Computer Vision I

Jannik Presberger, David Stein, Bjoern Andres

Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/25-winter/cv1/

Winter Term 2025/2026

Excursus: Maximum st-flow problem and minimum st-cut problem

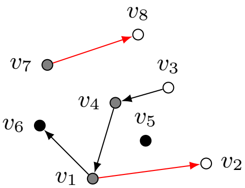
Contents

- Maximum st-flow problem
- Residual capacities and augmenting paths
- Minimum st-cut problem
- Maximum st-flow/minimum st-cut theorem
- Ford-Fulkerson algorithm

Excursus: Maximum st-flow problem and minimum st-cut problem

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

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



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

Excursus: Maximum st-flow problem and minimum st-cut problem

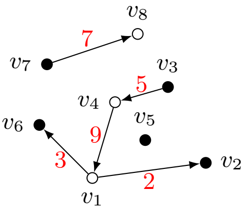
Definition 1. For any directed graph (V, E) and any $f \in \mathbb{N}_0^E$, the maps $\varphi^+, \varphi^-, \varphi : 2^V \to \mathbb{Z}$ such that

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

$$\varphi_U^- = \sum_{vu \in U^cU} f_{vu} \tag{2}$$

$$\varphi_U = \varphi_U^+ - \varphi_U^- \tag{3}$$

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



$U = \{v_1, v_4, v_8\}$

$\varphi_U^+ = 3 + 2$

$\varphi_U^- = 7 + 5$

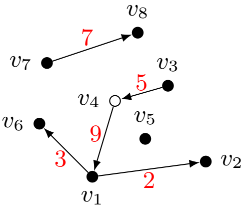
$\varphi_U = -7$

Excursus: Maximum st-flow problem and minimum st-cut problem

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.

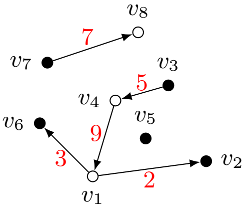


$$\varphi_{v_4}^+ = 9$$
$$\varphi_{v_4}^- = 5$$
$$\varphi_{v_4} = 4$$

Excursus: Maximum st-flow problem and minimum st-cut problem

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.

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

□

Excursus: Maximum st-flow problem and minimum st-cut problem

Definition 2. A 5-tuple $N = (V, E, s, t, c)$ is called a **network** iff (V, E) is a symmetric directed graph and $s \in V$ and $t \in V$ and $s \neq t$ and $c \in \mathbb{N}_0^E$.

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

For any edge e, the number 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

$$\begin{aligned} \forall e \in E\colon &\qquad 0 \leq f_e \leq c_e && (5) \\ \forall v \in V \setminus \{s\}\colon &\qquad \varphi_v \leq 0 \ . && (6) \end{aligned}$$

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

$$\forall v \in V \setminus \{s, t\}\colon \qquad \varphi_v = 0 \ . \qquad (7)$$

Excursus: Maximum st-flow problem and minimum st-cut problem

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

$$\begin{aligned}
&\max_{f \in \mathbb{N}_0^E} && \sum_{sv \in E} f_{sv} - \sum_{vs \in E} f_{vs} && (8)\\
&\text{subject to} && \forall e \in E\colon \quad 0 \leq f_e \leq c_e && (9)\\
& && \forall v \in V \setminus \{s, t\}\colon \quad \sum_{vw \in E} f_{vw} = \sum_{uv \in E} f_{uv} \ . && (10)
\end{aligned}$$

Note:

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

Excursus: Maximum st-flow problem and minimum st-cut problem

Definition 5. For any network $N = (V, E, s, t, c)$ and any st-preflow f in N, let $c' \in \mathbb{N}_0^E$ such that

$$\forall vw \in E\colon \quad c'_{vw} := c_{vw} - f_{vw} + f_{wv} \ . \tag{11}$$

For any edge e of N, the number c'_e is called the **residual capacity** of e wrt. f.

Excursus: Maximum st-flow problem and minimum st-cut problem

Lemma 2. Let $N = (V, E, s, t, c)$ be a network and f an st-preflow in N.

Let $n \in \mathbb{N}$, and let $p = (v_0 w_0, \dots, v_{n-1} w_{n-1})$ be a path in (V, E) from s to t such that $\forall j \in n\colon c'_{v_j w_j} > 0$. (Such a path is called an **augmenting path** of f in N).

Let

$$\delta := \min_{j \in n} \; c'_{v_j w_j} \quad . \tag{12}$$

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

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

is an st-preflow in N wrt. which

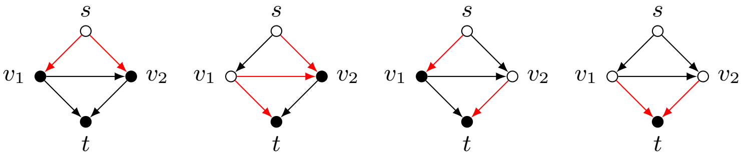
$$\varphi'_s = \varphi_s + \delta \quad . \tag{14}$$

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

Excursus: Maximum st-flow problem and minimum st-cut problem

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 $Y = \{vw \in E | v \in X \wedge w \notin X\}$.



Excursus: Maximum st-flow problem and minimum st-cut problem

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

$$\begin{aligned} \min_{x \in \{0,1\}^V} \quad & \sum_{vw \in E} c_{vw}\, x_v\, (1 - x_w) & (15) \\ \text{subject to} \quad & x_s = 1 & (16) \\ & x_t = 0 & (17) \end{aligned}$$

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

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

Excursus: Maximum st-flow problem and minimum st-cut problem

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.

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

□

Remark 1. 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.

Excursus: Maximum st-flow problem and minimum st-cut problem

Theorem 1. For any network $N = (V, E, s, t, c)$ and any st-flow f in N, the following three properties 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.

Excursus: Maximum st-flow problem and minimum st-cut problem

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 reachable in N from s by a path with positive residual capacities. Let S also include s itself.
- Then, $t \notin S$ (otherwise, an augmenting path would exist).
- Moreover, ...

Excursus: Maximum st-flow problem and minimum st-cut problem

- 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 residual capacities).
- 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 residual capacities).

□

Excursus: Maximum st-flow problem and minimum st-cut problem

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 $p = (v_0w_0, \ldots, v_{n-1}w_{n-1})$ of $f$
    $\delta := \min_{j \in n} c'_{v_j w_j}$
    **for all** $vw \in E$
        $f_{vw} := \begin{cases} f_{vw} + \delta & \text{if } vw \in p(n) \\ f_{vw} & \text{otherwise} \end{cases}$
```

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).