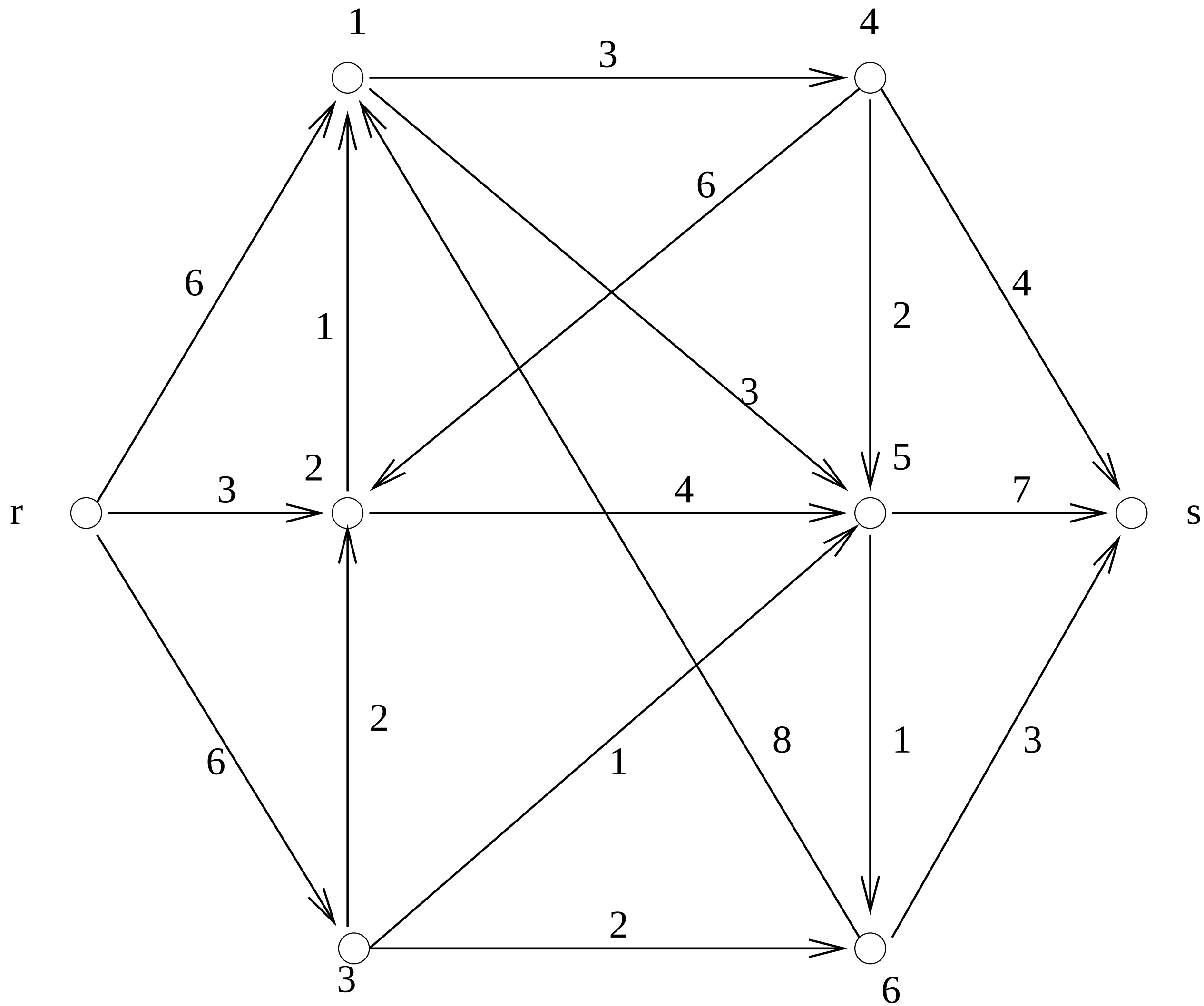
The Max Flow Problem

Jesper Larsen & Jens Clausen

jla,jc@imm.dtu.dk

Informatics and Mathematical Modelling
Technical University of Denmark


1
4
3
6
6
1
4
2
3
2
5
r
3
4
7
s
2
6
1
8
1
3
2
3
6

Max-Flow Terminology

- We consider a **digraph** $G = (V, E)$ which for each e has a **capacity** $u_e \in R_+$.
- Furthermore two “special” vertices r and s are given; these are called resp. the **source** and the **sink**.
- A **flow** in G is a function $x : E \to R_+$ satisfying:

$$\forall v \in V \setminus \{r, s\} \;:\; \sum_{(w,v)\in E} x_{wv} - \sum_{(w,v)\in E} x_{vw} = 0$$

$$\forall (v, w) \in E : 0 \leq x_{vw} \leq u_{vw}$$

Flow

- For a flow we define:

$$f_x(v) = \sum_{(w,v)\in E} x_{wv} - \sum_{(w,v)\in E} x_{vw}$$

- $f_x(v)$ is the **net flow into** v or the **excess for** x **in** v.
- $f_x(s)$ is called the **value** of x.

Math. Programming Model of Max Flow

$$
\begin{array}{lll}
\max & f_x(s) & \\
 & f_x(v) = 0 & v \in V \setminus \{r, s\} \\
 & 0 \leq x_e \leq u_e & e \in E \\
 & x_e \text{ integral}, e \in E &
\end{array}
$$

Max Flow and cuts

- We now consider $R \subset V$. $\delta(R)$ is the set of edges incident from a vertex in R to a vertex in $V \setminus R$. $\delta(R)$ is also called the cut **generated** by R:

$$\delta(R) = \{(v, w) \in E \mid v \in R, w \in V \setminus R\}$$

- R is an **r,s-cut** if $r \in R$ and $s \notin R$.
- The **capacity** of the cut R is defined as:

$$u(\delta(R)) = \sum_{(u,w) \in E, v \in R, w \in V \setminus R} u_{vw}$$

Relationship between flows and cuts

- **Proposition:** For any (r, s)-cut $\delta(R)$ and any (r, s)-flow x we have: $x(\delta(R)) - x(\delta(\bar{R})) = f_x(s)$
- **Corollary:** For any feasible (r, s)-flow x and any (r, s)-cut $\delta(R)$, we have: $f_x(s) \leq u(\delta(R))$.

The Residual graph

- Suppose that x is a flow in G. The **residual graph** shows *how flow excess can be moved* in G given that the flow x is already present.
- The residual graph G_x for G wrt. x is defined by:

$$
\begin{aligned}
V(G_x) &= V \\
E(G_x) &= \{(v,w) \mid (v,w) \in E \wedge x_{vw} < u_{vw}\} \cup \\
&\quad \{(w,v) \mid (v,w) \in E \wedge x_{vw} > 0\}
\end{aligned}
$$

Incrementing and augmenting path

- A path is called x-**incrementing** if for every forward arc e $x_e < u_e$ and for every backward arc $x_e > 0$.
- A x-incrementing path from r to s is called x-**augmenting**.

Algorithm for the Max-Flow Problem

1. Utilize spare capacity
2. dipaths in the residual path
3. flow augmenting path

Initialize $x_{vw} \leftarrow 0$ for all $(v, w) \in E$
repeat
    construct G_x
    find an augmenting path
    if (s is reached)
        augment x as much as possible
until (s is not reachable from r in G_x)

Finding an augmenting path

Input: The network $G = (V, E, u)$ and the current feasible flow x.
Output: An augmenting path from r to s and the capacity for the flow augmentation.

Initialize: all vertices are unlabeled;
$Q := \{r\}, S := \emptyset, p[.] := 0;$
$u_{\max}[v] := +\infty$ for $v \in G$

while $Q \neq \emptyset$ and s is not labeled yet
select a $v \in Q$ and *scan* $\{(v, w) \in E\} \cup \{(w, v) \in E\}$
if $(v, w) \in E$ and $x_{vw} < u_{vw}$ and w unlabeled:
label w; $Q := Q \cup \{w\}$; $p[w] := v$
$u_{\max}[w] := \min\{u_{\max}[v], u_{vw} - x_{vw}\}$
if $(w, v) \in E$ and $x_{wv} > 0$ and w unlabeled:
label w; $Q := Q \cup \{w\}$; $p[w] := v$
$u_{\max}[w] := \min\{u_{\max}[v], x_{wv}\}$
$Q := Q \setminus \{v\}; S := S \cup \{v\}$
if s is labeled then the r-s-aug. path is given by the predecessor index $p[]$.
Otherwise no r–s-augmenting path exists.

Augmenting a feasible flow x

Input: The network $G = (V, E, u)$, the current feasible flow x, an augmenting path P from r to s in G_x and the x-width of P.
Output: An "updated" feasible flow x'.

- for $(v, w) \in P : (v, w) \in E : x'_{vw} = x_{vw} + u_{\max}[s]$
- for $(v, w) \in P : (w, v) \in E : x'_{wv} = x_{wv} - u_{\max}[s]$
- for all other $(v, w) \in E : x'_{vw} = x_{vw}$

Max Flow - Min Cut Theorem

Given a network $G = (V, E, U)$ and a current feasible flow x. The following 3 statements are equivalent:

- x is a maximum flow in G.
- A flow augmenting r-s-path does not exist (an r-s-dipath in G_x).
- An r, s-cut R exists with capacity equal to the value of x, ie. $u(R) = f_x(s)$.

- We call an augmenting path from r to s **shortest** if it has the minimum possible number of arcs.
- The augmenting path algorithm with a breadth-first search solves the maximum flow problem in $O(nm^2)$.
- Breadth-first can easily be established using a queue.