4.5 Polynomial-time variations and extensions of Ford and Fulkerson algorithm

A connected graph have at least $\underline{n-1}$ edges

Assume throughout this section that $\underset{\overset{\uparrow}{|V|}}{n} = O(\underset{\overset{\uparrow}{|A|}}{m})$. $m \to n^2$?

If strong connected.

This is not restrictive, because if $m < n-1$, then the graph is disconnected and we can determine the connected component containing the source and focus on that one.

Moreover, let $U := u(A)$ (sum of all capacities)

We will discuss 2 efficient maximum flow algorithms:

(a) The capacity scaling algorithm

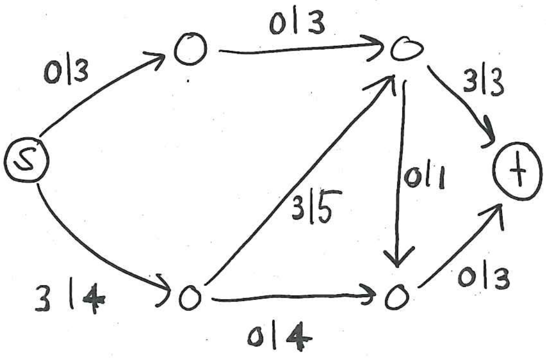
(b) Edmonds-Karp algorithm

4.5.1 Capacity scaling algorithm

Definition 4.30: $G_{f,\Delta}$

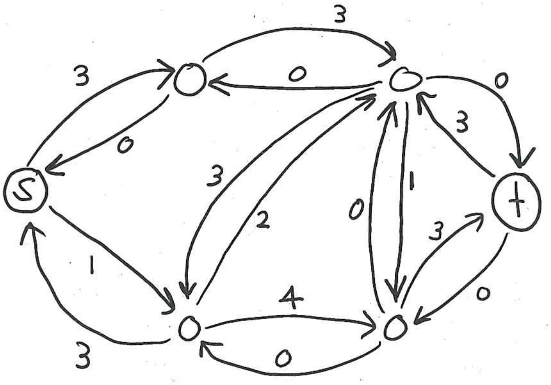
Let f be an s-t flow in the directed graph $G = (V, A)$ with capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and let $\Delta \in \mathbb{R}_{\geq 0}$. We denote by $G_{f,\Delta}$ the subgraph of the residual graph $G_f = (V, B)$ containing only the arcs with residual capacity of at least Δ.

$G = (V, A)$

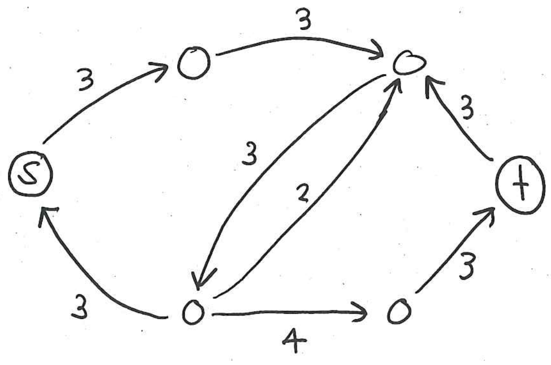


flow | capacity

$G_f = (V, B)$



$G_{f,2}$



Algorithm 6: Capacity scaling algorithm for maximum s-t flows

```
Input  : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum $s$-$t$ flow $f$.
$f(a) = 0 \;\forall a \in A$.                    // We start with the zero flow.
$\Delta = 2^{\lfloor \log_2(U) \rfloor}$.  to make sure $2^n$ power of 2; and most close to $U$
while $\Delta \geq 1$ do                          // These iterations are called phases.
    while $\exists f$-augmenting path $P$ in $G_{f,\Delta}$ do
        Augment $f$ along $P$ and set $f$ to the augmented flow.          $\delta$-phase {
    $\Delta = \frac{\Delta}{2}$.
return $f$                    No augmenting path exits anymore !!!
```

Theorem 4.31

Algorithm 6 returns a maximum s-t flow.

Proof Notice that throughout algorithm. we have $\delta \in \mathbb{Z}$.

$\rightarrow$ In last iteration. we have $\delta = 1$.

However, $G_{f,1} = G_f$. because f is integral throughout algorithm.

$\downarrow$

"Cut off zero-capacity arcs only!".

This iteration finishes when ~~there~~ is no augmenting path in $G_{f,1} = G_f$.

$\xRightarrow{\text{Theorem 4.13}}$ Returend f is maximum s-t flow.

flow

This is poly-time because $(u(0)+1)\cdot(u(a_2)+1)\cdot\ (u(a_n)+1) = \sum u(a) + \dots +$ at least

the input size is $\Theta\left(m + \sum_{a\in A} \log(u(a)+1)\right) = \Theta\left(m + \log \prod_{a \in A}(u(a)+1) \right) = \Omega(m + \log U)$.

& Running time $g = O.($ input size $)$

Not strongly polynominal because the number of operations depends on the numbers provided in the input (U)

Theorem 4.32

Algorithm 6 runs in $O(m^2 \log U)$ time.

Algorithm 6: Capacity scaling algorithm for maximum s-t flows

```
Input  : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum $s$-$t$ flow $f$.
$f(a) = 0\ \forall a \in A$.                          // We start with the zero flow.
$\Delta = 2^{\lfloor \log_2(U) \rfloor}$.
while $\Delta \geq 1$ do                    // These iterations are called phases.
    while $\exists f$-augmenting path $P$ in $G_{f,\Delta}$ do
        Augment $f$ along $P$ and set $f$ to the augmented flow.
    $\Delta = \frac{\Delta}{2}$.
return $f$
```

Proof # phases $= O(\log U)$.

We show that each phase takes $O(m^2)$ time.

Consider current flow f at start of some phase (which is defined by value of Δ)

$\Rightarrow \nexists$ s-t path in $G_{f,2\Delta}$ { This is termination criterion of previous phase (And is clearly true for first phase).

Let $C = \{v \in V : \exists s$-$v$ path in $G_{f,2\Delta}\}$.

→ By previous point. C is an s-t cut.

→ $u_f(a) < 2\Delta,\ \forall a \in \delta^+_{G_f}(s)$. by definition of δ

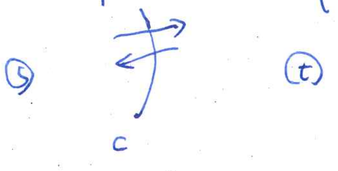
$\Rightarrow u_f(\delta^+_{G_f}(C)) \leq |\delta^+_{G_f}(C)| \cdot 2\Delta \leq 2\Delta \cdot m$

Sum of Maximal in residual Graph is the upper Bound you can improve your flow

Moreover, $u_f(\delta^+_{G_f}(C))$ is upper Bound. on. how much f can be improved in terms of value

$$u_f(\delta^+_{G_f}(C)) = \underbrace{u(\delta^+(C)) - f(\delta^+(C))}_{\text{residual capacities of arcs } \delta^+(C)} + f(\delta^-(C))$$

$\underbrace{- f(\delta^+(C)) + f(\delta^-(C))}_{= -\nu(f)}$ Lemma 4.3



$= \underbrace{u(\delta^+(C))}_{\Downarrow} - \nu(f)$

upperbound on max flow value (weak max-flow min-cut theorem).

Value → Number of Augmentation. Bound

$\Rightarrow$ Augmentations in phase Δ can augment flow by no more than $2\Delta m$.

Each augmentation in phase Δ has augmentation volumn $\geq \Delta$.

$\Rightarrow$ # Augmentations in phase $\Delta = O(m)$.

Each augmentation takes $O(m)$ time via BFS.

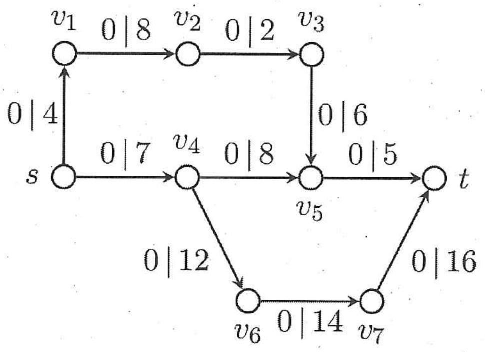
} $\Rightarrow$ time per phase $O(m^2)$ #

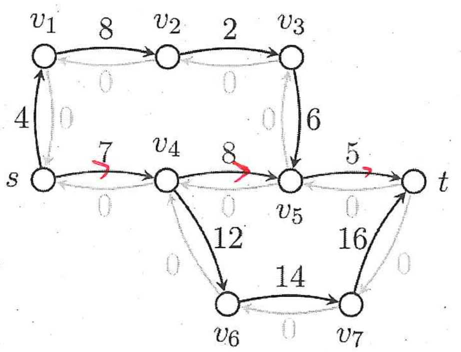
4.5.2 Edmonds-Karp algorithm

Idea : Augment always on shortest paths.

```
Algorithm 7: Edmonds-Karp algorithm
Input  : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum $s$-$t$ flow $f$.
$f(a) = 0 \ \forall a \in A$.
while $\exists f$-augmenting path in $G_f$ do
    Find an $f$-augmenting path $P$ in $G_f$ minimizing $|P|$.
    Augment $f$ along $P$ and set $f$ to augmented flow.
return $f$
```

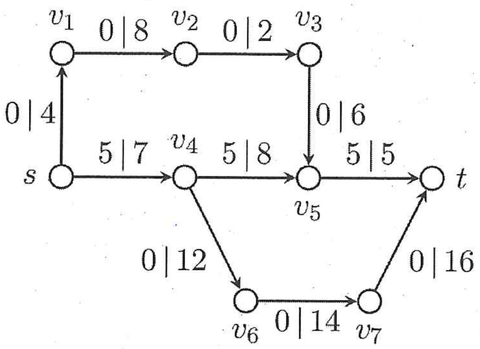
shortest path.
(numbers of vertex)

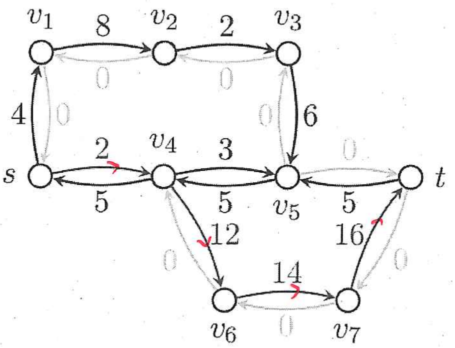




$\gamma = 5$

legth = 3

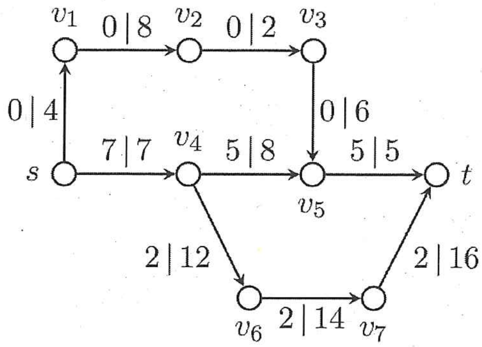


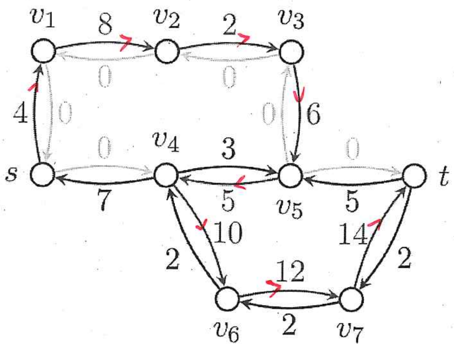


$\gamma = 2$

↓
Becomes larger
Not strictly

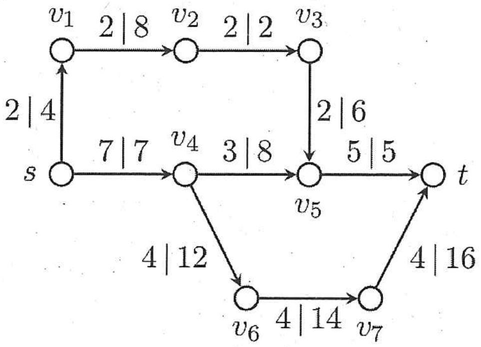
legth = 4

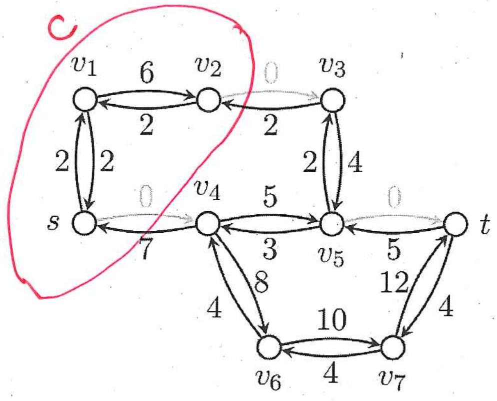




$\gamma = 2$

length = 8.





Key property : Distances from s and distances to t become larger in residual graphs, when only considering arcs with strictly positive residual capacity.

more formally

Lemma 4.33

Let $G = (V, A)$ be a directed graph with arc capacities $u \colon A \to \mathbb{Z}_{\geq 0}$, and let $s, t \in V$ with $s \neq t$. Moreover, let f_1 be an s-t flow in G, and let f_2 be an s-t flow obtained by augmenting f_1 along a shortest augmenting path P in G_{f_1}. Then,

$$d_{f_1}(s,v) \leq d_{f_2}(s,v) \quad \forall v \in V \ , \text{ and}$$
$$d_{f_1}(v,t) \leq d_{f_2}(v,t) \quad \forall v \in V \ ,$$

where $d_f(v,w)$ denotes, for $v, w \in V$ and an s-t flow f, the length (in terms of number of arcs) of a shortest v-w path $\boxed{\text{in } G_f}$ that only uses arcs with strictly positive f-residual capacity.

Proof.

It suffices to show first statement : $d_{f_1}(s,v) \leq d_{f_2}(s,v)$. $\forall v \in V$.

^ Indeed. ~~second~~ second one can be reduced to first one by

- reversing ~~arc~~ directions
- Set the values of capacities, ~~respectively~~, exactly the same.

Notice that G_{f_1}. G_{f_2} are same graphs (V, B) with differ. arc capacities u_{f_1} and u_{f_2}.

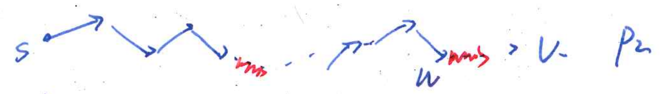
Let $B_i = \{ b \in B : u_{f_i}(b) > 0 \}$. $\forall i \in \{1, 2\}$.

~~def~~ $d_{f_i}(v,w) \leftarrow v$-$w$ distance in (V, B_i)

Assume by sake of contradiction. $\exists v \in V$. st.

$$d_{f_1}(v,w) > d_{f_2}(s,v) \qquad (*)$$

Among all such ~~arc~~ v, choose one where $d_{f_2}(s,v)$ is smallest. Let P a shortest-path ~~path~~ in (V, B_2)



: $P_2 \setminus B_1$. (appears because of residual process).

↳ These arc must be used in opposite direction by P

Claim. (w,v) is not in B_1. Because $d_{f_2}(s,v)$ is smallest among all v fulfilling $(*)$, we have

$$d_{f_1}(s,w) \leq d_{f_2}(s,w) = d_{f_2}(s,v) - 1$$

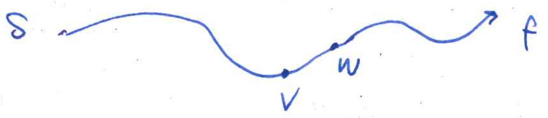
If we have (w,v) in B_1. $\Rightarrow d_{f_1}(s,v) \leq d_{f_1}(s,w) + 1$

$\Rightarrow d_{f_1}(s,v) \leq d_{f_2}(s,v)$. Contradicting $(*)$

Hence, $(w,v) \in B_2 \setminus B_1$. $\Rightarrow$ $(v,w) \in P$.

P is shortest s-t path in (V, B_1) containing (v,w)

$$\Rightarrow \quad d_{f_1}(s,w) = d_{f_1}(s,v)+1.$$



→ see script.

Theorem 4.34

Algorithm 7 runs in $O(nm^2)$ time.

Proof : ~~Alg.~~ lemma 4.33 implies that augmenting paths have non-decreasing ~~leg~~ lengths throughout algorithm.

we can divide Edwards-Karp algo into phases.

phase k: all augmentations on augmenting paths of length k.

phases : $O(n)$

we finish proof by showing that each phase performs $O(m)$ augmentations.

↳ this proves statement because each augmentation takes $O(m)$ time

Consider phase $k \in [n-1]$.

For an s-t flow ~~a~~ f and $v, w \in V$, let . .

Claim : If an arc is used in an augmenting path in phase k, then its reverse arc is not used in any augmentation in phase k.

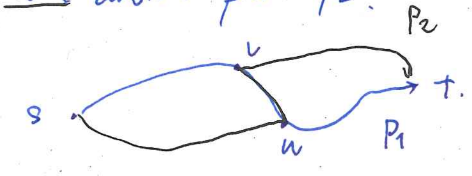
some arcs are gone, as many steps...

proof of claim ,

Assume by sake of Contradiction that $\exists (v,w) \in B$. s.t.

(i) (v,w) is used in a phase k augmenting path P_1 to augment the flow f_1

(ii) (w,v) is later another path P_2.



$$|p_2| = \underline{d_{f_2}(s,w)} + 1 + d_{f_2}(v,t) \geq \underline{d_{f_1}(s,w)} + 1 + d_{f_1}(v,t) = d_{f_1}(s,v) + 2 + d_{f_1}(v,t)$$

$$\uparrow$$

$$= d_{f_1}(s,v) + 1 \qquad\qquad = |p_1| + 2.$$

$\wr$

Because $|p_1| = |p_2|$, as both augmentations happen in phase k.

claim.

claim implies.

In each phase, for every arc $a \in A$, augmentation either never use a or never use a^R.

Hence. once an arc becomes saturated, wether the arc nor its reverse version is used in same phase.

$\Rightarrow$ # of times an arc gets saturated $\leq m$.

Each augmentation saturates at least one arc, by the way we define the augmentation value.

$\Rightarrow$ # of augmentaion in phase $k \leq m$. #.