Computer Vision I

Jannik Irmai, Jannik Presberger, Bjoern Andres

Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/24-winter/cv1/

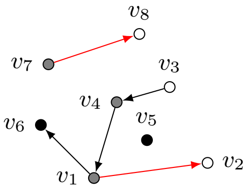
Winter Term 2024/2025

Excursus: Maximum st-Flow and Minimum st-Cut

- Maximum st-Flow Problem
- Residual networks and augmenting paths
- Minimum st-Cut Problem
- Maximum st-Flow/Minimum st-Cut Theorem
- Ford-Fulkerson-Algorithm

For any directed graph (V, E), any $U \subseteq V$ and any $W \subseteq V$ let

$$UW := \{uv \in E \mid u \in U \wedge w \in W\} \ .$$



$U = \{v_1, v_4, v_7\} \quad W = \{v_2, v_3, v_8\} \quad UW = \{v_1v_2, v_7v_8\}$

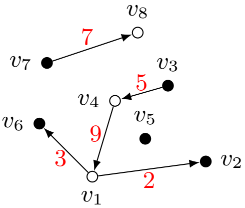
Definition 1. For any directed graph (V, E) and any $f \in \mathbb{N}_0^E$, the maps $\varphi^+, \varphi^-, \varphi : 2^V \to \mathbb{Z}$ such that

$$\forall U \in 2^V \quad \varphi_U^+ = \sum_{uv \in UU^c} f_{uv} \qquad (1)$$

$$\varphi_U^- = \sum_{vu \in U^cU} f_{vu} \qquad (2)$$

$$\varphi_U = \varphi_U^+ - \varphi_U^- \qquad (3)$$

are called the **outflux**, **influx** and **flux** in (V, E) wrt. f.

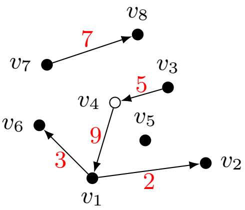


$$U = \{v_1, v_4, v_8\}$$
$$\varphi_U^+ = 3 + 2$$
$$\varphi_U^- = 7 + 5$$
$$\varphi_U = -7$$

For any $u \in V$,

$$\varphi_u^+ := \varphi_{\{u\}}^+$$
$$\varphi_u^- := \varphi_{\{u\}}^-$$
$$\varphi_u := \varphi_{\{u\}}$$

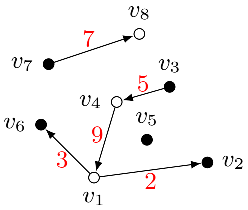
are called the **outflux**, **influx** and **flux** of u in (V, E) wrt. f.



$$\varphi_{v_4}^+ = 9$$
$$\varphi_{v_4}^- = 5$$
$$\varphi_{v_4} = 4$$

Lemma 1. For any directed graph (V, E), any $f \in \mathbb{N}_0^E$ and any $U \subseteq V$

$$\varphi_U = \sum_{u \in U} \varphi_u \quad . \tag{4}$$



Proof.

$$
\begin{aligned}
\varphi_U &= \sum_{uv \in UU^c} f_{uv} - \sum_{vu \in U^c U} f_{vu} \\
&= \left(\sum_{uv \in UV} f_{uv} - \sum_{uu' \in UU} f_{uu'} \right) - \left(\sum_{vu \in VU} f_{vu} - \sum_{u'u \in UU} f_{uu'} \right) \\
&= \sum_{uv \in UV} f_{uv} - \sum_{vu \in VU} f_{vu} \\
&= \sum_{u \in U} \left(\sum_{vw \in \{u\}\{u\}^c} f_{vw} - \sum_{vw \in \{u\}^c \{u\}} f_{vw} \right) \\
&= \sum_{u \in U} \varphi_u \quad .
\end{aligned}
$$

$\square$

Definition 2. A 5-tuple $N = (V, E, s, t, c)$ is called a **network** iff (V, E) is a directed graph and $s \in V$ and $t \in V$ and $s \neq t$ and $c \in \mathbb{N}^E$.

The nodes s and t are called the **source** and the **sink** of N, respectively.

For any edge $e \in E$, c_e is called the **capacity** of e in N.

Definition 3. A map $f \in \mathbb{N}_0^E$ is called an st-**preflow** in a network $N = (V, E, s, t, c)$ iff

$$\forall e \in E \quad 0 \leq f_e \leq c_e \tag{5}$$

$$\forall v \in V - \{s\} \quad \varphi_v \leq 0 \ . \tag{6}$$

An st-preflow f in N is called an st-**flow** in N iff, in addition,

$$\forall v \in V - \{s, t\} \quad \varphi_v = 0 \ . \tag{7}$$

Definition 4. The instance of the **Maximum st-Flow Problem** wrt. a network $N = (V, E, s, t, c)$ is to

$$\max_{f \in \mathbb{N}_0^E} \quad \sum_{sv \in E} f_{sv} - \sum_{vs \in E} f_{vs} \tag{8}$$

$$\text{subject to} \quad \forall e \in E \quad 0 \leq f_e \leq c_e \tag{9}$$

$$\forall v \in V - \{s, t\} \quad \sum_{vw \in E} f_{vw} = \sum_{uv \in E} f_{uv} \ . \tag{10}$$

Note:

$$\sum_{sv \in E} f_{sv} - \sum_{vs \in E} f_{vs} \quad = \quad \varphi_s$$

Definition 5. For any network $N = (V, E, s, t, c)$ and any st-preflow f in N, the **residual network** of N wrt. f is the network $N' = (V, E', s, t, c')$ such that

$$E' = E^+ \cup E^-$$
$$E^+ = \{vw \in E \mid c_{vw} - f_{vw} > 0\}$$
$$E^- = \{vw \in V^2 \mid wv \in E \wedge f_{wv} > 0\}$$

and

$$\forall vw \in E' \quad c'_{vw} = \begin{cases} c_{vw} - f_{vw} & \text{if } vw \in E^+ \\ f_{wv} & \text{if } vw \in E^- \end{cases} . \tag{11}$$

For any $e \in E'$, c'_e is called the **residual capacity** of e wrt. f.

Any path in (V, E') from s to t (if such a path exists) is called an **augmenting path** of f.

Lemma 2. Let $N = (V, E, s, t, c)$ be a network and f an st-preflow in N. Assume that an $n \in \mathbb{N}$ and an augmenting path $p = (v_1 w_1, \ldots, v_n w_n)$ of f exist.

Let

$$\delta := \min_{vw \in p([n])} c'_{vw} \quad . \tag{12}$$

Then, $f' \in \mathbb{N}_0^E$ such that

$$\forall vw \in E' : \quad f'_{vw} = \begin{cases} f_{vw} + \delta & \text{if } vw \in p([n]) \wedge vw \in E \\ f_{vw} - \delta & \text{if } vw \in p([n]) \wedge wv \in E \\ f_{vw} & \text{otherwise} \end{cases} \tag{13}$$

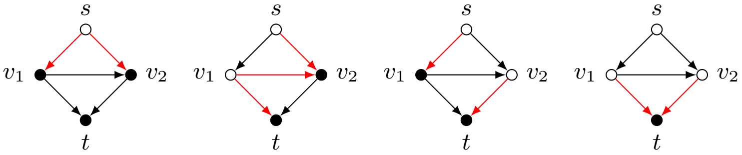
is an st-preflow in N wrt. which

$$\varphi'_s = \varphi_s + \delta \quad . \tag{14}$$

Moreover, if f is an st-flow in N, so is f'.

Definition 6. Let (V, E) be a directed graph. Let $s \in V$ and $t \in V$ and $s \neq t$.

- $X \subseteq V$ is called an st-**cutset** of (V, E) iff $s \in X$ and $t \notin X$.
- $Y \subseteq E$ is called an st-**cut** of (V, E) iff there exists an st-cutset X such that $Y = \{vw \in E | v \in X \wedge w \notin X\}$.



Definition 7. The instance of the **Minimum st-Cut Problem** wrt. a network $N = (V, E, s, t, c)$ is to

$$\min_{x \in \{0,1\}^V} \quad \sum_{vw \in E} x_v(1 - x_w)c_{vw} \tag{15}$$

$$\text{subject to} \quad x_s = 1 \tag{16}$$

$$x_t = 0 \tag{17}$$

Note: With $X := \{v \in V | x_v = 1\}$, we have

$$\sum_{vw \in E} x_v(1 - x_w)c_{vw} = \sum_{vw \in XX^c} c_{vw}$$

Lemma 3. For every network $N = (V, E, s, t, c)$, every st-flow f in N, and every st-cutset $X \subseteq V$,

$$\varphi_s \leq \sum_{vw \in XX^c} c_{vw} \quad . \tag{18}$$

Proof.

$$\begin{aligned} \varphi_s &= \sum_{v \in S} \varphi_v && \text{by (7) and } t \notin S \\ &= \varphi_S && \text{by Lemma 1} \\ &\leq \varphi_S^+ && \text{by (2), (3) and } 0 \leq f \\ &= \sum_{vw \in SS^c} f_{vw} && \text{by (1)} \\ &\leq \sum_{vw \in SS^c} c_{vw} && \text{by (5).} \end{aligned}$$

□

Lemma 3 does **not** hold analogously for every st-preflow, because, wrt. an st-preflow, φ_S need not be an upper bound on φ_s.

Theorem 1. For any network $N = (V, E, s, t, c)$, any $s, t \in V$ such that $s \neq t$, and any st-flow f in N, the following three conditions are equivalent

1. There exists an st-cut whose capacity is equal to φ_s.
2. The st-flow f is optimal, i.e., a solution of (8)–(10).
3. No augmenting path of f exists.

Proof.

(1) implies (2) by virtue of Lemma 3.

(2) implies (3) by virtue of Lemma 2.

We prove that (3) implies (1):

- Let f be an st-flow such that no augmenting path exists.
- Let S be the set of all nodes $v \in V$ such that there exists a path in the residual network wrt. f from s to v. Let S also include s itself.
- Then, $t \notin S$ (otherwise, the path from s to t in the residual network would be an augmenting path).
- Moreover, ...

- Moreover,

$$
\begin{aligned}
\varphi_s &= \sum_{v \in S} \varphi_v && \text{by (7) and } t \notin S \\
&= \varphi_S && \text{by Lemma 1} \\
&= \sum_{vw \in SS^c} f_{vw} \; - \sum_{vw \in S^c S} f_{vw} && \text{by definition of } \varphi_S \\
&= \sum_{vw \in SS^c} c_{vw} && \text{by the arguments below.}
\end{aligned}
$$

- For any $vw \in SS^c$, we have $f_{vw} = c_{vw}$ (otherwise, the contradiction $w \in S$ follows by construction of S and by definition of the residual network).
- For any $vw \in S^c S$, we have $f_{vw} = 0$ (otherwise, the contradiction $v \in S$ follows by construction of S and by definition of the residual network).

□

Algorithm 1. (Ford and Fulkerson, 1956)

Input: Network $N = (V, E, s, t, c)$
Output: $f : E \to \mathbb{N}_0$
for all $vw \in E$
$f_{vw} := 0$
while $\exists n \in \mathbb{N}$ $\exists$augmenting path $p = (v_1 w_1, \ldots, v_n w_n)$ of f
$\delta := \min_{vw \in p([n])} c'_{vw}$
for all $vw \in E$

$$f_{vw} := \begin{cases} f_{vw} + \delta & \text{if } vw \in P \wedge vw \in E \\ f_{vw} - \delta & \text{if } vw \in P \wedge wv \in E \\ f_{vw} & \text{otherwise} \end{cases}$$

Theorem 2. Algorithm 1 terminates. The output f is a maximum st-flow in N.

Proof. Termination.

- For every augmenting path processed, φ_s increases by at least 1.
- Moreover,

$$\varphi_s \leq \sum_{vw \in \{s\}\{s\}^c} c_{vw} \qquad \text{(by Lemma 3)}$$

- Therefore, only finitely many augmenting paths are processed.
- Thus, the algorithm terminates.

Optimality:

- Throughout the execution, f is an st-flow in N.
- When the algorithm terminates, no augmenting path exists.
- Thus, f is a maximum st-flow in N (by Theorem 1).