Ford–Fulkerson algorithm

Graph theory

University of Szeged

Szeged, 2023.

Def. Maximum flow (in a network N): A flow of maximum value.

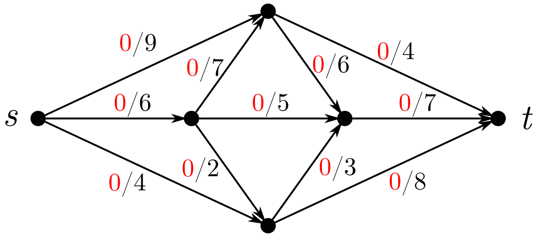
Lemma. f is a flow in a given network N. The flow f is not a maximum flow if and only if there exists an augmenting path in N with respect to f.

Def. Maximum flow (in a network N): A flow of maximum value.

Lemma. f is a flow in a given network N. The flow f is not a maximum flow if and only if there exists an augmenting path in N with respect to f.

Ford–Fulkerson algorithm. An efficient algorithm for finding a maximum flow in the input network N:

1. Start with the constant-0 flow (it is always feasible): $f := \underline{0}$.



Ford–Fulkerson algorithm. An efficient algorithm for finding a maximum flow in the input network N:

1. Start with the constant-0 flow (it is always feasible): $f := \underline{0}$.

2. Search for an augmenting path w.r.t. f (and find one in polinomial time if such a path exists).

2/a. If an augmenting patch P was found, then augment f along P and repeat Step 2.

2/b. If there is no augmenting path, provide an $[S, T]$-cut which proves that the actual flow f has maximum value, and terminate. The output is f and the $[S, T]$-cut.

Ford–Fulkerson algorithm. An efficient algorithm for finding a maximum flow in the input network N:

1. Start with the constant-0 flow (it is always feasible): $f := \underline{0}$.

2. Search for an augmenting path w.r.t. f (and find one in polinomial time if such a path exists).

2/a. If an augmenting patch P was found, then augment f along P and repeat Step 2.

2/b. If there is no augmenting path, provide an $[S, T]$-cut which proves that the actual flow f has maximum value, and terminate. The output is f and the $[S, T]$-cut.

Note. Edmonds and Karp proved that if Step 2 is implemented using a breadth-first search (i.e. we always pick a shortest augmenting path), then the number of iterations of Step 2 is polynomial. (When one picks an arbitrary augmenting path in Step 2, we can run into an infinite number of iterations!)

The key is how Step 2 implemented:

Augmenting path finder (subroutine for Step 2).

Input: A network N and a flow f in it.

The key is how Step 2 implemented:

Augmenting path finder (subroutine for Step 2).

Input: A network N and a flow f in it.

Build an auxiliary directed graph G_f as follows:
The vertices of G_f are exactly the vertices of N.
For each edge $e = \overrightarrow{uv}$ of N,

- if $f(e) < c(e)$, then add an edge from u to v in G_f.
- if $f(e) > 0$, then add an edge from v to u in G_f.

(And there are no other edges in G_f.)

The key is how Step 2 implemented:

Augmenting path finder (subroutine for Step 2).

Input: A network N and a flow f in it.

Build an auxiliary directed graph G_f as follows:
The vertices of G_f are exactly the vertices of N.
For each edge $e = \overrightarrow{uv}$ of N,
- if $f(e) < c(e)$, then add an edge from u to v in G_f.
- if $f(e) > 0$, then add an edge from v to u in G_f.

(And there are no other edges in G_f.)

It is easy to see that there exists an augmenting path wrt. f in N if and only if there exists a directed path from the source s to the sink t in G_f. The latter problem can be solved using a breadth-first search.

Augmenting path finder (subroutine for Step 2).

Input: A network N and a flow f in it.

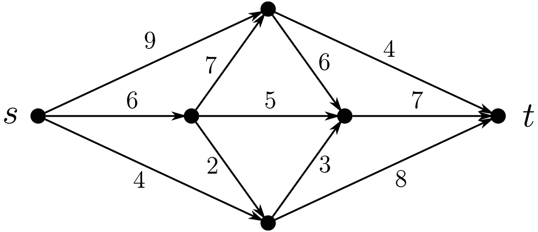
Build an auxiliary directed graph G_f as follows:
The vertices of G_f are exactly the vertices of N.
For each edge $e = \overrightarrow{uv}$ of N,

- if $f(e) < c(e)$, then add an edge from u to v in G_f.
- if $f(e) > 0$, then add an edge from v to u in G_f.

It is easy to see that there exists an augmenting path wrt. f in N if and only if there exists a directed path from the source s to the sink t in G_f. The latter problem can be solved using a breadth-first search.

It can be proved that if there is no $\overrightarrow{st}$-path in G_f, and S is the set of vertices that were reached by the BFS in G_f, and T is the set of unreached vertices, then the capacity of this $[S, T]$-cut equals to the value of f, proving that f has maximum value.

(G, s, t, c)

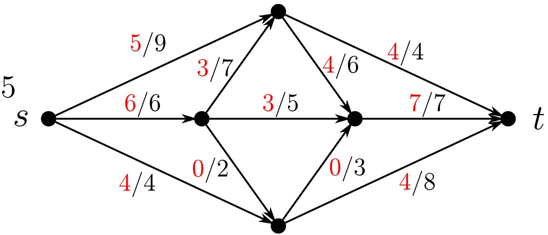


Example. Find a maximum flow in the following network.

(G, s, t, c)

f flow

$\mathrm{val}(f) = 15$



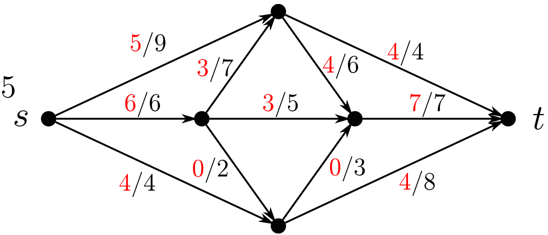
Example. Find a maximum flow in the following network.

Solution. The starting point of the Ford–Fulkerson-algorithm is an arbitrary feasible flow. In practice, we can always choose the everywhere-zero flow. For pedagogical purposes, in this presentation our starting point is a less trivial feasible flow f, assuming that several augmentation has been already performed.

(G, s, t, c)

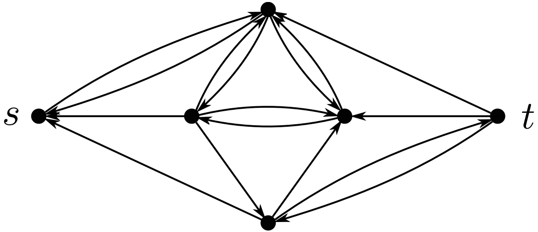
f flow

$\mathsf{val}(f) = 15$



Searching for augmenting path

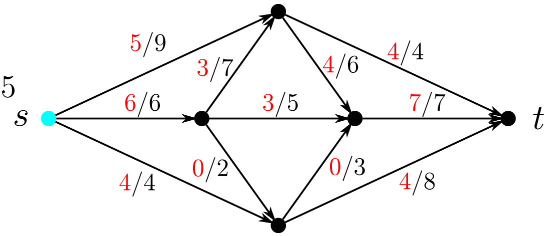
G_f



(G, s, t, c)

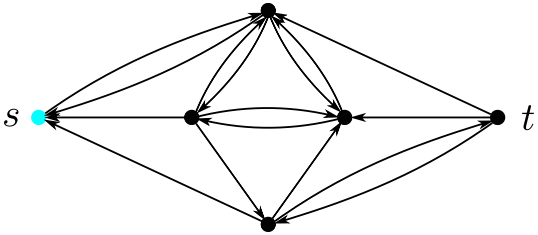
f flow

$\mathrm{val}(f) = 15$

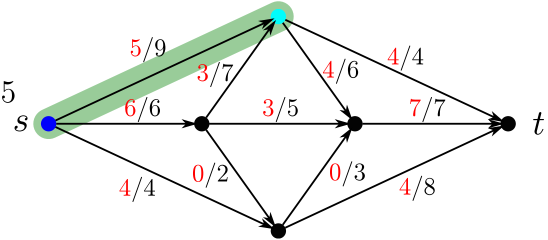


Searching for augmenting path

G_f

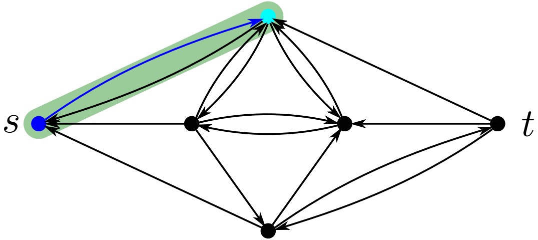


(G, s, t, c)
f flow
$\text{val}(f) = 15$



Searching for augmenting path

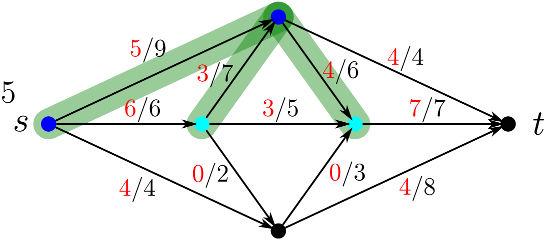
G_f



(G, s, t, c)

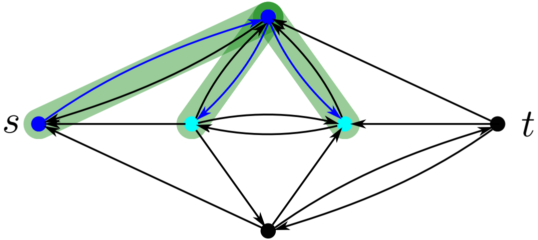
f flow

$\mathsf{val}(f) = 15$



Searching for augmenting path

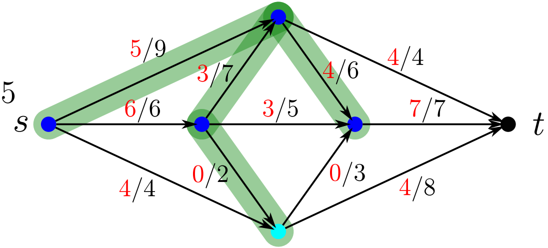
G_f



(G, s, t, c)

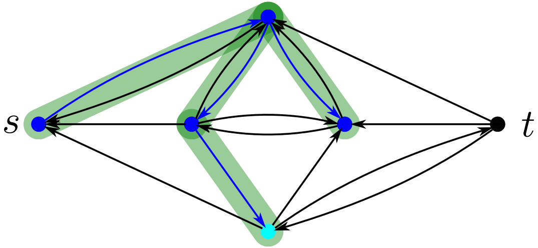
f flow

$\text{val}(f) = 15$



Searching for augmenting path

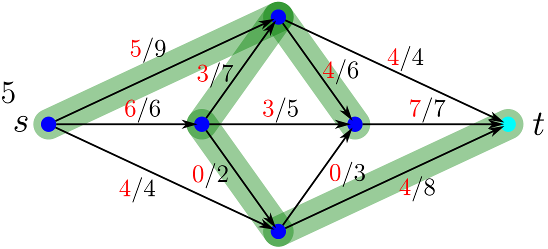
G_f



(G, s, t, c)

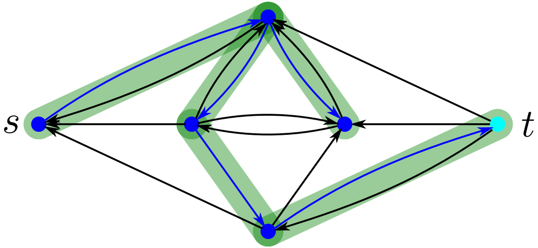
f flow

$\mathsf{val}(f) = 15$



Searching for augmenting path

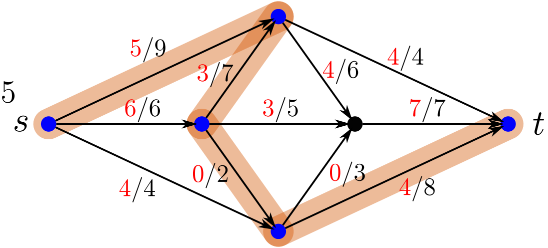
G_f



(G, s, t, c)

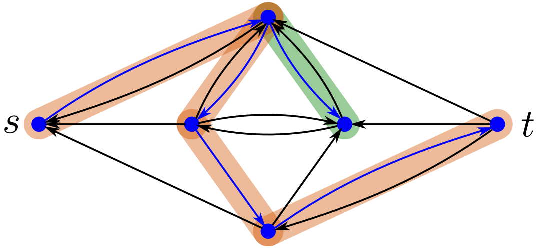
f flow

$\mathsf{val}(f) = 15$



An augmenting path was found.

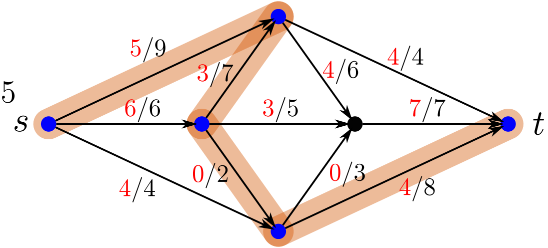
G_f



(G, s, t, c)

f flow

$\mathsf{val}(f) = 15$



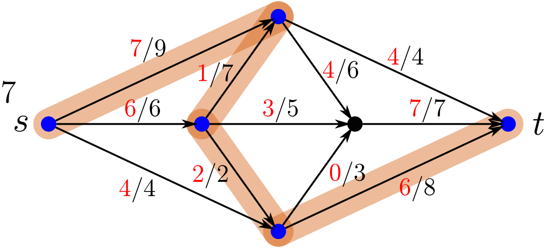
Augmentation: $\quad \delta = \min\{4, 3, 2, 4\} = 2$

G_f

(G, s, t, c)

f flow

$\mathsf{val}(f) = 17$



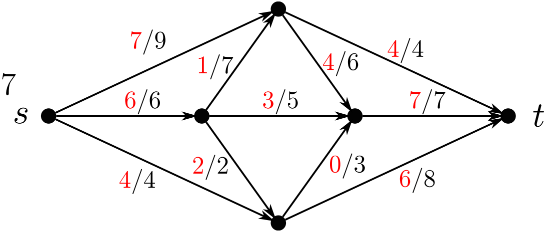
Augmentation: $\delta = \min\{4, 3, 2, 4\} = 2$

G_f

(G, s, t, c)

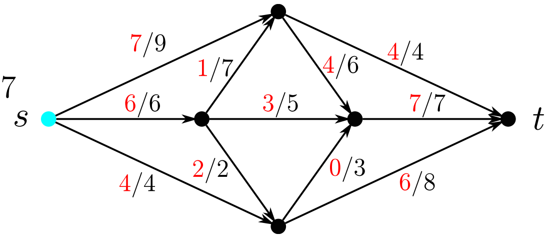
f flow

$\mathsf{val}(f) = 17$



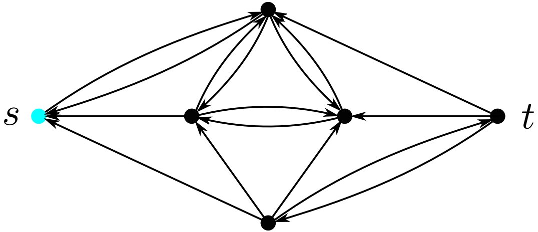
G_f

(G, s, t, c)
f flow
$\mathrm{val}(f) = 17$



Searching for augmenting path

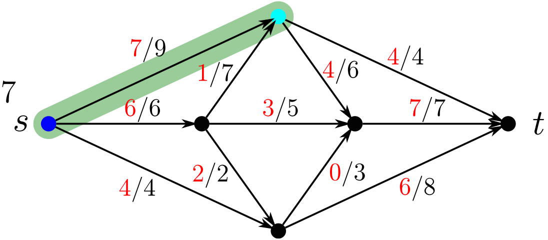
G_f



(G, s, t, c)

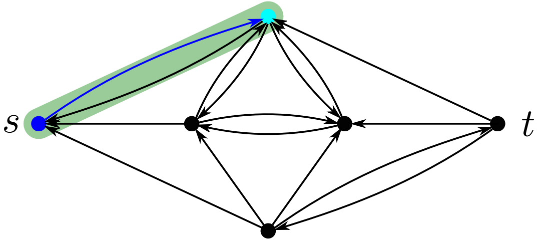
f flow

$\mathrm{val}(f) = 17$



Searching for augmenting path

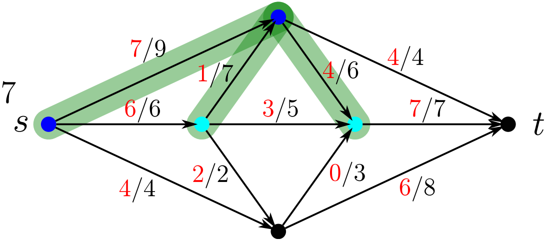
G_f



(G, s, t, c)

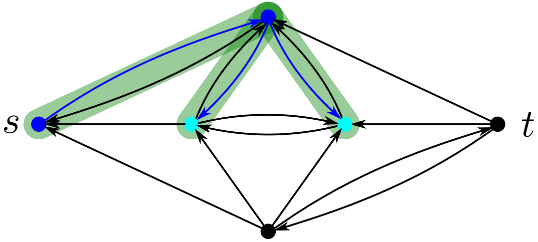
f flow

$\mathsf{val}(f) = 17$



Searching for augmenting path

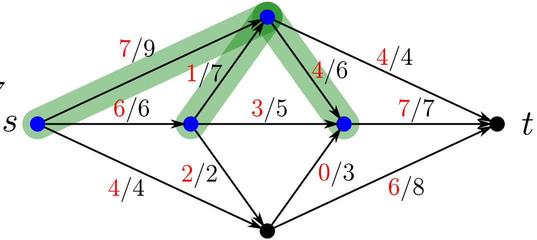
G_f



(G, s, t, c)

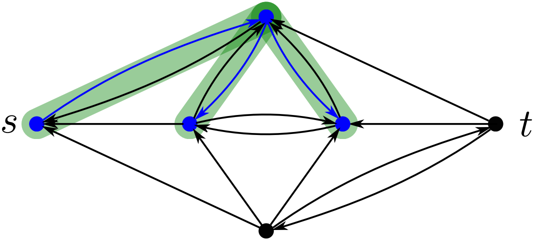
f flow

$\text{val}(f) = 17$



There is no augmenting path, STOP.

G_f

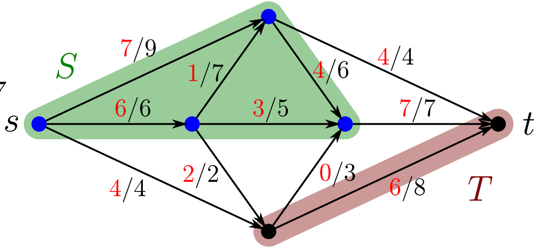


(G, s, t, c)

f flow

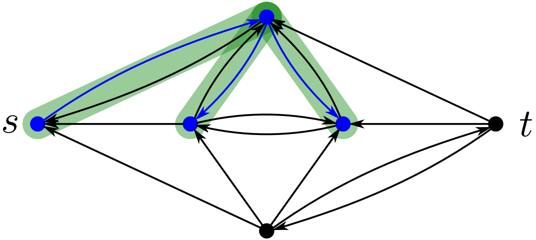
$\mathrm{val}(f) = 17$

$[S, T]$-cut



The endpoints of the partial augm. paths define a cut proving maximality.

G_f

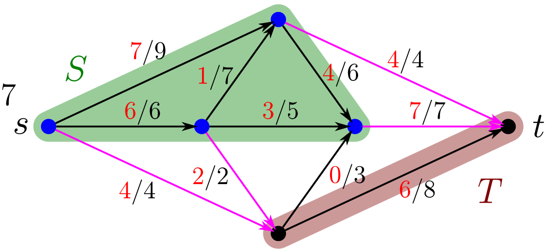


(G, s, t, c)

f flow

$\mathsf{val}(f) = 17$

$[S, T]$-cut



The endpoints of the partial augm. paths define a cut proving maximality.

(This $[S, T]$-cut is a minimum cut of the network.)

$$\max_{\mathfrak{f} \text{ flow}} \mathsf{val}(\mathfrak{f}) \leq c(S, T) = 4 + 2 + 7 + 4 = 17 = \mathsf{val}(f).$$

$$\Downarrow$$

f is a maximum flow, i.e. in this network the maximum flow value is 17. $\square$