Network Algorithm UE709

Rabih AMHAZ

Head of Digital department at Icam Strasbourg-Europe Campus
Researcher in AI and Blockchain at ICube Laboratory, CSTB team

February 14, 2024

Maximum Flow

Maximum Flow : Question

With an infinit input source, how much "flow" can we push through the network given that each edge has a certain capacity?

Edges capacity : can receive a certain amount of flow.

Maximum Flow

Maximum Flow : Question

With an infinit input source, how much "flow" can we push through the network given that each edge has a certain capacity?

Edges capacity : can receive a certain amount of flow.

MaxFlow: Ford Fulkerson

Ford-Fulkerson algorithm

method finds augmenting paths through the residual graph and augments the flow until no more augmenting paths can be found.

1. **Augmenting path:** a path of edges in the residual graph with unused capacity.
2. **bottleneck value:** smallest edge of a path
3. **Residual edges:** exist to "undo" bad augmenting flow paths wich do not lead to a maximum flow.
4. **the residual graph:** flow graph ≡ residual graph

MaxFlow: Ford Fulkerson

Ford-Fulkerson algorithm

method finds augmenting paths through the residual graph and augments the flow until no more augmenting paths can be found.

1. **Augmenting path:** a path of edges in the residual graph with unused capacity.
2. **bottleneck value:** smallest edge of a path
3. **Residual edges:** exist to "undo" bad augmenting flow paths wich do not lead to a maximum flow.
4. **the residual graph:** flow graph $\equiv$ residual graph

MaxFlow: Ford Fulkerson

Ford-Fulkerson algorithm

method finds augmenting paths through the residual graph and augments the flow until no more augmenting paths can be found.

1. **Augmenting path:** a path of edges in the residual graph with unused capacity.
2. **bottleneck value:** smallest edge of a path
3. **Residual edges:** exist to "undo" bad augmenting flow paths wich do not lead to a maximum flow.
4. **the residual graph:** flow graph ≡ residual graph

MaxFlow: Ford Fulkerson

Ford-Fulkerson algorithm

method finds augmenting paths through the residual graph and augments the flow until no more augmenting paths can be found.

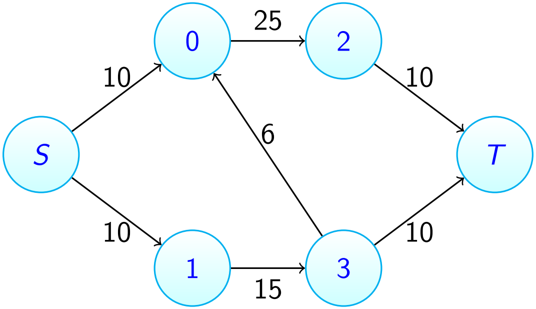
1. **Augmenting path:** a path of edges in the residual graph with unused capacity.
2. **bottleneck value:** smallest edge of a path
3. **Residual edges:** exist to "undo" bad augmenting flow paths wich do not lead to a maximum flow.
4. **the residual graph:** flow graph ≡ residual graph

Ford-Fulkerson algorithm

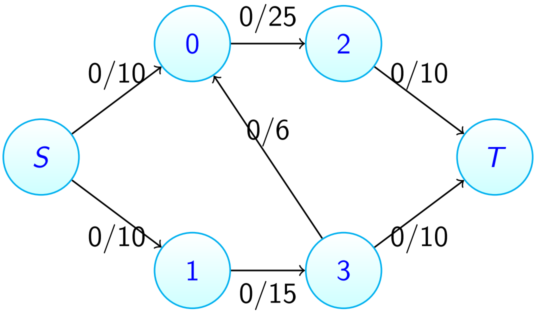
method finds augmenting paths through the residual graph and augments the flow until no more augmenting paths can be found.

1. **Augmenting path:** a path of edges in the residual graph with unused capacity.
2. **bottleneck value:** smallest edge of a path
3. **Residual edges:** exist to "undo" bad augmenting flow paths wich do not lead to a maximum flow.
4. **the residual graph:** flow graph $\equiv$ residual graph

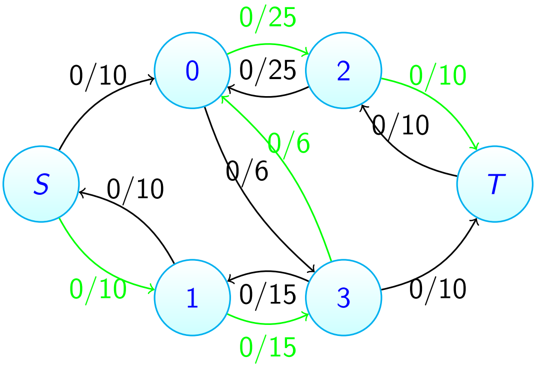
MaxFlow: Ford Fulkerson



MaxFlow: Ford Fulkerson

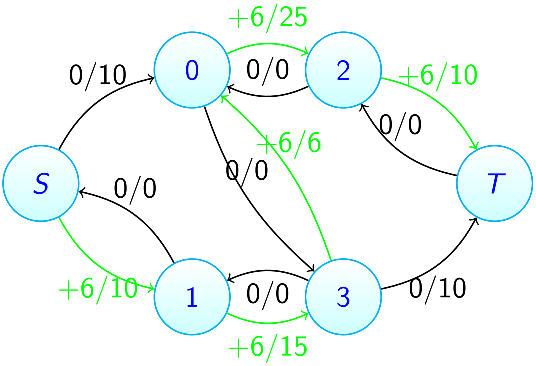


MaxFlow: Ford Fulkerson



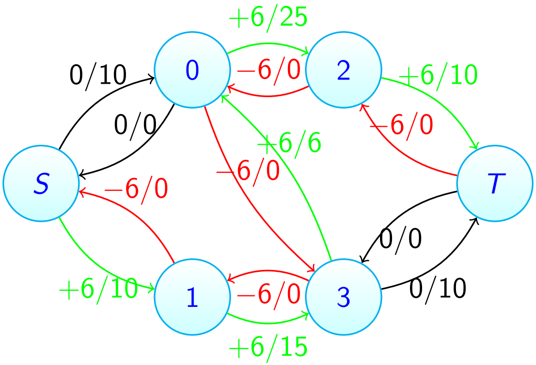
MaxFlow: Ford Fulkerson

Path 1



MaxFlow: Ford Fulkerson

Path 1



MaxFlow: Ford Fulkerson

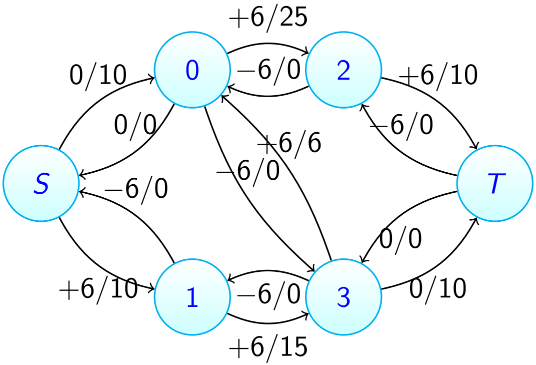
Path 1

Bottleneck Value of this path

min(10 - 0 ; 15 - 0 ; 6 - 0 ; 25 - 0 ; 10 - 0) = 6

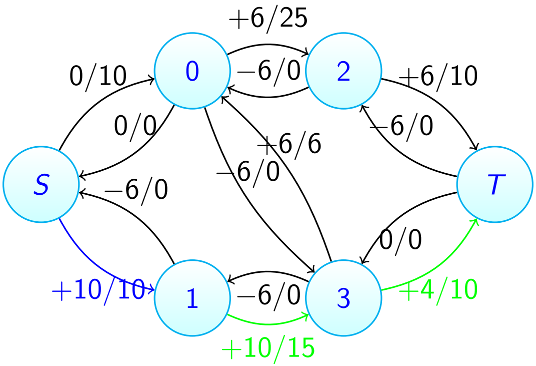
MaxFlow: Ford Fulkerson

Path 2



MaxFlow: Ford Fulkerson

Path 2



MaxFlow: Ford Fulkerson

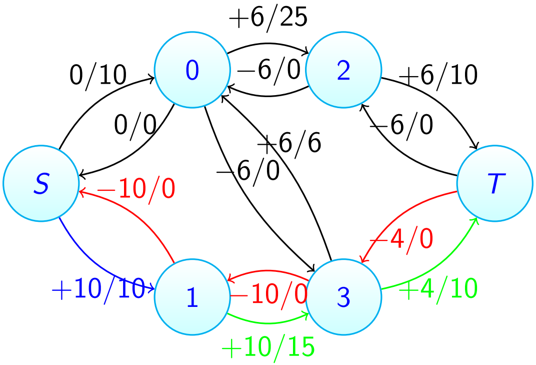
Path 2

Bottleneck Value of this path

min(10 - 6 ; 15 - 6 ; 10 - 4) $= 4$

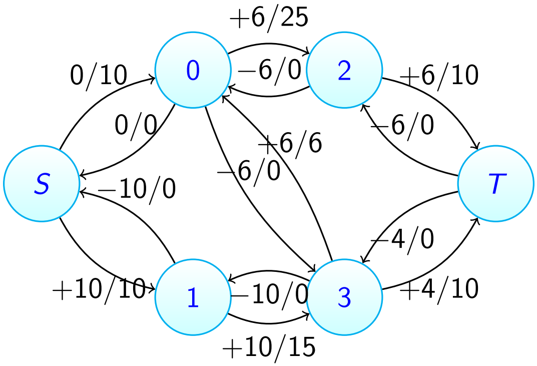
MaxFlow: Ford Fulkerson

Path 2



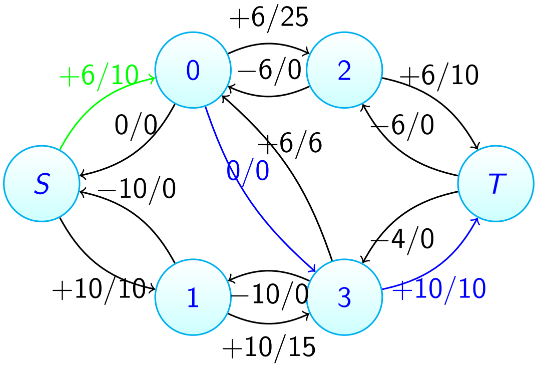
MaxFlow: Ford Fulkerson

Path 3



MaxFlow: Ford Fulkerson

Path 3



MaxFlow: Ford Fulkerson

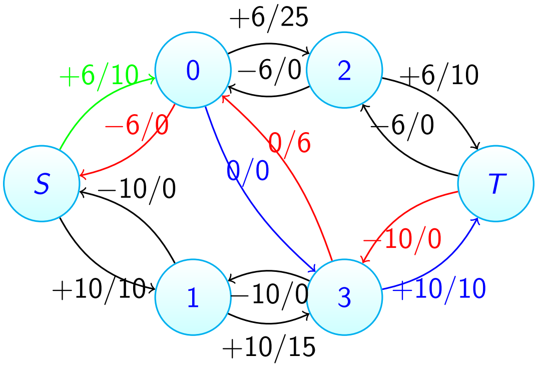
Path 3

Bottleneck Value of this path

min(10 - 0 ; 0 - (- 6) ; 10 - 4) = 6

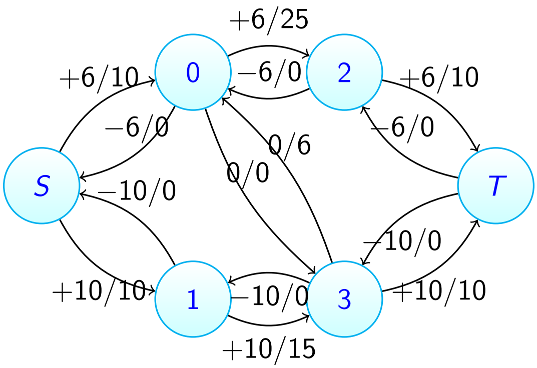
MaxFlow: Ford Fulkerson

Path 3



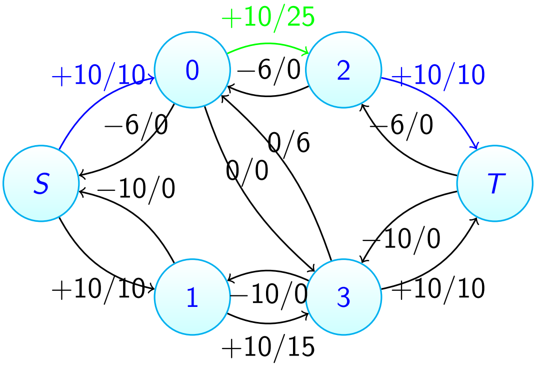
MaxFlow: Ford Fulkerson

Path 4



MaxFlow: Ford Fulkerson

Path 4



MaxFlow: Ford Fulkerson

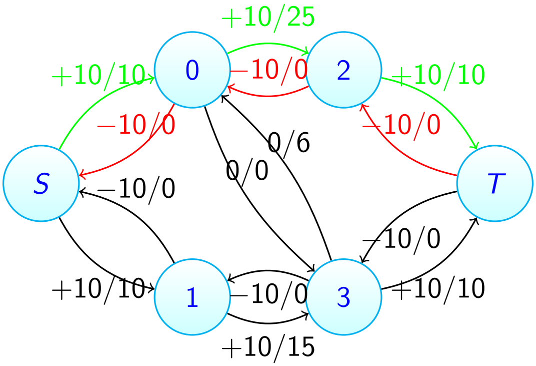
Path 4

Bottleneck Value of this path

min(10 - 6 ; 25 - 6 ; 10 - 6) = 4

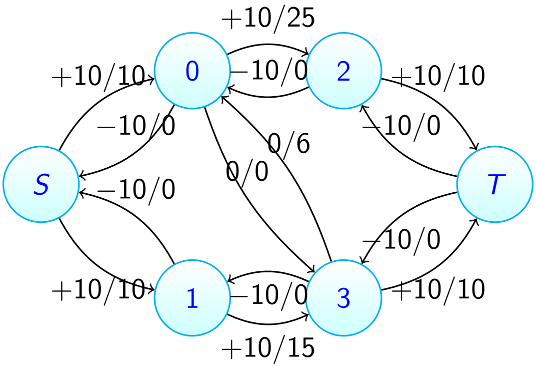
MaxFlow: Ford Fulkerson

Path 4



MaxFlow: Ford Fulkerson

Can We found other ways ?



MaxFlow: Ford Fulkerson

The Maximum Flow of this graph

No more augmenting paths

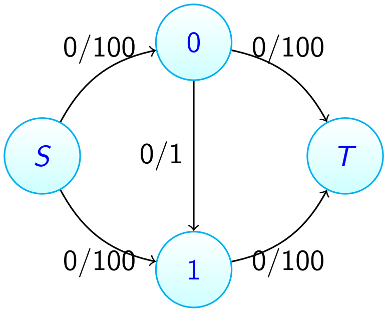
maxflow $= 6 + 4 + 6 + 4$ (the sum of all bottleneck values)

Ford-Fulkerson

How we found or we try to build the augmented path?

We use the DFS approach to acheive the destination.
Complexity of the algorithm is $O(Ef)$ with E is number of edges and f is the value of maximum flow.

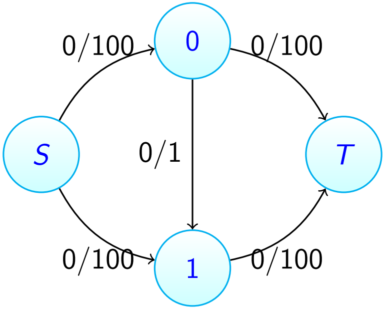
MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

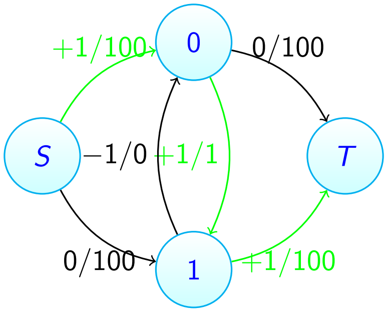
MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

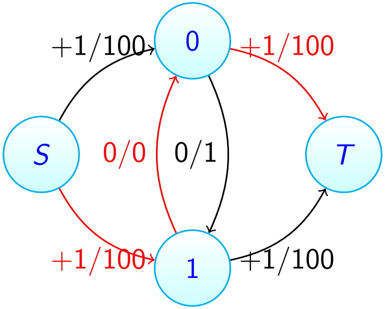
MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

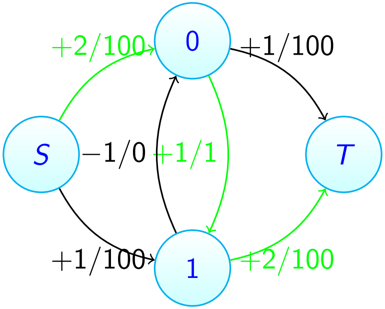
MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

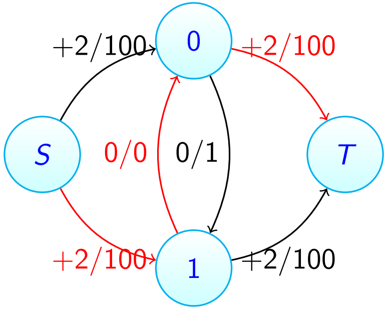
MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

MaxFlow: Ford Fulkerson



We are using DFS

let us consider that DFS chooses edges in random order, so it is possible to pick the middle edge every time when finding the augmented path.

Maximum Flow

For this reason we can switch to other algorithms:

- Edmonds-Karp: using BFS for finding augmenting paths $O(E^2V)$
- Capacity scaling: Adds a heuristic on top of Ford-Fulkerson to pick larger paths first $O(E^2log(V))$
- Dinic's: Uses combination of BFS and DFS to find aumenting paths $O(V^2E)$
- Push Relabel: uses a concept of maintaining a "preflow" instead of finding the augmenting paths to acheive a max-flow solution $O(V^2E)$ or $O(V^2\sqrt{E})$ variant.

Maximum Flow

For this reason we can switch to other algorithms:

- Edmonds-Karp: using BFS for finding augmenting paths $O(E^2V)$
- Capacity scaling: Adds a heuristic on top of Ford-Fulkerson to pick larger paths first $O(E^2 log(V))$
- Dinic's: Uses combination of BFS and DFS to find aumenting paths $O(V^2E)$
- Push Relabel: uses a concept of maintaining a "preflow" instead of finding the augmenting paths to acheive a max-flow solution $O(V^2E)$ or $O(V^2\sqrt{E})$ variant.

Maximum Flow

For this reason we can switch to other algorithms:

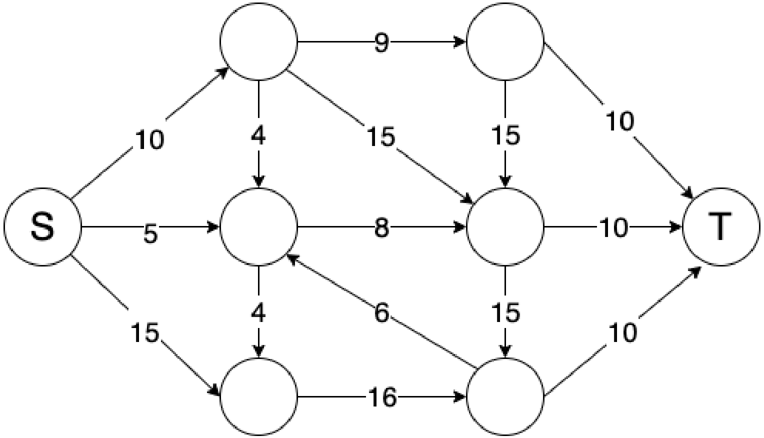
- Edmonds-Karp: using BFS for finding augmenting paths $O(E^2V)$
- Capacity scaling: Adds a heuristic on top of Ford-Fulkerson to pick larger paths first $O(E^2log(V))$
- Dinic's: Uses combination of BFS and DFS to find aumenting paths $O(V^2E)$
- Push Relabel: uses a concept of maintaining a "preflow" instead of finding the augmenting paths to acheive a max-flow solution $O(V^2E)$ or $O(V^2\sqrt{E})$ variant.

Maximum Flow

For this reason we can switch to other algorithms:

- Edmonds-Karp: using BFS for finding augmenting paths $O(E^2V)$
- Capacity scaling: Adds a heuristic on top of Ford-Fulkerson to pick larger paths first $O(E^2log(V))$
- Dinic's: Uses combination of BFS and DFS to find aumenting paths $O(V^2E)$
- Push Relabel: uses a concept of maintaining a "preflow" instead of finding the augmenting paths to acheive a max-flow solution $O(V^2E)$ or $O(V^2\sqrt{E})$ variant.

EX: Ford Fulkerson


S
T
10
5
15
9
4
15
8
4
6
16
15
15
10
10
10