25. Flow in Networks

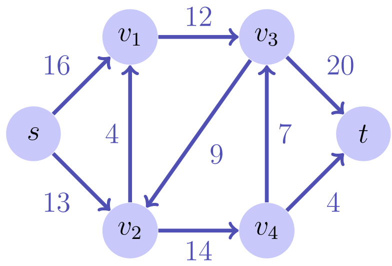
Flow Network, Maximal Flow, Cut, Rest Network, Max-flow Min-cut Theorem, Ford-Fulkerson Method, Edmonds-Karp Algorithm, Maximal Bipartite Matching [Ottman/Widmayer, Kap. 9.7, 9.8.1], [Cormen et al, Kap. 26.1-26.3]

Motivation

Modelling flow of fluents, components on conveyors, current in electrical networks or information flow in communication networks.

Flow Network

- *Flow network* $G = (V, E, c)$: directed graph with *capacities*
- Antiparallel edges forbidden: $(u, v) \in E \Rightarrow (v, u) \notin E$.
- Model a missing edge (u, v) by $c(u, v) = 0$.
- *Source* s and *sink* t: special nodes. Every node v is on a path between s and t : $s \rightsquigarrow v \rightsquigarrow t$

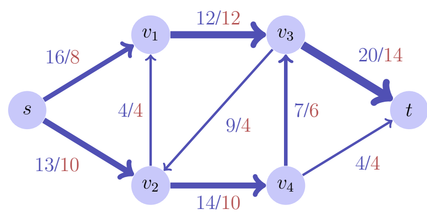


Flow

A *Flow* $f : V \times V \to \mathbb{R}$ fulfills the following conditions:

- *Bounded Capacity*: For all $u, v \in V$: $0 \leq f(u, v) \leq c(u, v)$.
- *Conservation of flow*: For all $u \in V \setminus \{s, t\}$:

$$\sum_{v \in V} f(v, u) - \sum_{v \in V} f(u, v) = 0.$$



Value of the flow: $w(f) = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$. Here $w(f) = 18$.

How large can a flow possibly be?

Limiting factors: cuts

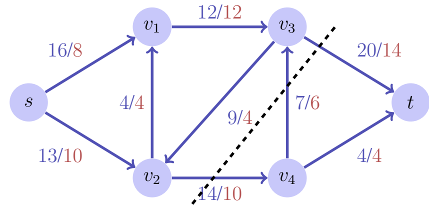
- *cut separating s from t*: Partition of V into S and T with $s \in S$, $t \in T$.
- *Capacity* of a cut: $c(S,T) = \sum_{v\in S, v'\in T} c(v,v')$
- *Minimal cut*: cut with minimal capacity.
- *Flow over the cut*: $f(S,T) = \sum_{v\in S, v'\in T} f(v,v') - \sum_{v\in S, v'\in T} f(v',v)$

How large can a flow possibly be?

For each flow and each cut it holds that $f(S,T) = w(f)$:

$$\begin{aligned} f(S,T) &= \sum_{v\in S, v'\in T} f(v,v') - \sum_{v\in S, v'\in T} f(v',v) \\ &= \sum_{v\in S, v'\in V} f(v,v') - \sum_{v\in S, v'\in S} f(v,v') - \sum_{v\in S, v'\in V} f(v',v) + \sum_{v\in S, v'\in S} f(v',v) \\ &= \sum_{v'\in V} f(s,v') - \sum_{v'\in V} f(v',s) \end{aligned}$$

Second equality: amendment, last equality: conservation of flow.

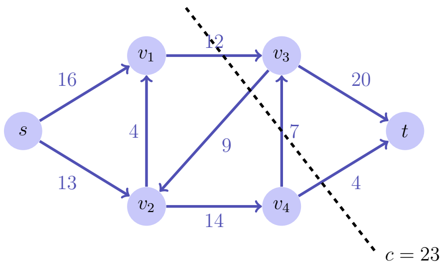


Maximal Flow ?

In particular, for each cut (S,T) of V.

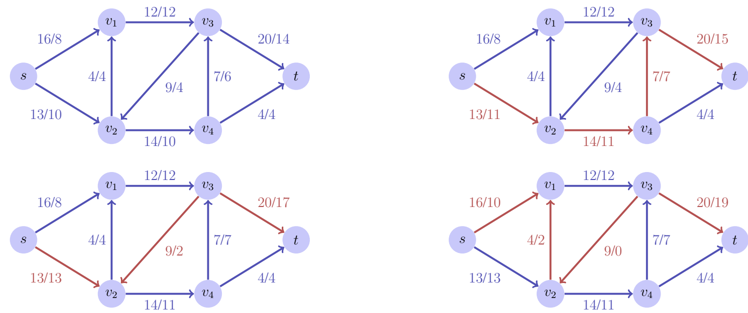
$$f(S,T) \le \sum_{v\in S, v'\in T} c(v,v') = c(S,T)$$

Will discover that equality holds for $\min_{S,T} c(S,T)$.



Maximal Flow ?

Naive Procedure



Conclusion: greedy increase of flow does not solve the problem.

The Method of Ford-Fulkerson

- Start with $f(u, v) = 0$ for all $u, v \in V$
- Determine rest network* G_f and expansion path in G_f
- Increase flow via expansion path*
- Repeat until no expansion path available.

*Will now be explained

Increase of flow, negative!

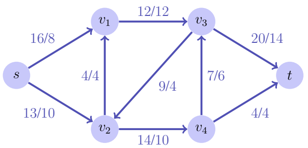
Let some flow f in the network be given.

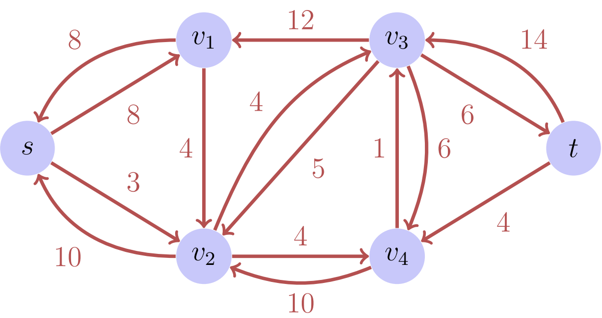
Finding:

- Increase of the flow along some edge possible, when flow can be increased along the edge,i.e. if $f(u, v) < c(u, v)$.
  Rest capacity $c_f(u, v) = c(u, v) - f(u, v)$.
- Increase of flow *against the direction* of the edge possible, if flow can be reduced along the edge, i.e. if $f(u, v) > 0$.
  Rest capacity $c_f(v, u) = f(u, v)$.

Rest Network

Rest network G_f provided by the edges with positive rest capacity:





Rest networks provide the same kind of properties as flow networks with the exception of permitting antiparallel edges

Observation

Theorem

Let $G = (V, E, c)$ be a flow network with source s and sink t and f a flow in G. Let G_f be the corresponding rest networks and let f' be a flow in G_f. Then $f \oplus f'$ defines a flow in G with value $w(f) + w(f')$.

$$(f \oplus f')(u, v) = \begin{cases} f(u, v) + f'(u, v) - f'(v, u) & (u, v) \in E \\ 0 & (u, v) \notin E. \end{cases}$$

Proof

Limit of capacity:

$$\begin{aligned}(f \oplus f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) \\ &\geq f(u,v) + f'(u,v) - f(u,v) = f'(u,v) \geq 0\end{aligned}$$

$$\begin{aligned}(f \oplus f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) \\ &\leq f(u,v) + f'(u,v) \\ &\leq f(u,v) + c_f(u,v) \\ &= f(u,v) + c(u,v) - f(u,v) = c(u,v).\end{aligned}$$

Proof

Conservation of flow

$$\begin{aligned}\sum_{u\in V}(f \oplus f')(u,v) &= \sum_{u\in V} f(u,v) + \sum_{u\in V} f'(u,v) - \sum_{u\in V} f'(v,u) \\ \text{(Flow conservation of } f \text{ and } f'\text{)} &= \sum_{u\in V} f(v,u) + \sum_{u\in V} f'(v,u) - \sum_{u\in V} f'(u,v) \\ &= \sum_{u\in V}(f \oplus f')(v,u)\end{aligned}$$

Beweis

Value of $f \oplus f'$ (in the sequel $N^+ := N^+(s)$, $N^- := N^-(s)$):

$$\begin{aligned}w(f \oplus f') &= \sum_{v\in N^+}(f \oplus f')(s,v) - \sum_{v\in N^-}(f \oplus f')(v,s) \\ &= \sum_{v\in N^+} f(s,v) + f'(s,v) - f'(v,s) - \sum_{v\in N^-} f(v,s) + f'(v,s) - f'(s,v) \\ &= \sum_{v\in N^+} f(s,v) - \sum_{v\in N^-} f(v,s) + \sum_{v\in N^+\cup N^-} f'(s,v) + \sum_{v\in N^+\cup N^-} f'(v,s) \\ &= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{v\in V} f'(s,v) + \sum_{v\in V} f'(v,s) \\ &= w(f) + w(f').\end{aligned}$$

■

Flow in G_f

expansion path p: path from s to t in the rest network G_f.

Rest capacity $c_f(p) = \min\{c_f(u,v) : (u,v) \text{ edge in } p\}$

Theorem

The mapping $f_p : V \times V \to \mathbb{R}$,

$$f_p(u,v) = \begin{cases} c_f(p) & \text{if } (u,v) \text{ edge in } p \\ 0 & \text{otherwise}\end{cases}$$

provides a flow in G_f with value $w(f_p) = c_f(p) > 0$.

[Proof: exercise]

Consequence

Strategy for an algorithm:

With an expansion path p in G_f the flow $f \oplus f_p$ defines a new flow with value $w(f \oplus f_p) = w(f) + w(f_p) > w(f)$

Max-Flow Min-Cut Theorem

Theorem

Let f be a flow in a flow network $G = (V, E, c)$ with source s and sink t. The following statementsa are equivalent:

1. *f is a maximal flow in G*
2. *The rest network G_f does not provide any expansion paths*
3. *It holds that $w(f) = c(S, T)$ for a cut (S, T) of G.*

Proof

- $(3) \Rightarrow (1)$:
  It holds that $w(f) \leq c(S, T)$ for all cuts S, T. From $w(f) = c(S, T)$ it follows that $w(f)$ is maximal.
- $(1) \Rightarrow (2)$:
  f maximal Flow in G. Assumption: G_f has some expansion path $w(f \oplus f_p) = w(f) + w(f_p) > w(f)$. Contradiction.

Proof $(2) \Rightarrow (3)$

Assumption: G_f has no expansion path. Define
$S = \{v \in V : \text{ there is a path } s \rightsquigarrow v \text{ in } G_f\}$. $(S, T) := (S, V \setminus S)$ is a cut:
$s \in S, t \notin S$. Let $u \in S$ and $v \in T$.

- If $(u, v) \in E$, then $f(u, v) = c(u, v)$, otherwise it would hold that $(u, v) \in E_f$.
- If $(v, u) \in E$, then $f(v, u) = 0$, otherwise it would hold that $c_f(u, v) = f(v, u) > 0$ and $(u, v) \in E_f$
- If $(u, v) \notin E$ and $(v, u) \notin E$, then $f(u, v) = f(v, u) = 0$.

Thus

$$
\begin{aligned}
w(f) = f(S, T) &= \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{v \in T} \sum_{u \in s} f(v, u) \\
&= \sum_{u \in S} \sum_{v \in T} c(u, v) - \sum_{v \in T} \sum_{u \in s} 0 = \sum_{u \in S} \sum_{v \in T} c(u, v) = c(S, T).
\end{aligned}
$$

Algorithm Ford-Fulkerson(G, s, t)

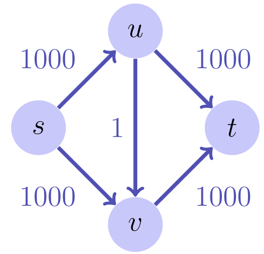
Input : Flow network $G = (V, E, c)$
Output : Maximal flow f.

for $(u, v) \in E$ **do**
    $f(u, v) \leftarrow 0$

while Exists path $p : s \rightsquigarrow t$ in rest network G_f **do**
    $c_f(p) \leftarrow \min\{c_f(u, v) : (u, v) \in p\}$
    foreach $(u, v) \in p$ **do**
        if $(u, v) \in E$ **then**
            $f(u, v) \leftarrow f(u, v) + c_f(p)$
        else
            $f(v, u) \leftarrow f(u, v) - c_f(p)$

Analysis

- The Ford-Fulkerson algorithm does not necessarily have to converge for irrational capacities. For integers or rational numbers it terminates.
- For an integer flow, the algorithms requires maximally $w(f_{\max})$ iterations of the while loop. Search a single increasing path (e.g. with DFS or BFS $\mathcal{O}(|E|))$ Therefore $\mathcal{O}(f_{\max}|E|)$.



With an unlucky choice the algorithm may require up to 2000 iterations here.

Edmonds-Karp Algorithm

Choose in the Ford-Fulkerson-Method for finding a path in G_f the expansion path of shortest possible length (e.g. with BFS)

Edmonds-Karp Algorithm

Theorem

When the Edmonds-Karp algorithm is applied to some integer valued flow network $G = (V, E)$ with source s and sink t then the number of flow increases applied by the algorithm is in $\mathcal{O}(|V| \cdot |E|)$

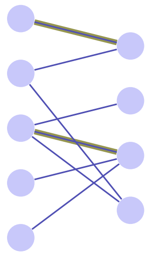
[Without proof]

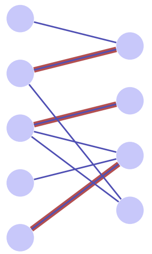
Application: maximal bipartite matching

Given: bipartite undirected graph $G = (V, E)$.

Matching M: $M \subseteq E$ such that $|\{m \in M : v \in m\}| \leq 1$ for all $v \in V$.

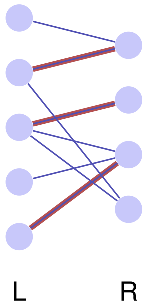
Maximal Matching M: Matching M, such that $|M| \geq |M'|$ for each matching M'.

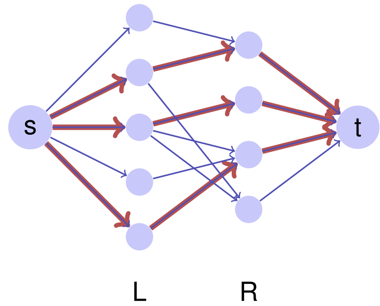




Corresponding flow network

Construct a flow network that corresponds to the partition L, R of a bipartite graph with source s and sink t, with directed edges from s to L, from L to R and from R to t. Each edge has capacity 1.





Integer number theorem

Theorem

If the capacities of a flow network are integers, then the maximal flow generated by the Ford-Fulkerson method provides integer numbers for each $f(u, v)$, $u, v \in V$.

[without proof]

Consequence: Ford-Fulkerson generates for a flow network that corresponds to a bipartite graph a maximal matching $M = \{(u, v) : f(u, v) = 1\}$.