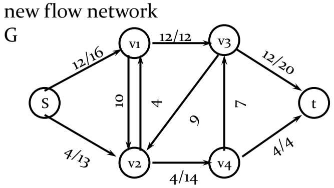

```
for each edge (u, v) ∈ E [G]
   do f [u, v] = 0
      f [v, u] = 0
while there exists a path p from s to t
in the residual network Gf
   do cf(p) = min{cf(u, v) | (u, v) ∈ p}
      for each edge (u, v) in p
          do f [u, v] = f [u, v] + cf(p)
             f [v, u] = - f [u, v]
```

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



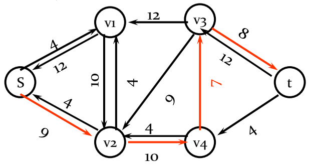
new flow network
G



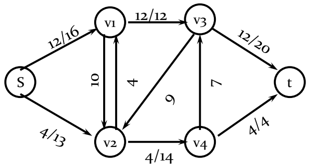

```
for each edge (u, v) ∈ E [G]
   do f [u, v] = 0
      f [v, u] = 0
while there exists a path p from s to t
in the residual network Gf
   do cf(p) = min{cf(u, v) | (u, v) ∈ p}
      for each edge (u, v) in p
          do f [u, v] = f [u, v] + cf(p)
             f [v, u] = - f [u, v]
```

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network G

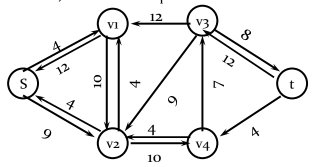


for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

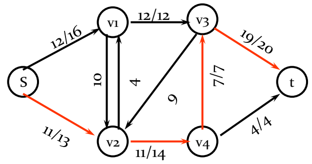
temporary variable:
$c_f(p) = 7$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network G

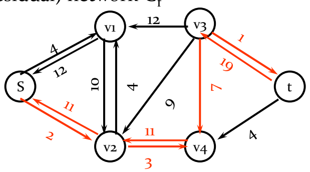


for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

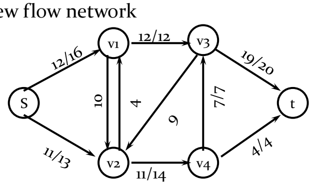
temporary variable:
$c_f(p) = 7$

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



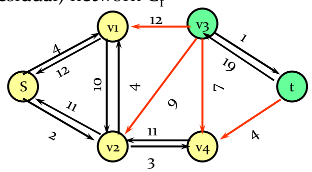
new flow network G



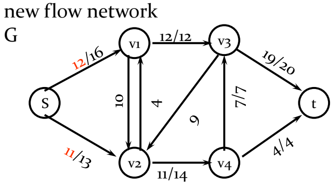
for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

The basic Ford Fulkerson algorithm
example of an execution

(residual) network G_f



new flow network G



for *each edge (u, v)* $\in$ *E [G]*
  do f [u, v] = 0
    f [v, u] = 0
while *there exists a path p from s to t in the residual network* G_f
  do $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    for *each edge (u, v) in p*
      do f [u, v] = f [u, v] + $c_f(p)$
        f [v, u] = - f [u, v]

Finally we have:
$| f | = f (s, V) = 23$

29