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

Assume throughout this section that $\underset{|V|}{n} = O(\underset{|A|}{m})$.

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

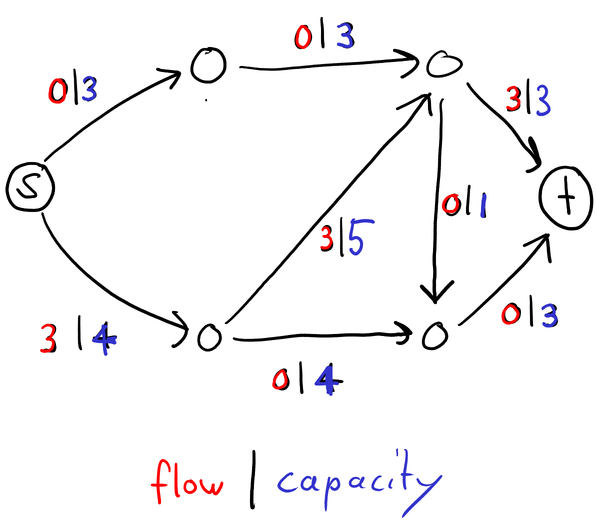
(b) Edmonds-Karp algorithm

4.5.1 Capacity scaling algorithm

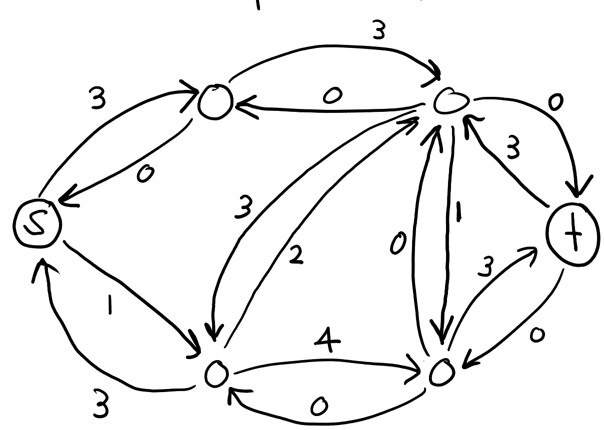
Definition 4.30: $G_{f,\Delta}$

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.

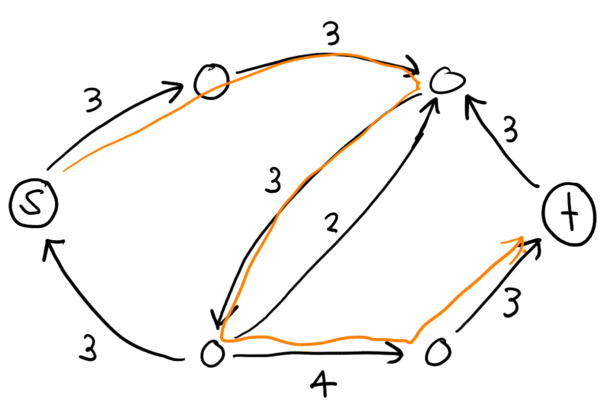
$G = (V, A)$



$G_f = (V, B)$



$G_{f,2}$



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.`
$\Delta = 2^{\lfloor \log_2(U) \rfloor}$.
while $\Delta \geq 1$ **do** `// These iterations are called` *`phases`*`.`
  Δ-phase
  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

Theorem 4.31

Algorithm 6 returns a maximum s-t flow.

Proof

Notice that throughout algorithm, we have $\Delta \in \mathbb{Z}$.

$\to$ In last iteration, we have $\Delta = 1$.

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

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

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

#

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

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).

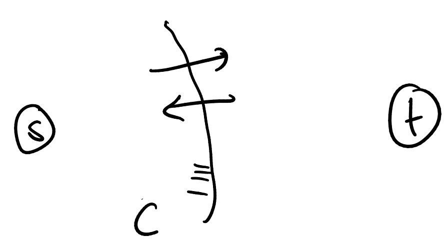
$\nexists$ s-t path in $G_{f,2\Delta}$ $\left\{\begin{array}{l} \text{This is termination criterion of previous phase.} \\ \text{(And is clearly true for first phase.)} \end{array}\right.$

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

→ By previous point, C is an s-t cut.

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

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



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

$$2\Delta m \geq u_f(\delta^+_{G_f}(C)) = \underbrace{u(\delta^+(C)) - f(\delta^+(C))}_{\substack{\text{residual capacities}\\ \text{of arcs } \delta^+(C)}} + f(\delta^-(C)) = \underbrace{u(\delta^+(C))}_{\substack{\text{upper bound on}\\ \text{max flow value.}\\ \text{(weak max-flow min-cut}\\ \text{theorem)}}} - \nu(f)$$

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

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

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

$\Rightarrow$ # augmentations in phase Δ $= O(m)$

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

$\Rightarrow$ time per phase : $O(m^2)$.

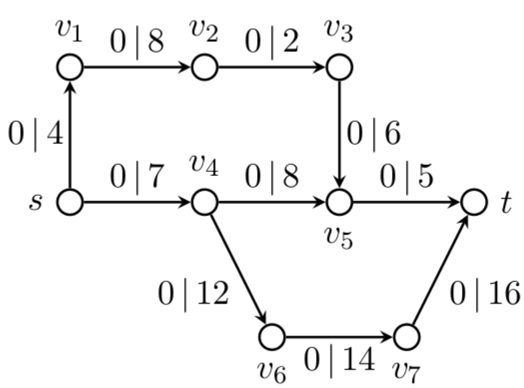
#

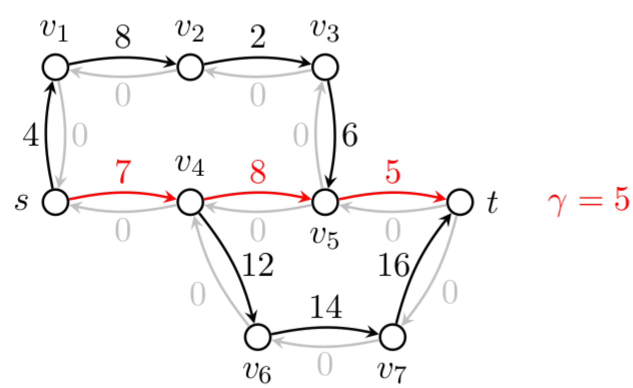
4.5.2 Edmonds-Karp algorithm

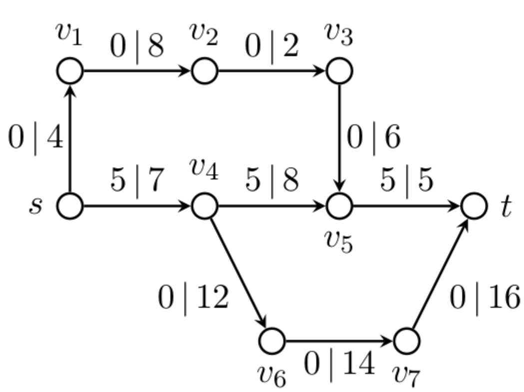
Idea: Augment always on shortest paths.

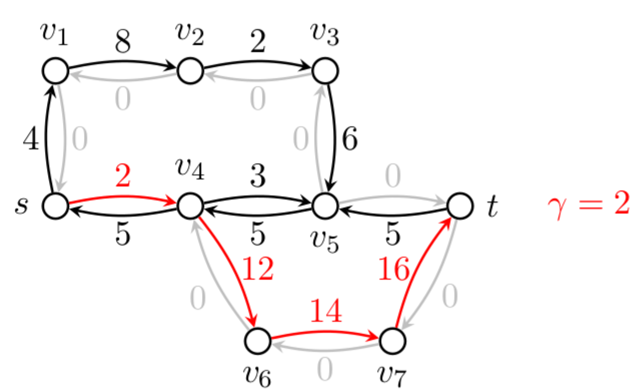
Algorithm 7: Edmonds-Karp algorithm

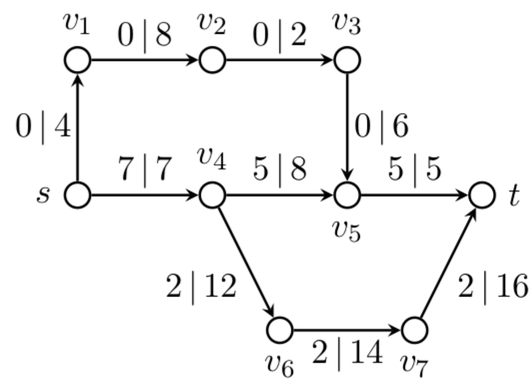
Input : Directed graph $G = (V, A)$ with arc capacities $u\colon A \to \mathbb{Z}_{\geq 0}$ and $s, t \in V$, $s \neq t$.
Output: A maximum s-t flow f.
$f(a) = 0 \;\; \forall a \in A$.
while $\exists f$*-augmenting path in* G_f **do**
  Find an f-augmenting path P in G_f minimizing $|P|$.
  Augment f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

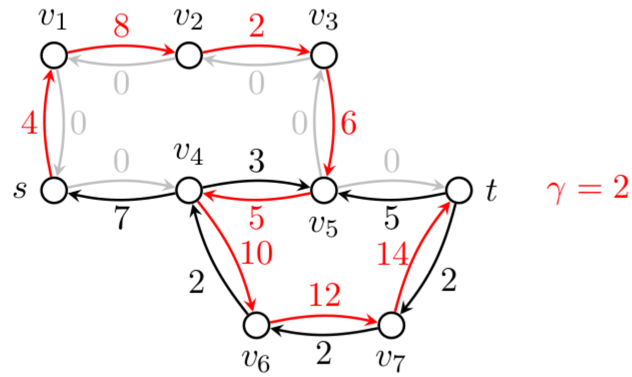


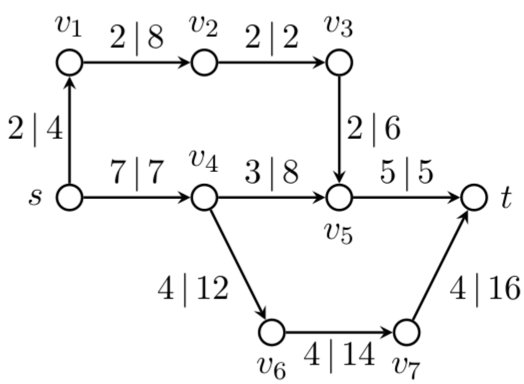


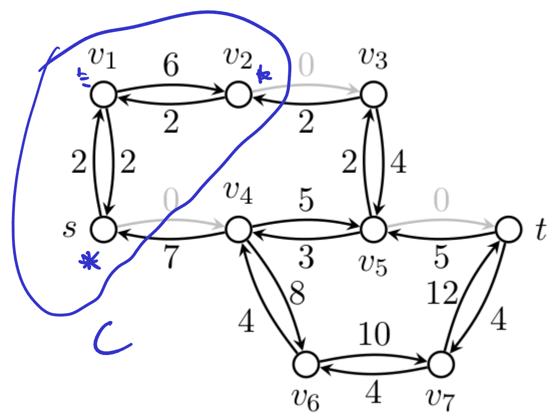





v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
$0 \mid 8$
$0 \mid 2$
$0 \mid 4$
$0 \mid 6$
$7 \mid 7$
$5 \mid 8$
$5 \mid 5$
$2 \mid 12$
$2 \mid 16$
$2 \mid 14$


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
8
2
0
0
4
0
0
6
0
3
0
7
5
5
10
14
2
2
12
2
$\gamma = 2$


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
$2 \mid 8$
$2 \mid 2$
$2 \mid 4$
$2 \mid 6$
$7 \mid 7$
$3 \mid 8$
$5 \mid 5$
$4 \mid 12$
$4 \mid 16$
$4 \mid 14$


v_1
v_2
v_3
v_4
v_5
v_6
v_7
s
t
6
0
2
2
2
2
2
4
0
5
0
7
3
5
8
12
4
4
10
4
C

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 $u: A \to \mathbb{Z}_{\geq 0}$, 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,

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

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 in G_f that only uses arcs with strictly positive f-residual capacity.

Proof

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

Indeed, second one can be reduced to first one by
- reversing arc directions and flows on arcs, and
- exchange roles of s and t.

Notice that G_{f_1} and G_{f_2} are same graphs $(V, \overset{A \cup A^R}{\overset{\|}{B}})$ with different arc capacities u_{f_1} and u_{f_2}, respectively.

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

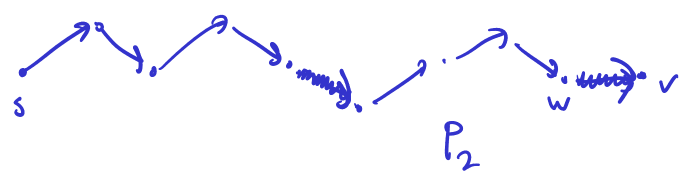
$d_{f_1}(v,w) \leftarrow$ v-w distance in (V, B_1).

Assume by sake of contradiction $\exists\, v \in V$ s.t.

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

$\rightsquigarrow$ Among all such v, we choose one where $d_{f_2}(s,v)$ is smallest.

Let P_2 be a shortest s-v path in (V, B_2). $\rightarrow$ $d_{f_2}(s,v) = |P_2|$.



$\rightsquigarrow$: $P_2 \setminus B_1$

↳ These arcs must have been used in opposite direction by P.

Claim: $(w,v) \notin B_1$

Because $d_{f_2}(s,v)$ is smallest among all v fulfilling (*), we have

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

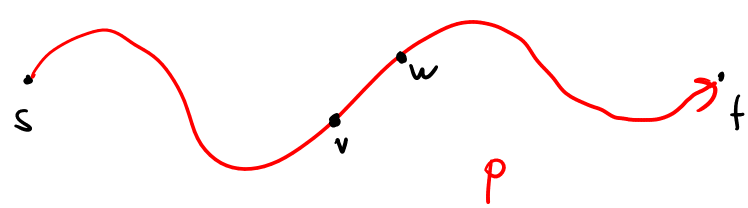
If we had $(w,v) \in B_1 \Rightarrow d_{f_1}(s,v) \leq d_{f_1}(s,w) + 1$

$$\overset{(\square)}{\Longrightarrow} d_{f_1}(s,v) \leq d_{f_2}(s,v) \quad \text{contradicting } (*)$$

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

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

$$\Longrightarrow d_{f_1}(s,w) = d_{f_1}(s,v) + 1$$



$$\left.\begin{array}{l} \Rightarrow d_{f_1}(s,v) < d_{f_1}(s,w) \\ d_{f_2}(s,v) > d_{f_2}(s,w) \end{array}\right\} \Rightarrow$$ Contradiction with choice of v
$\rightsquigarrow$ one could have chosen w instead. $\lightning$

$\rightarrow$ see script for more details. #