

The Max Flow Problem

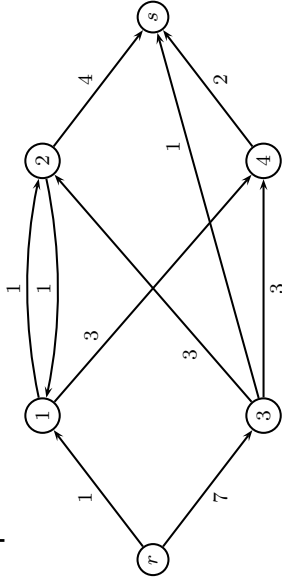
Push-relabel algorithms

Jesper Larsen & Jens Clausen & David Pisinger
jesla, jenc1, pisinger@man.dtu.dk

DTU Management
Technical University of Denmark

The Push-relabel method (preflow push)

- Consider again a directed graph $G = (V, E)$, in which each edge e has a capacity $u_e \in \mathbb{R}_+$.
- Two “special” vertices r and s are given; these are called resp. the **source** and the **sink**.



"Dual approach". Start from better-than optimal solution. Search for feasible solution.

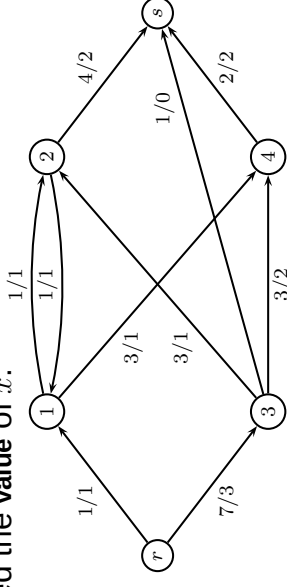
Flow

- A **flow** in G is a function $x : E \rightarrow \mathbb{R}_+$ satisfying:

$$\forall i \in V \setminus \{r, s\} : \sum_{(j,i) \in E} x_{ji} - \sum_{(i,j) \in E} x_{ij} = 0$$

$$\forall (i, j) \in E : 0 \leq x_{ij} \leq u_{ij}$$

- For a flow we define: $f_x(i) = \sum_{(j,i) \in E} x_{ji} - \sum_{(i,j) \in E} x_{ij}$
- $f_x(i)$ is the **net flow into i** or the **excess for x in i** . $f_x(s)$ is called the **value of x** .



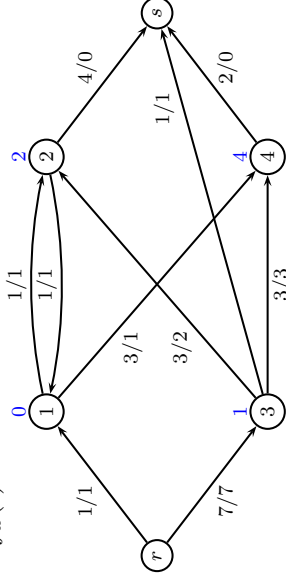
Preflow

- A **preflow** in G is a function $x : E \rightarrow \mathbb{R}_+$ satisfying:

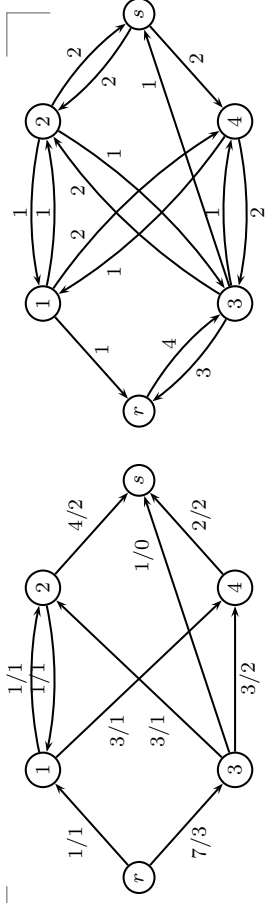
$$\forall i \in V \setminus \{r, s\} : \sum_{(j,i) \in E} x_{ji} - \sum_{(i,j) \in E} x_{ij} \geq 0$$

$$\forall (i, j) \in E : 0 \leq x_{ij} \leq u_{ij}$$

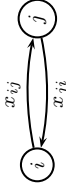
- In other words: for any vertex i except r and s , the flow excess $f_x(i)$ is non-negative - “more runs into i than out of i ”. If $f_x(i) > 0$ the i is called **active**.



The residual graph



- The Residual graph G_x remains defined as previously.
- Let $\tilde{u}_{ij}$ be the “residual capacity” of the arc (i, j) , that is, the total flow we can push from i to j
 $\tilde{u}_{ij} = u_{ij} - x_{ij} + x_{ji}$.



The Max Flow Problem – p. 5

Feasible preflows

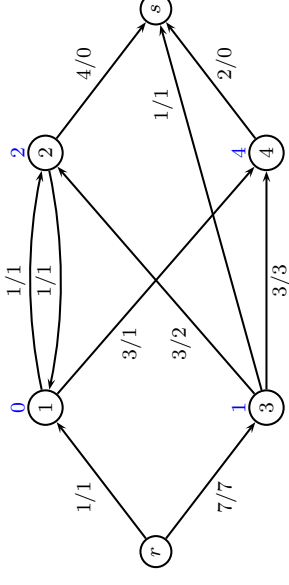
- Looking at a
 - feasible preflow,
 - an edge (i, j) with $\tilde{u}_{ij} > 0$ and $f_x(i) > 0$
 pushing up to $\epsilon = \min\{\tilde{u}_{ij}, f_x(i)\}$ will result in a new feasible preflow.
- Notice: A preflow with no active nodes is a flow.

Preflow: $f_x(i) = \sum x_{ji} - \sum x_{ji} \geq 0 \quad \forall i \in V \setminus \{r, s\}$

$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in E$

The Max Flow Problem – p. 6

A Push Operation



- node 2: push 2 units forward
- node 4: push 2 units forward (4 units not possible)

The Max Flow Problem – p. 7

General idea

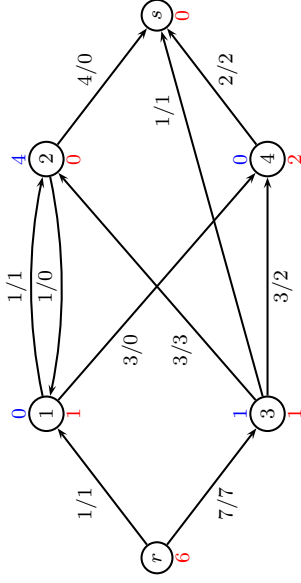
- The general idea is to push as much flow as possible from the source towards the sink.
- However when no more flow can be pushed towards the sink s , and there are still active nodes we push the excess back toward the source r . This restores the balance of flow.
- The challenge is to “control” the push operations.

The Max Flow Problem – p. 8

Labelling

- In order to control the direction of the pushes we introduce estimates on the distances in G_x .
- **Valid labelling** $d[i] \in \mathbb{Z}$ of the vertices in V wrt. a preflow x

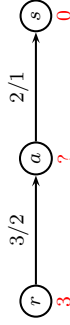
$$d[r] = n, \quad d[s] = 0, \quad \forall (i, j) \in E_x : d[i] \leq d[j] + 1$$
- $d[i]$ is a lower bound on the number of edges in a path from i to s in G_x , if such a path exists.



The Max Flow Problem – p. 9

Preflows vs. valid labellings

- Not all feasible preflows admits a valid labelling.

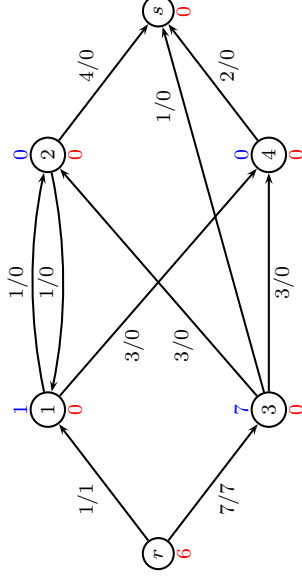


$$d[r] \leq d[a] + 1 \quad d[a] \leq d[s] + 1 \quad \Rightarrow \quad d[r] \leq d[s] + 2$$

The Max Flow Problem – p. 10

Initialize

- It is easy to construct a feasible preflow and a valid labelling for it: (called **initialize** x, d)
 1. set $x_e \leftarrow u_e$ for all outgoing arcs of r
 2. set $x_e \leftarrow 0$ for all remaining arcs.
 3. set $d[r] \leftarrow n$ and $d[i] \leftarrow 0$ otherwise



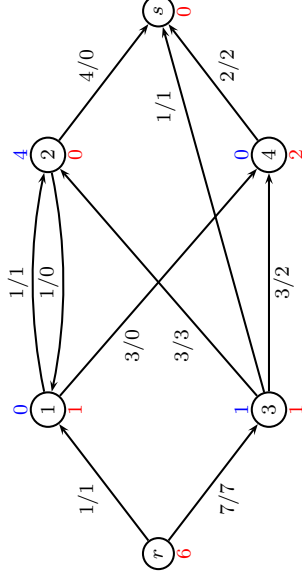
The Max Flow Problem – p. 11

Valid labeling and saturated cut

A valid labelling for a preflow implies a “saturated cut”

Proof: Labels $d[i] \in \mathbb{Z}$ but $n = |V|$ nodes.
Exists $0 < k < n$ such that $d[i] \neq k$ for all $i \in V$.

$$R = \{i \in V : d[i] > k\} \quad \bar{R} = \{i \in V : d[i] < k\}$$



$$d[r] = n, \quad d[s] = 0, \quad \forall (i, j) \in E_x : d[i] \leq d[j] + 1$$

The Max Flow Problem – p. 12

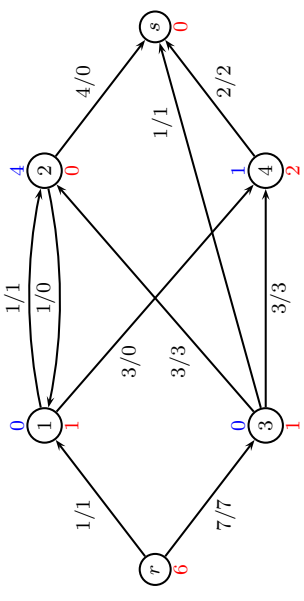
Preflow vs. maximum flow

- **Corollary:** If a feasible flow x has a valid labelling, then x is a maximum flow.
- This gives us a possible termination condition for an algorithm.
- Therefore a push-relabel algorithm maintains a feasible preflow and a valid labelling (and a saturated cut) and terminates when the preflow becomes a flow.
- Note the duality with the augmenting path algorithm, which maintains a feasible flow and terminates when a cut becomes saturated.

The Max Flow Problem – p. 13

Approx. to distances: special cases

- $d[i]$ is a lower bound on the distance from i to s .
 $d_x(i, s) \geq d[i] - d[s] = d[i]$.
- $d[i] - n$ is a lower bound on the distance from i to r .
 $d_x(i, r) \geq d[i] - d[r] = d[i] - n$.
- If $d[i] \geq n$ this means that there is no path from i to s .
- Admissible edges can be used to push flow excess towards s (or, if s is not reachable, towards r).



The Max Flow Problem – p. 15

Approx. to distances

- Let $d_x(i, j)$ denote the number of arcs in a shortest directed path from i to j in G_x .
- With the definition d we can prove that for any feasible preflow x and any valid labelling d for x we have

$$d_x(i, j) \geq d[i] - d[j]$$

- We try to push flow towards nodes j having $d[j] < d[i]$, since such nodes are estimated to be closer to the ultimate destination.

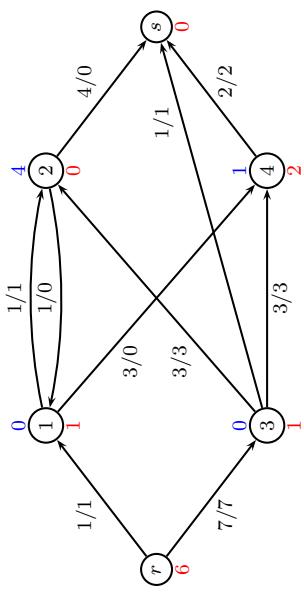


$$d[i] \leq d[j] + d_x(i, j)$$

The Max Flow Problem – p. 14

Admissible edge

- An edge (i, j) is called **admissible** wrt. a preflow x and a valid labelling d , if $(i, j) \in E_x$, i is active and $d[i] = d[j] + 1$.



- Edge $(4,3)$ is admissible

The Max Flow Problem – p. 16

The Push operation

- Consider an **admissible** edge (i, j) .
- Calculate the amount of flow, which can be pushed to j

$$\min\{f_x(i), (x_{ji} + (u_{ij} - x_{ij}))\}$$
- Push this by first reducing x_{ji} as much as possible and then increasing x_{ij} as much as possible.

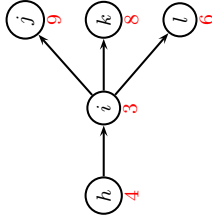
The Preflow-Push Algorithm

- Initialize x, d
- while** x is not a flow
- select an active node i
- while** there exists an admissible arc (i, j)
- push on (i, j)
- if** i is active
- relabel i

Typically we relabel before we push

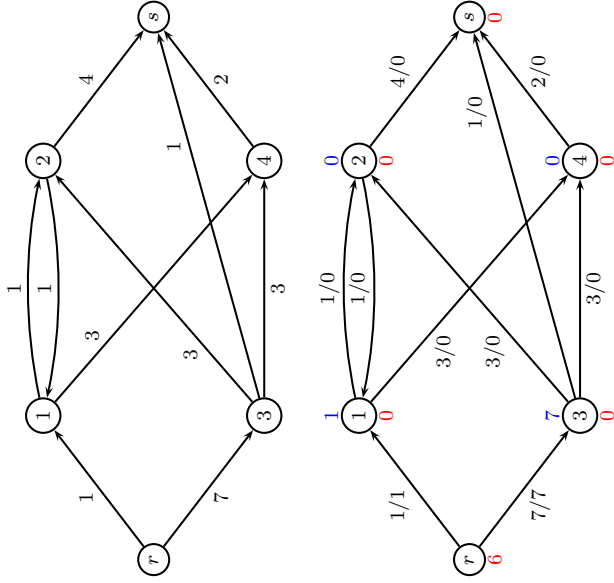
The Relabel operation

- No more admissible edges and a node i is still active?
- Relabel**
- Consider an active vertex i , for which **no** edge $(i, j) \in E_x$ with $d[i] = d[j] + 1$.
- Increase $d[i]$ to $\min\{d[j] + 1 : (i, j) \in E_x\}$ - resulting in a new valid labelling.
- This will not violate the validity of the labelling.

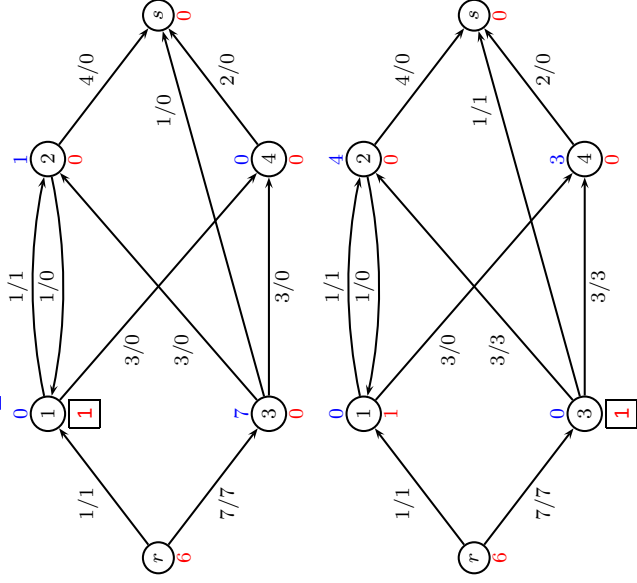


$d[r] = n, \quad d[s] = 0, \quad \forall (i, j) \in E_x : d[i] \leq d[j] + 1$

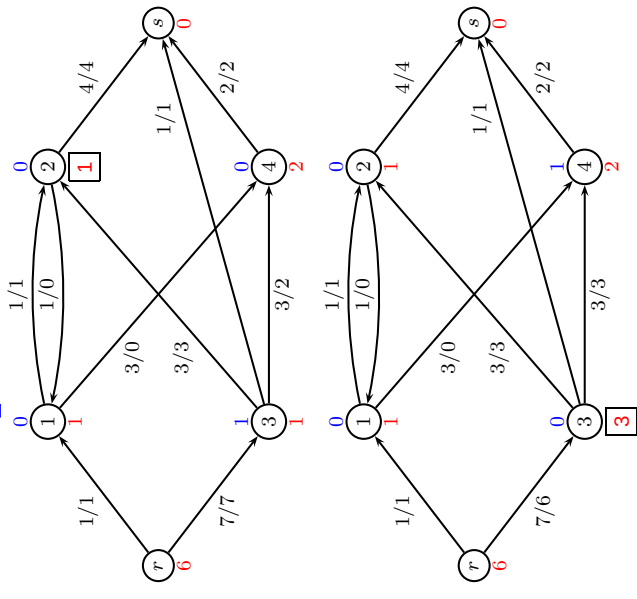
Example



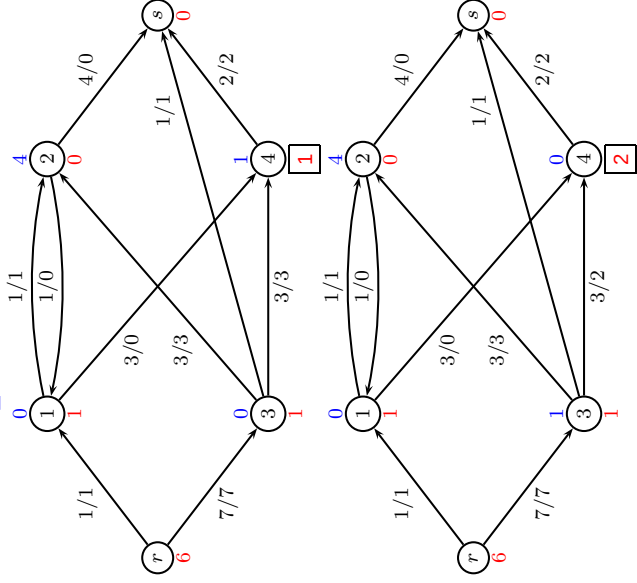
Example, iteration 1, 2



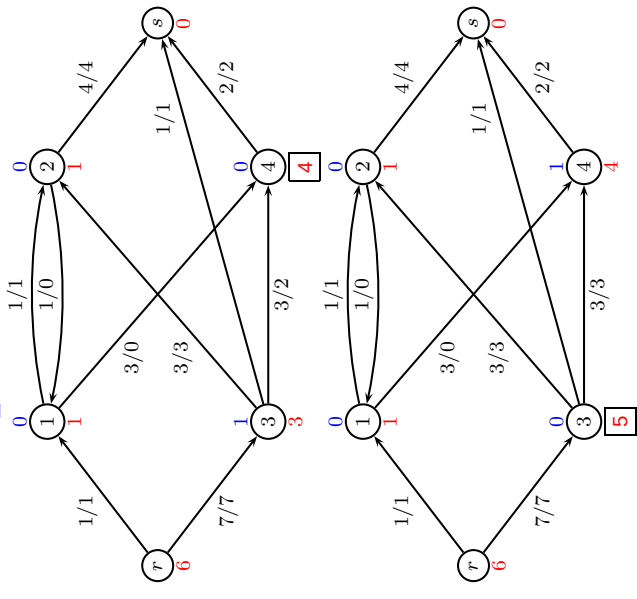
Example, iteration 5, 6



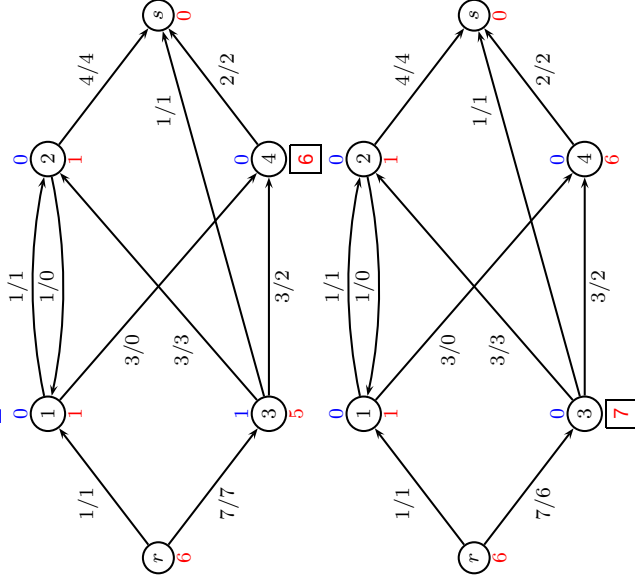
Example, iteration 3, 4



Example, iteration 7, 8



Example, iteration 9, 10



The Max Flow Problem – p. 25

Running times

- Notice, that every iteration provides a cut
- Analysis of the running times of the preflow-push algorithms is based on an analysis of the number of saturating pushes and non-saturating pushes.
- It can be shown that the push-relabel maximum flow algorithm can be implemented to run in time $O(n^2m)$ or $O(n^3)$.

The Max Flow Problem – p. 26