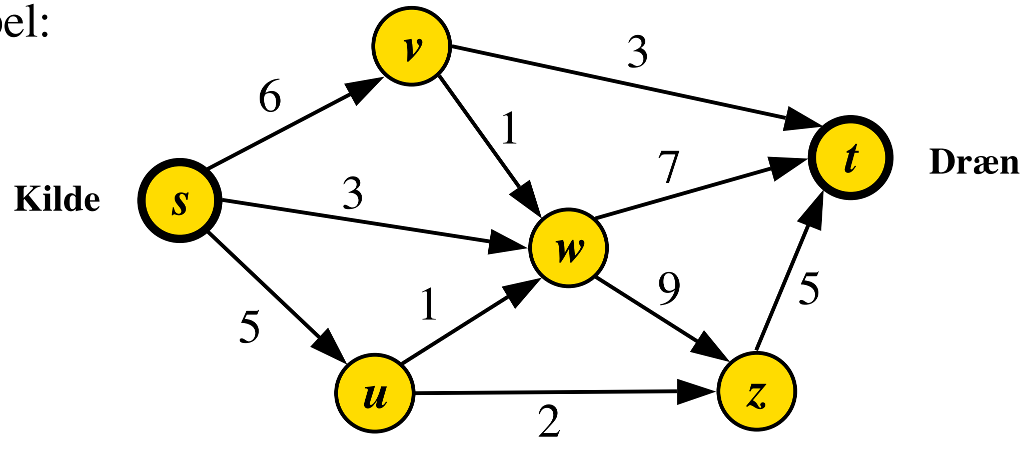
Maksimal strømning

Strømningsnetværk

Et **strømningsnetværk** (eller blot et **netværk**) ***N*** består af

- En vægtet, orienteret graf ***G*** med ikke-negative heltallige kantvægte, hvor vægten af en kant ***e*** kaldes **kapaciteten c(*e*)** af ***e***
- To særlige knuder, ***s*** og ***t*** i ***G***, der kaldes henholdsvis **kilden** og **drænet** (eller **terminalen**), hvor ***s*** ikke har nogen indgående kanter, og ***t*** ikke har nogen udgående kanter

Eksempel:



Strømning

En **strømning** $\boldsymbol{f}$ for et netværk $\boldsymbol{N}$ er en tildeling af en heltallig værdi $\boldsymbol{f(e)}$ til enhver kant $\boldsymbol{e}$, som tilfredsstiller følgende egenskaber:

Kapacitetsreglen: For enhver kant $\boldsymbol{e}$, gælder at $0 \leq \boldsymbol{f(e)} \leq \boldsymbol{c(e)}$

Konserveringsreglen: For enhver knude $\boldsymbol{v} \neq \boldsymbol{s,t}$, gælder at

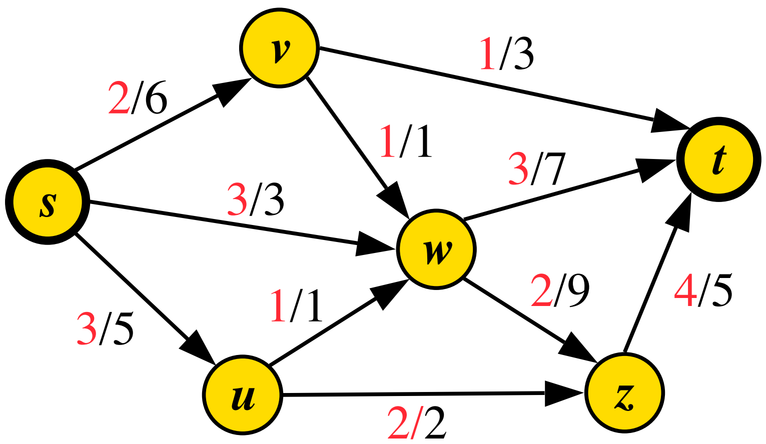
$$\sum_{e \in E^-(v)} f(e) = \sum_{e \in E^+(v)} f(e)$$

hvor $\boldsymbol{E^-(v)}$ og $\boldsymbol{E^+(v)}$ er henholdsvis de indgående og udgående kanter for $\boldsymbol{v}$

Strømningsværdien, betegnet $|f|$, er den samlede strømning fra kilden, hvilket er det samme som den samlede strømning til drænet

Eksempel:

$|f| = 8$



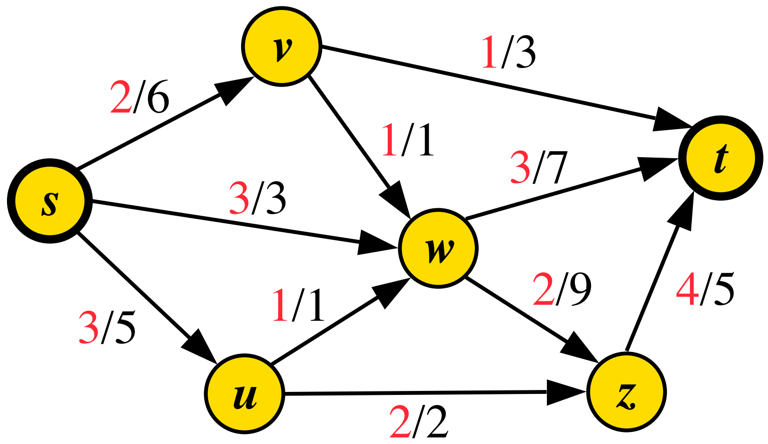
Maksimal strømning

En strømning for et netværk N siges at være **maksimal**, hvis dens værdi er den største af alle strømninger for N

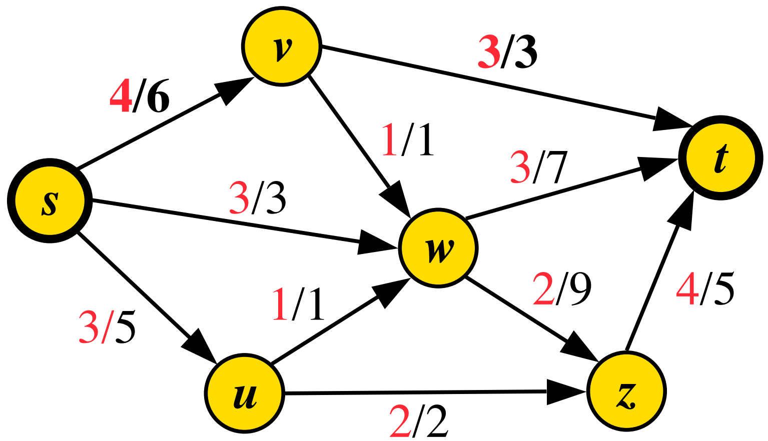
Det maksimale strømningsproblem (maxflow) består i at bestemme en maksimalt strømning for et givet netværk

Anvendelser

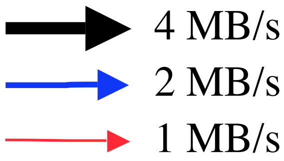
- Hydrauliske systemer
- Elektriske kredsløb
- Trafiksystemer
- Fragt
- Planlægning
- Computernetværk



Strømning med værdi 8 = 2 + 3 + 3 = 1 + 3 + 4



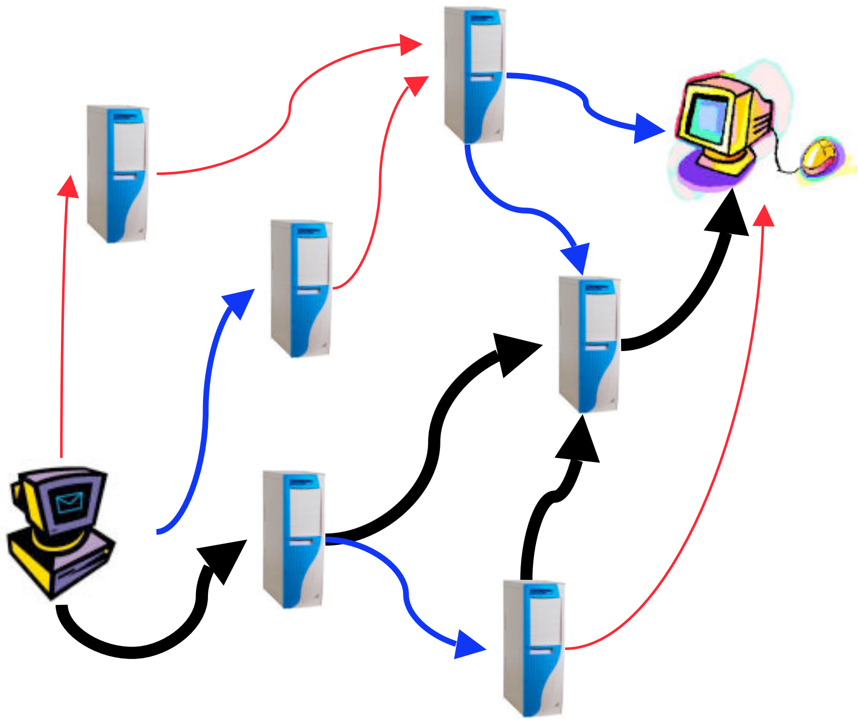
Maksimal strømning med værdi 10 = **4** + 3 + 3 = **3** + 3 + 4



Eksempel på anvendelse

Betragt en del af Internettet, modelleret som en vægtet, orienteret graf, hvor hver *knude* svarer til en computer, hver *kant* svarer til en forbindelse imellem to computere, og *vægten* på en kant er forbindelsens båndbredde (målt i MB/s).

Hvis vi ønsker at sende en meddelelse hurtigst muligt fra en computer til en anden, er det en god ide at opdele meddelelsen i pakker og sende disse igennem netværket, som angivet ved den maksimale strømning.



Snit

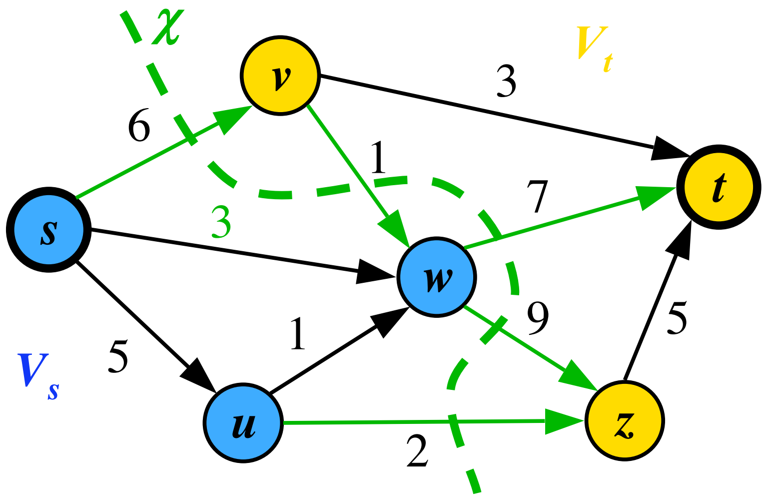
Et **snit** i et netværk $\boldsymbol{N}$ med kilde $\boldsymbol{s}$ og dræn $\boldsymbol{t}$ er en opdeling $\boldsymbol{\chi} = (\boldsymbol{V_s},\boldsymbol{V_t})$ af $\boldsymbol{N}$'s knuder, hvor $\boldsymbol{s} \in \boldsymbol{V_s}$ and $\boldsymbol{t} \in \boldsymbol{V_t}$

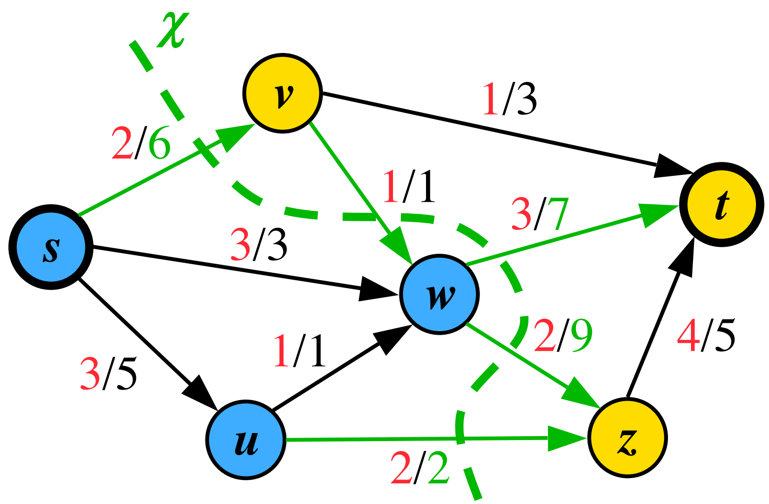
- **Fremadrettet kant** for en snit $\boldsymbol{\chi}$: startknude i $\boldsymbol{V_s}$ og slutknude i $\boldsymbol{V_t}$
- **Bagudrettet kant** for en snit $\boldsymbol{\chi}$: startknude i $\boldsymbol{V_t}$ og slutknude i $\boldsymbol{V_s}$
- **Strømningen** $f(\boldsymbol{\chi})$ for et snit $\boldsymbol{\chi}$: den samlede strømning af de fremadrettede minus den samlede strømning for de bagudrettede kanter
- **Kapaciteten** $\boldsymbol{c}(\boldsymbol{\chi})$ af et snit $\boldsymbol{\chi}$: den samlede kapacitet af de fremadrettede kanter

Eksempel:

$f(\chi) = 2 + 3 + 2 + 2 - 1 = 8$

$c(\chi) = 6 + 7 + 9 + 2 = 24$



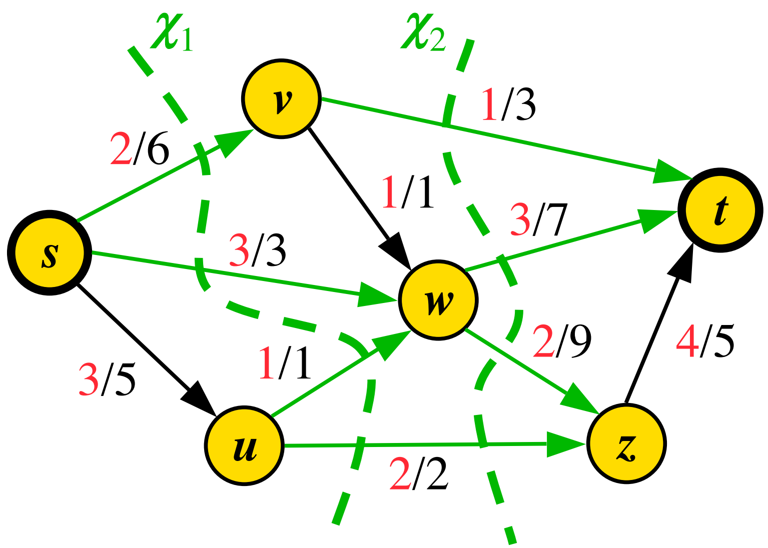


Strømning og snit

Lemma 1:
Strømningen $f(\chi)$ for ethvert snit χ er lig med strømningsværdien $|f|$

Lemma 2:
Strømningen $f(\chi)$ for et snit χ er mindre end eller lig med kapaciteten af snittet, $c(\chi)$

Sætning:
Værdien for enhver strømning er mindre end eller lig med kapaciteten af ethvert snit. D.v.s. for enhver strømning f og ethvert snit χ, gælder at $|f| \leq c(\chi)$

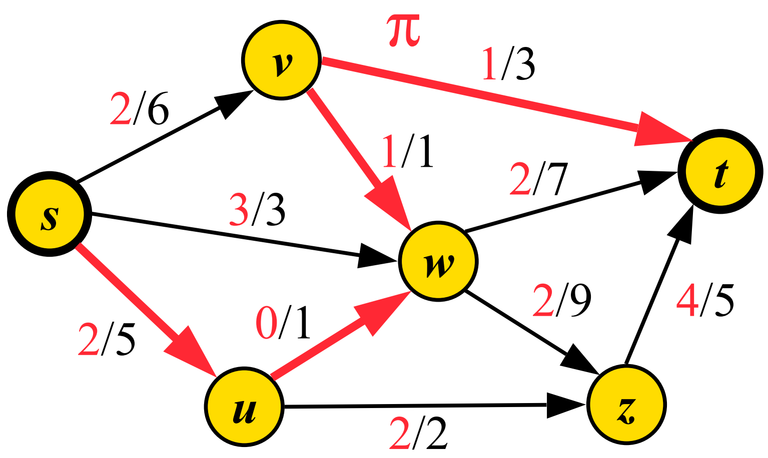


$$c(\chi_1) = 12 = 6 + 3 + 1 + 2$$
$$c(\chi_2) = 21 = 3 + 7 + 9 + 2$$
$$|f| = 8$$

Forøgende vej

Betragt en strømning $\boldsymbol{f}$ for et netværk $\boldsymbol{N}$
Lad $\boldsymbol{e}$ være en kant fra $\boldsymbol{u}$ til $\boldsymbol{v}$:

- **Residualkapaciteten** af $\boldsymbol{e}$ fra $\boldsymbol{u}$ til $\boldsymbol{v}$:
  $\Delta_f(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{c}(\boldsymbol{e}) - \boldsymbol{f}(\boldsymbol{e})$
- **Residualkapaciteten** af $\boldsymbol{e}$ fra $\boldsymbol{v}$ til $\boldsymbol{u}$:
  $\Delta_f(\boldsymbol{v}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{e})$



Lad $\boldsymbol{\pi}$ være en vej fra $\boldsymbol{s}$ til $\boldsymbol{t}$

- Residualkapaciteten $\Delta_f(\boldsymbol{\pi})$ er den mindste af residualkapaciteterne af $\boldsymbol{\pi}$'s kanter i retning fra $\boldsymbol{s}$ til $\boldsymbol{t}$

$\Delta_f(\boldsymbol{s},\boldsymbol{u}) = 3$
$\Delta_f(\boldsymbol{u},\boldsymbol{w}) = 1$
$\Delta_f(\boldsymbol{w},\boldsymbol{v}) = 1$
$\Delta_f(\boldsymbol{v},\boldsymbol{t}) = 2$
$\Delta_f(\boldsymbol{\pi}) = 1$
$|\boldsymbol{f}| = 7$

En vej $\boldsymbol{\pi}$ fra $\boldsymbol{s}$ til $\boldsymbol{t}$ kaldes en **forøgende vej**, hvis $\Delta_f(\boldsymbol{\pi}) > 0$

Forøgelse af strømning

Lemma 3:
Lad $\boldsymbol{\pi}$ være en forøgende vej for strømning $\boldsymbol{f}$ i netværket $\boldsymbol{N}$. Der eksisterer da en strømning $\boldsymbol{f'}$ for $\boldsymbol{N}$ med værdi

$$|\boldsymbol{f'}| = |\boldsymbol{f}| + \Delta_f(\boldsymbol{\pi})$$

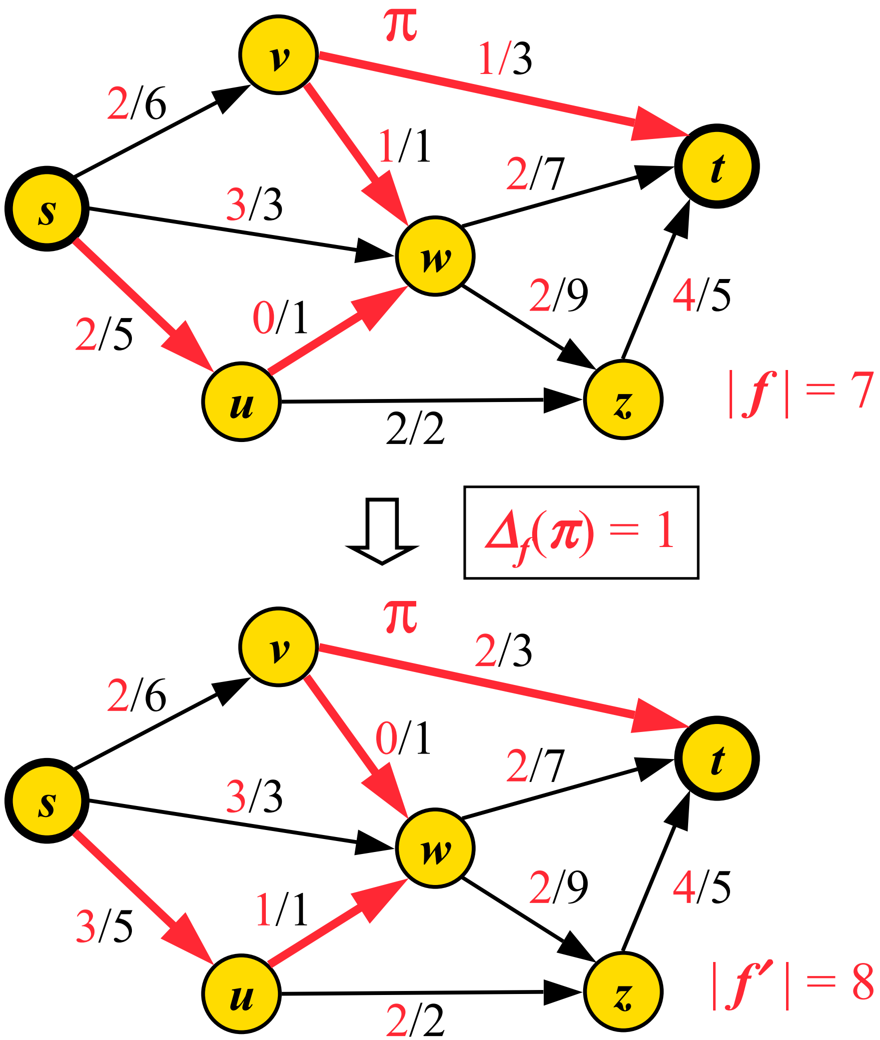
Bevis:
Vi beregner strømningen $\boldsymbol{f'}$ ved at ændre strømningen på $\boldsymbol{\pi}$'s kanter:

Fremadrettet kant:

$$\boldsymbol{f'}(\boldsymbol{e}) = \boldsymbol{f}(\boldsymbol{e}) + \Delta_f(\boldsymbol{\pi})$$

Bagudrettet kant:

$$\boldsymbol{f'}(\boldsymbol{e}) = \boldsymbol{f}(\boldsymbol{e}) - \Delta_f(\boldsymbol{\pi})$$



Ford-Fulkerson's algoritme

Til at begynde med er $\boldsymbol{f(e)} = 0$ for enhver kant $\boldsymbol{e}$

Gentag:

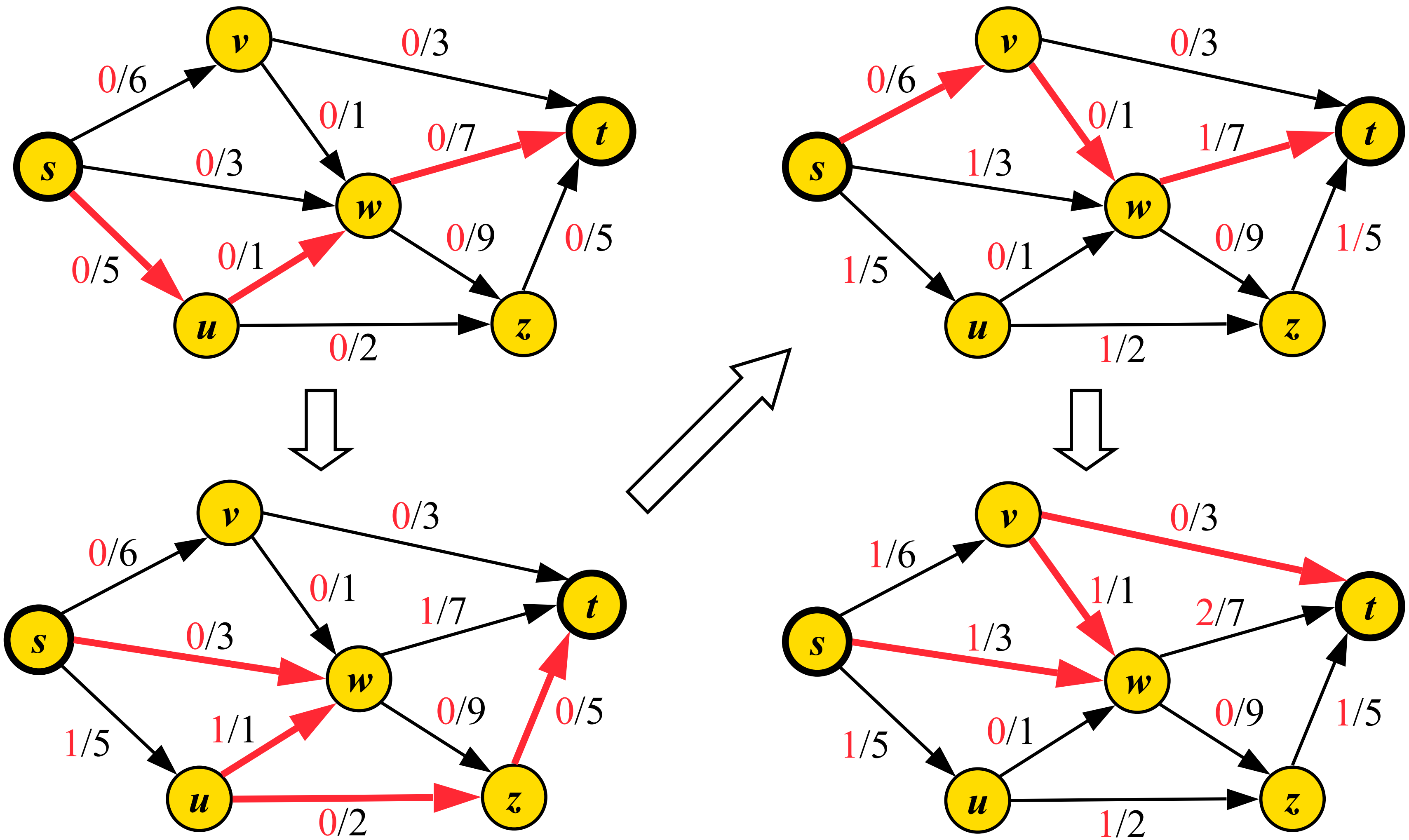
> Søg efter en forøgende vej $\boldsymbol{\pi}$
>
> Forøg strømningen langs $\boldsymbol{\pi}$'s kanter med $\Delta_f(\boldsymbol{\pi})$

En specialisering af DFS (eller BFS) søger efter en forøgende vej.
En kant $(\boldsymbol{u}, \boldsymbol{v})$ traverseres kun, hvis

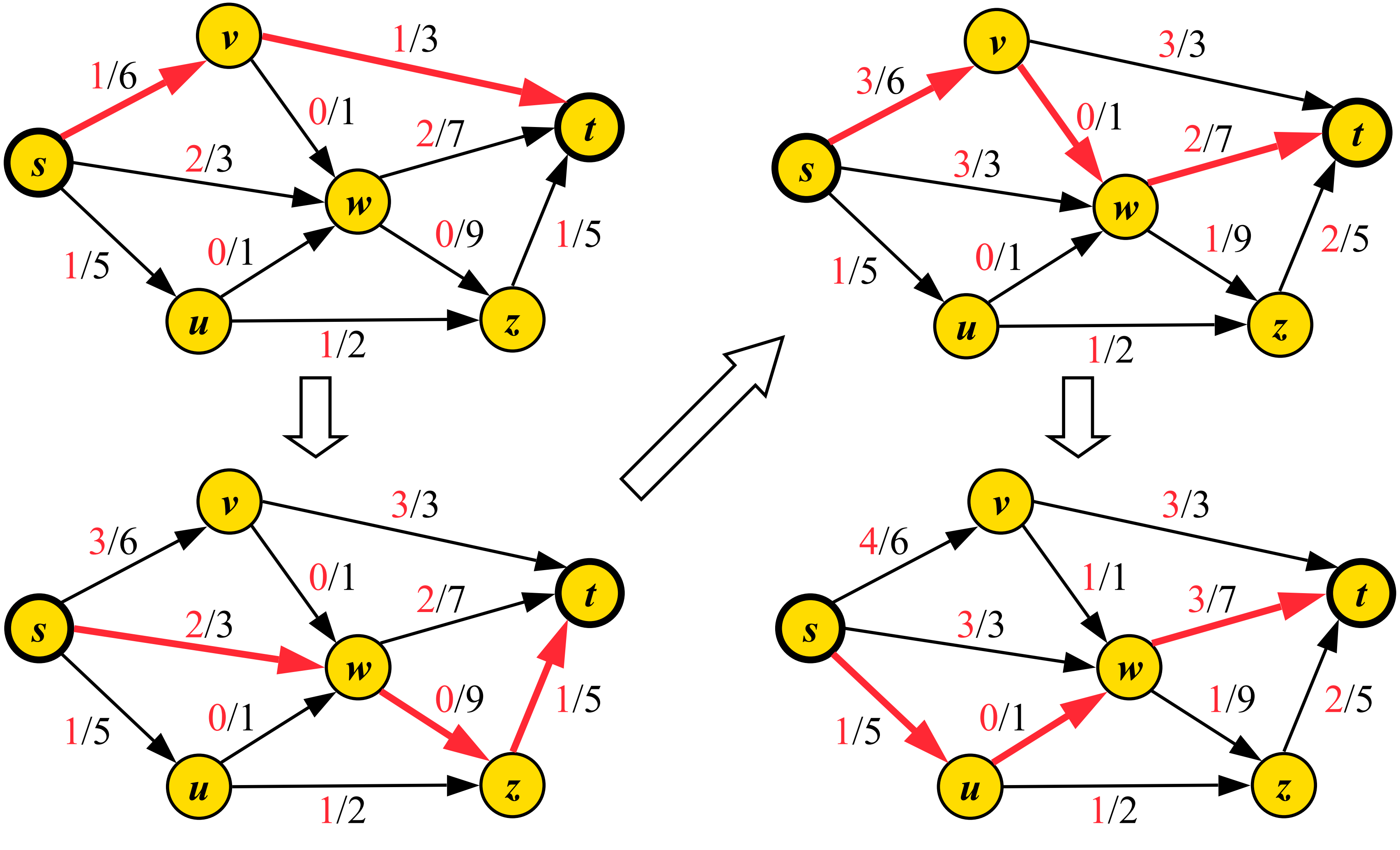
$$\Delta_f(\boldsymbol{u}, \boldsymbol{v}) > 0$$

```
Algorithm FordFulkersonMaxFlow(N)
   for all  e ∈ G.edges()
      setFlow(e, 0)
   while G has an augmenting path π
      { compute residual capacity Δ of π }
      Δ ← ∞
      for all edges e ∈ π
         { compute residual capacity δ of e }
         if e is a forward edge of π
            δ ← getCapacity(e) − getFlow(e)
         else { e is a backward edge }
            δ ← getFlow(e)
         if δ < Δ
            Δ ← δ
      { augment flow along π }
      for all edges e ∈ π
         if e is a forward edge of π
            setFlow(e, getFlow(e) + Δ)
         else { e is a backward edge }
            setFlow(e, getFlow(e) − Δ)
```

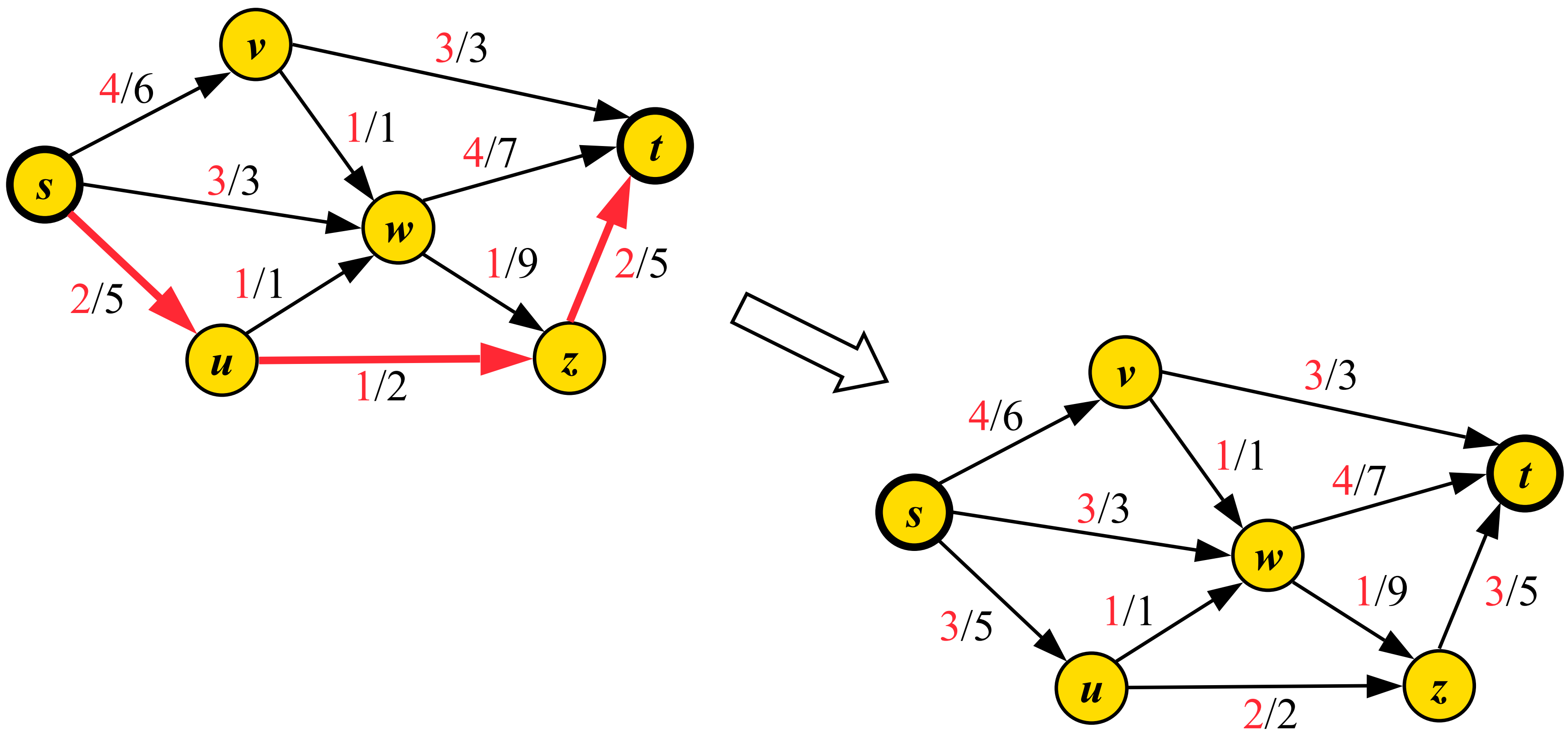
Eksempel


v
0/6
0/3
0/1
0/7
t
s
0/3
w
0/9
0/5
0/5
0/1
u
z
0/2
v
0/6
0/3
0/1
1/7
t
s
1/3
w
0/9
1/5
1/5
0/1
u
z
1/2
v
0/6
0/3
0/1
1/7
t
s
0/3
w
0/9
0/5
1/5
1/1
u
z
0/2
v
1/6
0/3
1/1
2/7
t
s
1/3
w
0/9
1/5
1/5
0/1
u
z
1/2

Eksempel (fortsat)



Eksempel (fortsat)



Maksimal strømning og minimalt snit

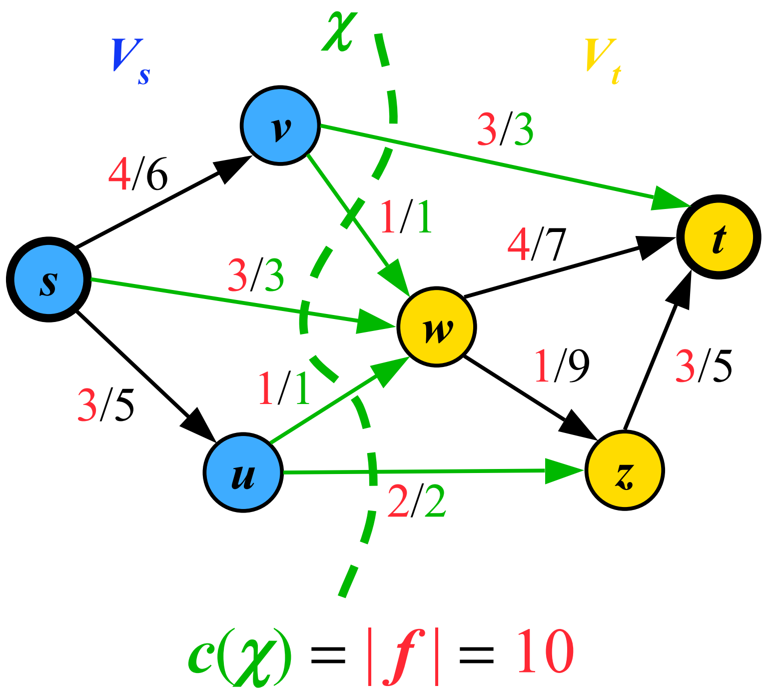
Terminering af Ford-Fulkerson's algoritme:
Der findes ingen forøgende vej fra s til t med hensyn til den aktuelle strømning f

Sætning:
Den maksimale strømning er lig med kapaciteten for et minimalt snit

Definer
V_s mængden af knuder, der kan nås fra s med en forøgende vej
V_t mængden af resterende knuder

Snittet $\chi = (V_s, V_t)$ har kapacitet $c(\chi) = |f|$
Fremadrettet kant: $f(e) = c(e)$
Bagudrettet kant: $f(e) = 0$

Derfor er strømningen f maksimal, og snittet χ har minimal kapacitet



Analyse

I værste tilfælde udfører Ford-Fulkerson's algoritme $|f^*|$ strømningsforøgelser, hvor f^* er en maksimal strømning

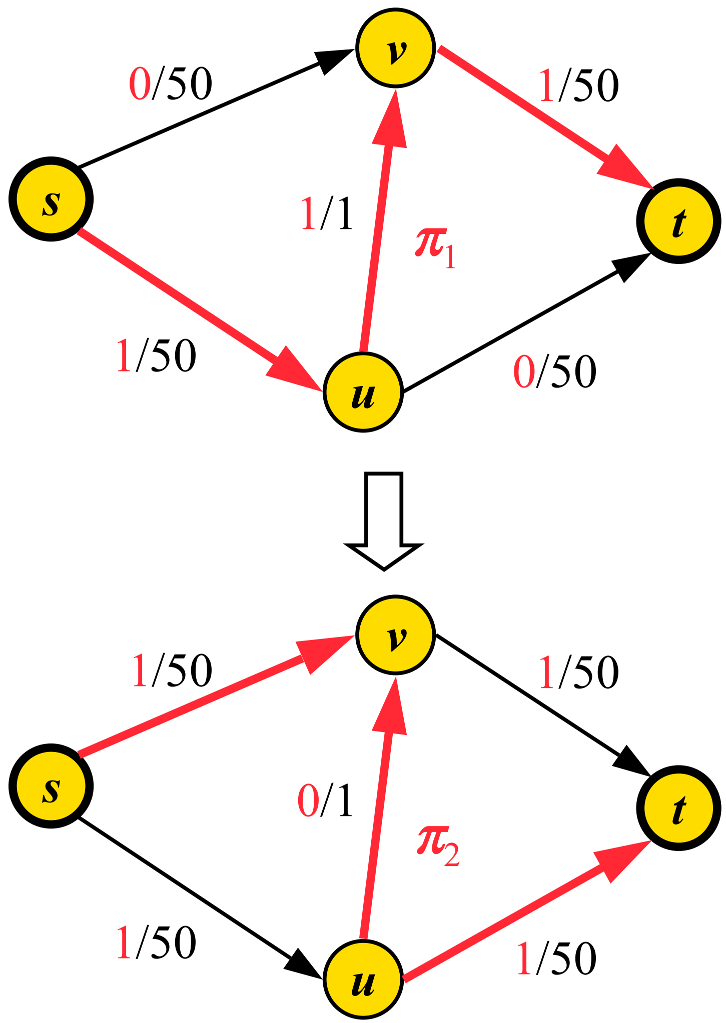
Eksempel:

De forøgende veje, der findes, skifter imellem π_1 og π_2

Algoritmen udfører 100 forøgelser

Bestemmelse af en forøgende vej og efterfølgende forøgelse af strømningen tager $O(n + m)$ tid

Køretiden for Ford-Fulkerson's algoritme er $O(|f^*|(n + m))$



Edmonds-Karp's algoritme

En variation af Ford-Fulkerson's algoritme, som giver hurtigere køretider

Vælg i hver iteration den forøgende vej, der består af færrest kanter

En specialisering af BFS søger efter den forøgende vej.

En kant $(\boldsymbol{u}, \boldsymbol{v})$ traverseres kun, hvis

$$\Delta_f(\boldsymbol{u}, \boldsymbol{v}) > 0$$

Analyse

Lemma 1: Længden af en korteste forøgende vej bliver aldrig formindsket

Lemma 2: Efter højst $\boldsymbol{m}$ forøgelser bliver længden af en korteste forøgende vej øget med mindst 1 kant

Antallet af forøgelser er derfor højst $\boldsymbol{nm}$

Hver forøgelse udføres i $O(\boldsymbol{m})$ tid

Givet et netværk med $\boldsymbol{n}$ knuder og $\boldsymbol{m}$ kanter vil Edmonds-Karp algoritmen bestemme en maksimal strømning i $O(\boldsymbol{nm}^2)$ tid

Algoritmer til bestemmelse af maksimal strømning

Edmonds-Karp's algoritme har køretid $O(\boldsymbol{nm}^2)$
Kan forbedres til $O(\boldsymbol{n}^2\boldsymbol{m})$ ved samtidig opdatering af forøgende veje af samme længde (Dinitz)

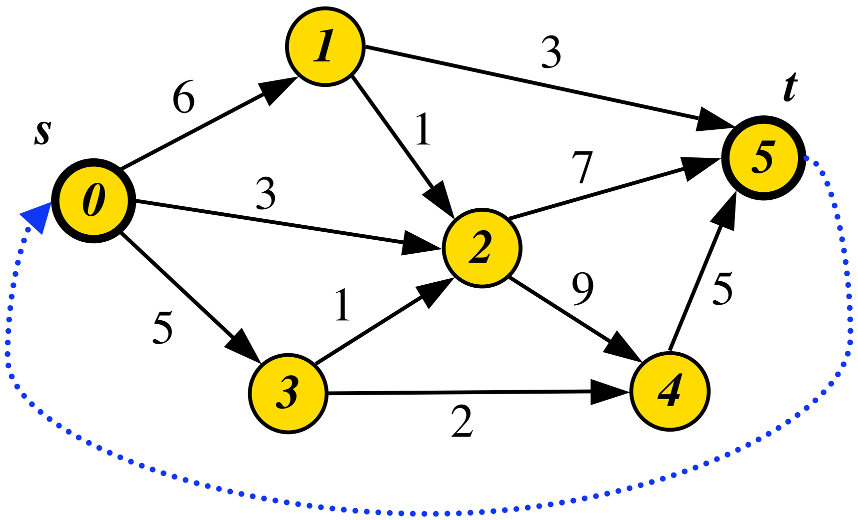
"Push-relabel" (Goldberg) har køretid $O(\boldsymbol{n}^2\boldsymbol{m})$
Kan ved valg af passende valg af rækkefølge forbedres til $O(\boldsymbol{n}^3)$

Den hurtigst kendte algoritme har køretid

$O(\min\{\boldsymbol{n}^{2/3}, \boldsymbol{m}^{1/2}\}\ \boldsymbol{m} \log(1 + \boldsymbol{n}^2/\boldsymbol{m})) \log(\mathit{maksimal\ kapacitet})$

Brug af lineær programmering

Et maxflow-problem kan formuleres som et LP-problem



Maksimer x_{50} under følgende begrænsninger

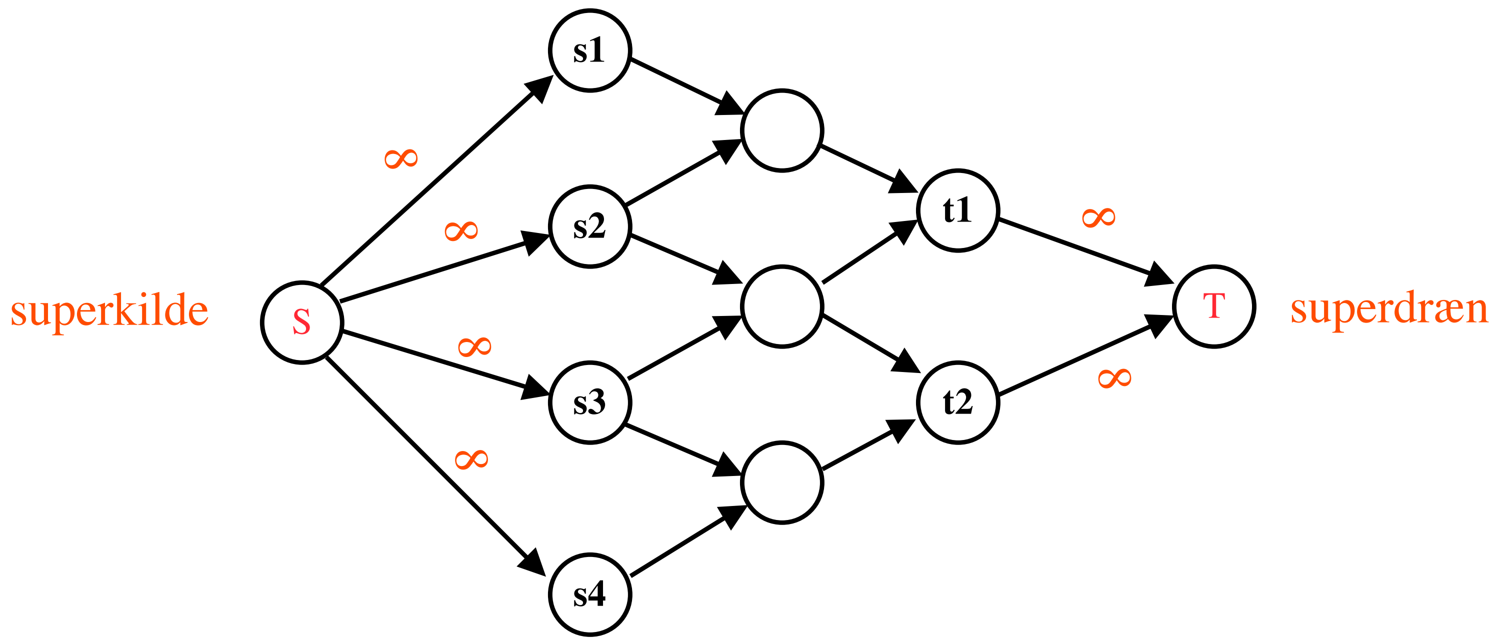
$$\left.\begin{array}{l} x_{01} \le 6,\ x_{02} \le 3,\ x_{03} \le 5 \\ x_{12} \le 1,\ x_{15} \le 3, \\ x_{24} \le 9,\ x_{25} \le 7, \\ x_{32} \le 1,\ x_{34} \le 2, \\ x_{45} \le 5 \end{array}\right\} \text{kapacitets-regel}$$

$$\left.\begin{array}{l} x_{50} = x_{01} + x_{02} + x_{03} \\ x_{01} = x_{12} + x_{15} \\ x_{02} = x_{24} + x_{25} \\ x_{03} = x_{32} + x_{34} \\ x_{12} + x_{02} + x_{32} = x_{25} + x_{24} \\ x_{15} + x_{25} + x_{45} = x_{50} \end{array}\right\} \text{konserverings-regel}$$

Historie

	Opfinder	Metode	Store-O
1951	Dantzig	Simplex	mn^2U
1955	Ford, Fulkerson	Forøgende vej	mnU
1970	Edmonds-Karp	Korteste vej	m^2n
1970	Dinitz	Korteste vej	mn^2
1972	Edmonds-Karp, Dinitz	Kapacitetsskalering	$m^2 \log U$
1973	Dinitz-Gabow	Kapacitetsskalering	$mn \log U$
1974	Karzanov	Preflow-push	n^3
1983	Sleator-Tarjan	Dynamiske træer	$mn \log n$
1986	Goldberg-Tarjan	FIFO preflow-push	$mn \log (n^2 / m)$
...	...	...	...
1997	Goldberg-Rao	Længdefunktion	$m^{3/2} \log (n^2 / m) \log U$ $mn^{2/3} \log (n^2 / m) \log U$

Netværk med flere kilder og dræn

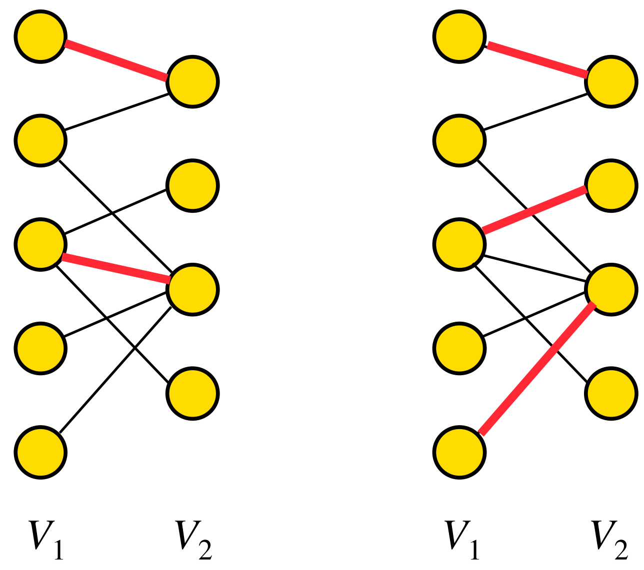


Maksimal parring i en todelt graf

En **todelt graf** er en graf, hvor knuderne kan opdeles i to mængder V_1 og V_2, og hvor hver kant forbinder en knude fra V_1 med en knude fra V_2

En **parring** i en graf er en mængde af kanter, der ikke har endepunkter til fælles

En **maksimal parring** er en parring med et maksimalt antal kanter



En anvendelse

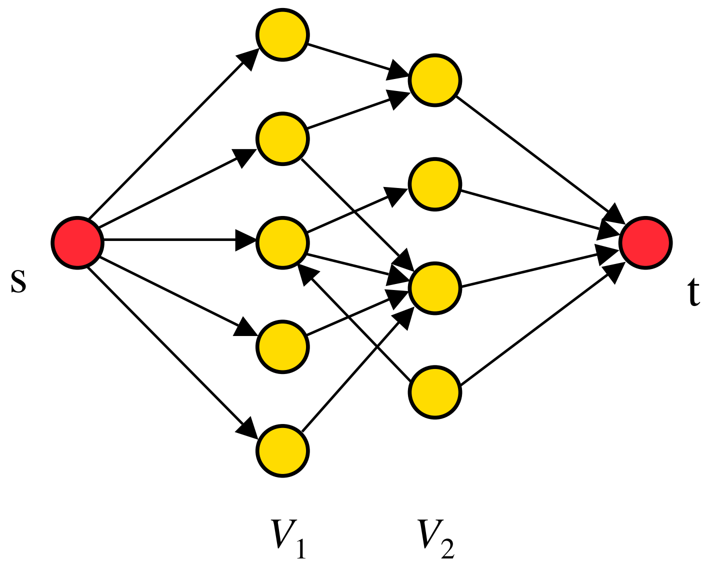
Lad V_1 være en mængde af maskiner og V_2 en mængde af opgaver

En kant betyder, at en maskine kan udføre en given opgave

En maksimal parring giver arbejde til så mange maskiner som muligt i parallel

Løsning som et maxflow-problem

- Tilføj en kilde, s, og et dræn, t
- Orientér kanterne fra V_1 til V_2
- Indsæt en orienteret kant fra s til enhver knude i V_1
- Indsæt en orienteret kant fra enhver knude i V_2 til t
- Tildel alle kanter kapaciteten 1



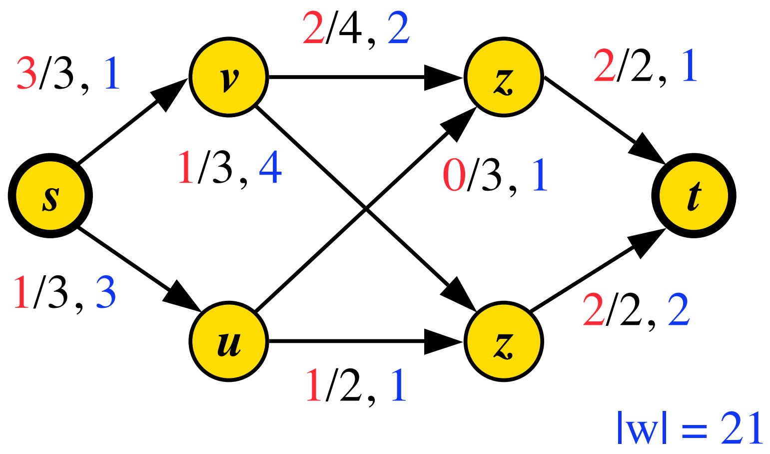
Billigste maksimale strømning

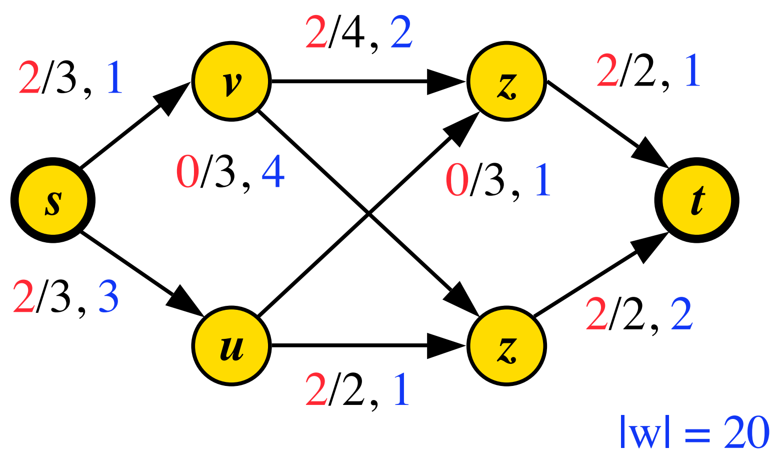
Der er normalt flere løsninger til maxflow-problemet

Vi kan da være interesseret i at finde den af de maksimale strømninger, der er "billigst"

Antag at der til hver kant ***e*** er knyttet en omkostning, ***w***(***e***), for at transportere 1 enhed igennem kanten. Vi kan da definere omkostningen af en strømning ***f*** som

$$w(f) = \sum_{e \in E} w(e) f(e)$$



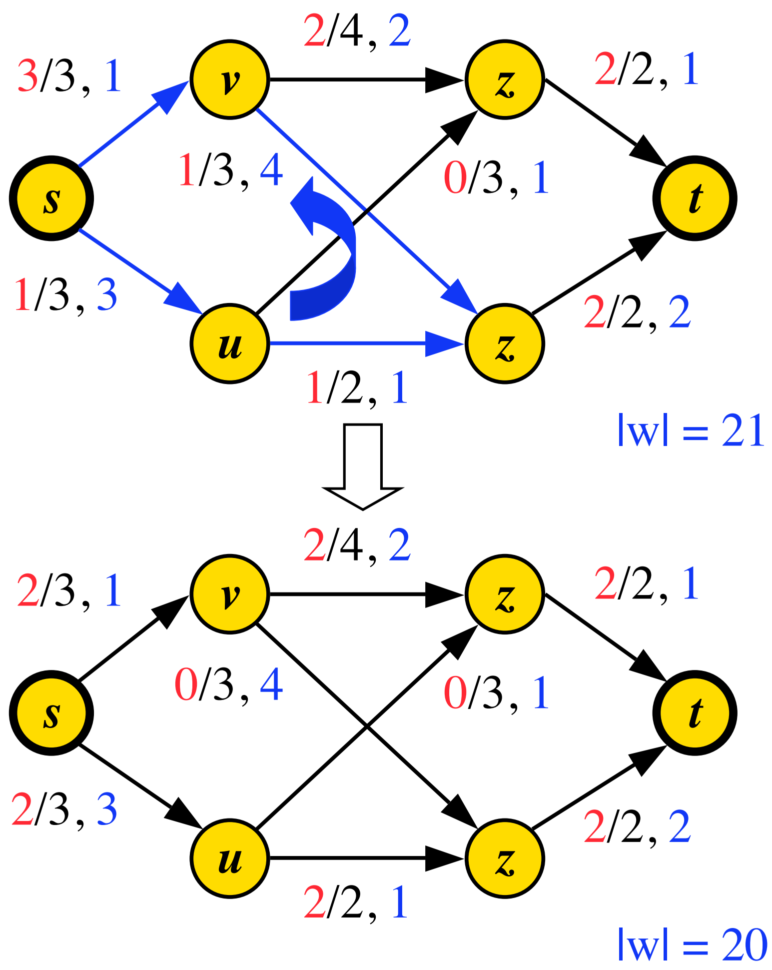


Forøgende cykel

En **forøgende cykel** er en forøgende vej, hvor første og sidste knude er den samme

Hvis vi adderer residualkapaciteten for en forøgende cykel til strømningen for hver af cyklens kanter, ændres strømningsværdien ikke, men det gør omkostningen af strømningen i netværket muligvis

Hvis der findes en forøgende cykel med negativ omkostning, kan vi sænke omkostningen for netværket uden at ændre strømningsværdien



Algoritme til bestemmelse af den billigste maksimale strømning

Sætning: En strømning ***f*** har en minimal omkostning af alle strømninger med værdi |***f***|, hvis og kun hvis der ikke findes en negativ forøgende cykel med hensyn til ***f***

1. Bestem en maksimal strømning
2. Så længe der findes en negativ forøgende cykel:

   Adder cyklens residualkapacitet til strømningen for hver af dens kanter

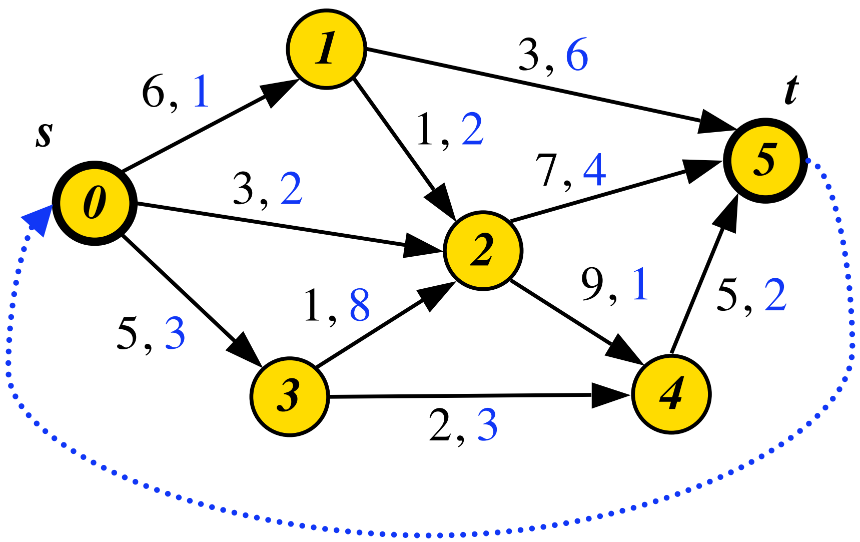
Analyse:

En negativ forøgende cykel kan bestemmes i $O(\boldsymbol{nm})$ tid ved hjælp af Bellman-Fords korteste-vej algoritme

Omkostningen sænkes med mindst 1 for hver sådan cykel

Køretiden er derfor $O(\boldsymbol{w^*nm})$, hvor $\boldsymbol{w^*}$ er den billigste omkostning for en maksimal strømning

Brug af lineær programmering



Minimer

$$c = x_{01} + 2x_{02} + 3x_{03} + 2x_{12} + 6x_{15} + x_{24} + 8x_{32} + 3x_{34} - 100x_{50}$$

under følgende begrænsninger

$$\left.\begin{array}{l} x_{01} \le 6,\ x_{02} \le 3, x_{03} \le 5 \\ x_{12} \le 1,\ x_{15} \le 3, \\ x_{24} \le 9,\ x_{25} \le 7, \\ x_{32} \le 1,\ x_{34} \le 2, \\ x_{45} \le 5 \end{array}\right\} \text{kapacitets-regel}$$

$$\left.\begin{array}{l} x_{50} = x_{01} + x_{02} + x_{03} \\ x_{01} = x_{12} + x_{13} \\ x_{02} = x_{24} + x_{25} \\ x_{03} = x_{32} + x_{34} \\ x_{12} + x_{02} + x_{32} = x_{25} + x_{24} \\ x_{15} + x_{25} + x_{45} = x_{50} \end{array}\right\} \text{konserverings-regel}$$