Network Flow

- Maximum Flow: Problem Formulation
- Maximum Flow: Upper Bound
- Maximum Flow: Adding flow along paths
- Residual Network and Augmenting Path
- Ford-Fulkerson Algorithm – Max-Flow-Min-Cut Theorem
- Edmond-Karp Algorithm
- Maximum Flow: Variants and Applications

Imdad ullah Khan

Max Flow : Problem Formulation

Input: A flow network $G = (V, E, c)$, $c : E \to \mathbb{R}^+$

$s \in V$ a source and $t \in V$ a sink

$f : E \to \mathbb{R}^+$ $(f_e = f(e))$ is a flow if it satisfies

1. capacity constraints $\forall\, e \in E \;:\; 0 \leq f_e \leq c_e$
2. flow conservation constraints $\forall\, v \in V\,, v \neq s, t \quad f^{out}(v) = f^{in}(v)$

$$size(f) \;=\; f^{out}(s) \;=\; f^{in}(t)$$

Output: A flow f of maximum size

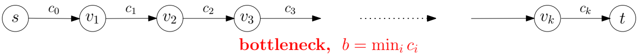
Let f be a flow in G and let $\left[A, \overline{A}\right]$ be any $s - t$ cut in G, then

$$size(f) \;\leq\; c\left(\left[A, \overline{A}\right]\right)$$

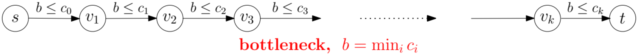
Max Flow : Adding Flow along a path

Greedy Algorithm – build up flow little bit at a time

1. Start with a 0 flow
   - ▷ Note: the flow f with $f_e = 0$ for all $e \in E$ satisfies both constraints
2. Add more flow to f via a $s - t$ path



Adding flow along a path keeps flow conservation constraints satisfied

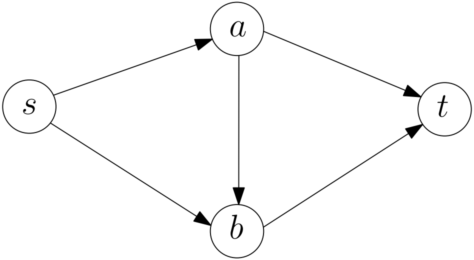


Adding flow equal to path-bottleneck keeps capacity constraints satisfied

The two ensure any intermediate flow by greedy algorithm is valid

Max Flow : Problems with the algorithm

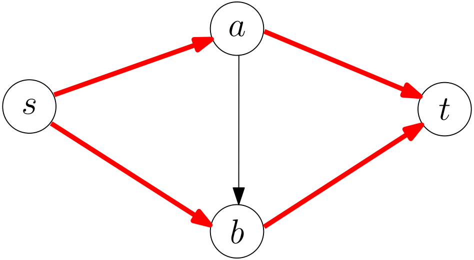
Consider the flow network – all edge capacity are 1



Max Flow : Problems with the algorithm

Consider the flow network – all edge capacity are 1

The max flow clearly is of size 2

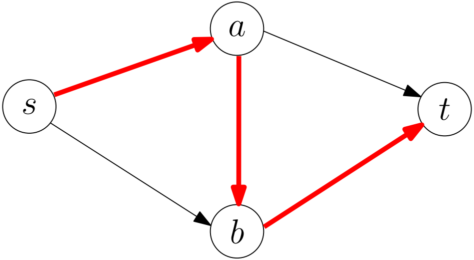


Max Flow : Problems with the algorithm

Consider the flow network – all edge capacity are 1

The max flow clearly is of size 2

If the greedy algorithm adds a flow of size 1 via the $s-t$ path s, a, b, t



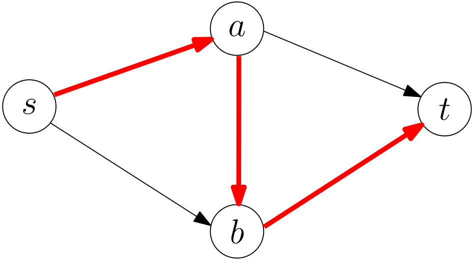
Max Flow : Problems with the algorithm

Consider the flow network – all edge capacity are 1

The max flow clearly is of size 2

If the greedy algorithm adds a flow of size 1 via the $s - t$ path s, a, b, t

No $s - t$ path in the remaining graph



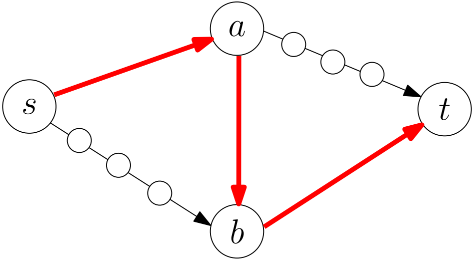
Max Flow : Problems with the algorithm

Consider the flow network – all edge capacity are 1

The max flow clearly is of size 2

If the greedy algorithm adds a flow of size 1 via the $s - t$ path s, a, b, t

No $s - t$ path in the remaining graph

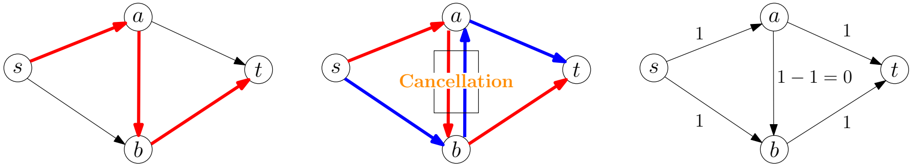


▷ **The issue is not the hop-length of paths**

Max Flow : Fix for the algorithm

A more general way of pushing further flow is to

1. Push forward flow on edges where some capacity is remaining
2. Cancel existing flow on edges ▷ pushing flow backward



- Add one unit of flow via the s, a, b, t path
- Add one unit of flow via the s, b, a, t path

  ▷ $ba \notin E$, but we cancel the existing flow on edge $ab \in E$

- Add the red and blue flows

Max Flow : Residual Network

Cancellation of existing flows on edges (if need be) is the right framework to add more flow

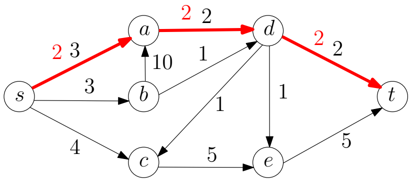
Residual network

- Porvides, a systematic way to search for the right places to cancel flow and adding more flow
- Associated with a flow f, it encodes "places" where f can be increased (by adding new flows and canceling existing flows)

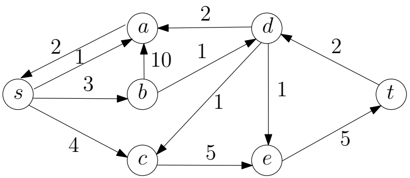
Max Flow : Residual Network

Given network G and flow f, the residual graph G_f of G w.r.t f is

- Vertex set of G_f is the same as that of G
- Forward edges: For each $e = uv$ of G on which $f_e < c_e$, there is an edge $e = uv$ in G_f with capacity $c_e - f_e > 0$
- Backward edges: For each edge $e = uv$ of G on which $f_e > 0$, there is an edge $e' = vu$ in G_f with capacity f_e



Flow network with flow shown in red



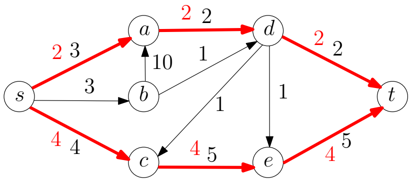
The corresponding residual network

▷ For any G and f, G_f has at most twice as many edges as G

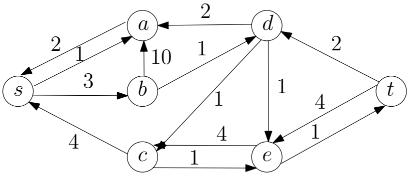
Max Flow : Residual Network

Given network G and flow f, the residual graph G_f of G w.r.t f is

- Vertex set of G_f is the same as that of G
- Forward edges: For each $e = uv$ of G on which $f_e < c_e$, there is an edge $e = uv$ in G_f with capacity $c_e - f_e > 0$
- Backward edges: For each edge $e = uv$ of G on which $f_e > 0$, there is an edge $e' = vu$ in G_f with capacity f_e



Flow network with flow shown in red



The corresponding residual network

▷ For any G and f, G_f has at most twice as many edges as G

Max Flow: Augmenting Path

An **augmenting path** is a simple $s - t$ path in the residual graph G_f

$\triangleright$ It is used to augment the flow f

For an augmenting path P in G_f, $bottleneck(P, f) = \min_{e \in P} c'_e$

$\triangleright$ the minimum residual capacity of any edge on P in G_f

Note c'_e is the residual capacity of the edge e (its capacity in G_f)

Algorithm AUGMENT(P, f) augment flow using a path P in G_f

$b \leftarrow bottleneck(P, f)$
$f' \leftarrow f$
for each edge $e = uv \in P$ **do**
  if e is a forward edge **then**
    $f'_e \leftarrow f_e + b$
  else if e is a backward edge **then**
    $f'_{vu} \leftarrow f_{vu} - b$

Max Flow : Correctness of $\textsc{augment}(P, f)$

The output f' of $\textsc{augment}(P, f)$ is a flow

f' satisfies capacity constraints $\quad \forall\, e \in E \;:\; 0 \leq f'_e \leq c'_e$

$$s \xrightarrow{b \leq c_0} v_1 \xrightarrow{b \leq c_1} v_2 \xrightarrow{b \leq c_2} v_3 \xrightarrow{b \leq c_3} \cdots\cdots \longrightarrow v_k \xrightarrow{b \leq c_k} t$$

bottleneck, $b = \min_i c_i$

- **Case 1:** e is not on P, $f'_e = f_e$
- **Case 2:** If $e \in P$ is a forward edge $(c'_e = c_e - f_e)$, then $f'_e = f_e + b$
  - Since $0 < b \leq c'_e = c_e - f_e$
  - We have $0 \leq f_e \leq f'_e = f_e + b \leq f_e + c_e - f_e = c_e$
- **Case 3:** If $e \in P$ is a backward edge $(c'_e = f_e)$, then $f'_e = f_e - b$
  - Since $0 < b \leq c'_e = f_e$
  - We have $c_e \geq f_e \geq f'_e = f_e - b \geq f_e - f_e = 0$

Max Flow : Correctness of $\textsc{augment}(P, f)$

The output f' of $\textsc{augment}(P, f)$ is a flow

f' satisfies the flow conservation $\quad \forall\ v \in V\ , v \neq s, t \quad f'^{out}(v) = f'^{in}(v)$

- **Case 1:** e_{in} is forward edge and e_{out} is forward edge
  - $f'^{out}(v) = f^{out}(v) + b$ as $f'_{vy} = f_{vy} + b$ and $f'^{in}(v) = f^{in}(v) + b$ as $f'_{xv} = f_{xv} + b$
  - Hence change in in-flow to v = change in out-flow from v
- **Case 2:** e_{in} is reverse edge and e_{out} is forward edge
  - $f'^{out}(v) = f^{out}(v) - b + b$ as $f'_{vx} = f_{vx} - b$ (since vx is a reverse edge) and $f'_{vy} = f_v y + b$ (since vy is a forward edge)
  - Hence $f'^{out}(v) = f'^{in}(v)$
- **Case 3:** e_{in} is reverse edge and e_{out} is reverse edge
- **Case 4:** e_{in} is forward edge and e_{out} is reverse edge