Flows in networks, III

Lecture 27

A function $f : E \to \mathbf{R}$ is called a flow in G if for every vertex $v \in V - s - t$,

$$div_f(v) = \sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) = 0. \tag{1}$$

If $0 \leq f(e) \leq \mathbf{c}(e)$ for every $e \in E$, then the flow is called feasible (for G).

A function $f : E \to \mathbf{R}$ is called a flow in G if for every vertex $v \in V - s - t$,

$$div_f(v) = \sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) = 0. \qquad (1)$$

If $0 \leq f(e) \leq \mathbf{c}(e)$ for every $e \in E$, then the flow is called feasible (for G).

An (s, t)-cut in a network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ is a partition $(S, \overline{S})$ of V into sets S and $\overline{S}$ such that $s \in S$ and $t \in \overline{S}$.

A function $f : E \to \mathbf{R}$ is called a flow in G if for every vertex $v \in V - s - t$,

$$div_f(v) = \sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) = 0. \quad (1)$$

If $0 \leq f(e) \leq \mathbf{c}(e)$ for every $e \in E$, then the flow is called feasible (for G).

An (s,t)-cut in a network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ is a partition $(S, \overline{S})$ of V into sets S and $\overline{S}$ such that $s \in S$ and $t \in \overline{S}$.

The capacity of $(S, \overline{S})$ is

$$\mathbf{c}(S, \overline{S}) = \sum_{xy \in E : x \in S, y \in \overline{S}} \mathbf{c}(xy). \quad (2)$$

Important inequality

Claim 4.17. For every feasible flow f in in a network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ and every s, t-cut $(S, \overline{S})$ of V,

$$M(f) \leq \mathbf{c}(S, \overline{S}). \tag{3}$$

Important inequality

Claim 4.17. For every feasible flow f in in a network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ and every s, t-cut $(S, \overline{S})$ of V,

$$M(f) \leq \mathbf{c}(S, \overline{S}). \tag{3}$$

Proof. Consider $F(f, S) = \sum_{v \in S} div_f(v)$.
By the definition of a flow, since $s \in S$ and $t \notin S$,
$F(f, S) = div_f(s) = M(f)$.

Important inequality

Claim 4.17. For every feasible flow f in in a network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ and every s, t-cut $(S, \overline{S})$ of V,

$$M(f) \leq \mathbf{c}(S, \overline{S}). \tag{3}$$

Proof. Consider $F(f, S) = \sum_{v \in S} div_f(v)$.
By the definition of a flow, since $s \in S$ and $t \notin S$, $F(f, S) = div_f(s) = M(f)$.

On the other hand, by (1),

$$F(f, S) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right).$$

So,

$$F(f, S) = \sum_{v \in S} \left(\sum_{e \in E^{+}(v)} f(e) - \sum_{e \in E^{-}(v)} f(e) \right) . \tag{4}$$

So,

$$F(f, S) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right) . \tag{4}$$

Let $e = xy$. Then e contributes $f(e)$ to $div_f(x)$ and $-f(e)$ to $div_f(y)$.
So, if $\{x, y\} \subset S$, then the net contribution of e is $f(e) - f(e) = 0$.
Also, if $\{x, y\} \subset \overline{S}$, then the net contribution of e is 0.
If $x \in S$ and $y \in \overline{S}$, then e contributes $f(e)$ into the RHS of (4).
Finally, if $y \in S$ and $x \in \overline{S}$, then e contributes $-f(e)$.

So,

$$F(f,S) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right). \qquad (4)$$

Let $e = xy$. Then e contributes $f(e)$ to $div_f(x)$ and $-f(e)$ to $div_f(y)$.
So, if $\{x, y\} \subset S$, then the net contribution of e is $f(e) - f(e) = 0$.
Also, if $\{x, y\} \subset \overline{S}$, then the net contribution of e is 0.
If $x \in S$ and $y \in \overline{S}$, then e contributes $f(e)$ into the RHS of (4).
Finally, if $y \in S$ and $x \in \overline{S}$, then e contributes $-f(e)$.

Hence $F(f,S) = \sum_{xy \in E: x \in S, y \in \overline{S}} f(xy) - \sum_{yx \in E: x \in S, y \in \overline{S}} f(yx)$.

So,

$$F(f,S) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right). \qquad (4)$$

Let $e = xy$. Then e contributes $f(e)$ to $div_f(x)$ and $-f(e)$ to $div_f(y)$.
So, if $\{x,y\} \subset S$, then the net contribution of e is $f(e) - f(e) = 0$.
Also, if $\{x,y\} \subset \overline{S}$, then the net contribution of e is 0.
If $x \in S$ and $y \in \overline{S}$, then e contributes $f(e)$ into the RHS of (4).
Finally, if $y \in S$ and $x \in \overline{S}$, then e contributes $-f(e)$.

Hence $F(f,S) = \sum_{xy \in E: x \in S, y \in \overline{S}} f(xy) - \sum_{yx \in E: x \in S, y \in \overline{S}} f(yx)$.

Since $0 \leq f(e) \leq \mathbf{c}(e)$ for each $e \in E$, the last sum is at most $\sum_{xy \in E: x \in S, y \in \overline{S}} \mathbf{c}(xy) = \mathbf{c}(S, \overline{S})$.

So,

$$F(f,S) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right). \qquad (4)$$

Let $e = xy$. Then e contributes $f(e)$ to $div_f(x)$ and $-f(e)$ to $div_f(y)$.
So, if $\{x,y\} \subset S$, then the net contribution of e is $f(e) - f(e) = 0$.
Also, if $\{x,y\} \subset \overline{S}$, then the net contribution of e is 0.
If $x \in S$ and $y \in \overline{S}$, then e contributes $f(e)$ into the RHS of (4).
Finally, if $y \in S$ and $x \in \overline{S}$, then e contributes $-f(e)$.

Hence $F(f,S) = \sum_{xy \in E: x \in S, y \in \overline{S}} f(xy) - \sum_{yx \in E: x \in S, y \in \overline{S}} f(yx)$.

Since $0 \le f(e) \le \mathbf{c}(e)$ for each $e \in E$, the last sum is at most $\sum_{xy \in E: x \in S, y \in \overline{S}} \mathbf{c}(xy) = \mathbf{c}(S, \overline{S})$. So, $M(f) \le \mathbf{c}(S, \overline{S})$.

Max-flow min-cut Theorem

Theorem 4.18. For every network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$, maximum of $M(f)$ over all feasible flows equals to the minimum of $\mathbf{c}(S, \overline{S})$ over all s, t-cuts $(S, \overline{S})$.

Max-flow min-cut Theorem

Theorem 4.18. For every network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$, maximum of $M(f)$ over all feasible flows equals to the minimum of $\mathbf{c}(S, \overline{S})$ over all s, t-cuts $(S, \overline{S})$.

An idea for Ford-Fulkerson Algorithm:
1. Finding the maximum $M(f)$ for a feasible flow in a network G is finding the maximum of a linear function $div_f(s)$ on the closed bounded set formed by the inequalities $0 \leq f(e) \leq \mathbf{c}(e)$ for every $e \in E$ and equations $div_f(v) = 0$ for each $v \in V - s - t$. Since each continuous function on a compact set achieves its maximum, the maximum flow does exist.

Max-flow min-cut Theorem

Theorem 4.18. For every network $G = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$, maximum of $M(f)$ over all feasible flows equals to the minimum of $\mathbf{c}(S, \overline{S})$ over all s, t-cuts $(S, \overline{S})$.

An idea for Ford-Fulkerson Algorithm:
1. Finding the maximum $M(f)$ for a feasible flow in a network G is finding the maximum of a linear function $div_f(s)$ on the closed bounded set formed by the inequalities $0 \leq f(e) \leq \mathbf{c}(e)$ for every $e \in E$ and equations $div_f(v) = 0$ for each $v \in V - s - t$. Since each continuous function on a compact set achieves its maximum, the maximum flow does exist.

2. Let f be a feasible flow in G with the maximum $M(f)$. By Theorem 4.17, $f = \sum_{i=1}^{k} f_i$, where each f_i is either
(a) a flow along a cycle, or
(b) a flow along a t, s-path, or
(c) a flow along an s, t-path.

An idea for Ford-Fulkerson Algorithm

1. The maximum flow does exist.

2. Let f be a feasible flow in G with the maximum $M(f)$. By Theorem 4.16, $f = \sum_{i=1}^{k} f_i$, where each f_i is either
(a) a flow along a cycle, or
(b) a flow along a t, s-path, or
(c) a flow along an s, t-path.

If some f_{i_0} satisfies (a), then $f - f_{i_0}$ has the same value $M(f - f_{i_0})$ and is feasible.
If some f_{i_0} satisfies (b), then $f - f_{i_0}$ has even larger $M(f - f_{i_0})$ and is feasible.

An idea for Ford-Fulkerson Algorithm

1. The maximum flow does exist.

2. Let f be a feasible flow in G with the maximum $M(f)$. By Theorem 4.16, $f = \sum_{i=1}^{k} f_i$, where each f_i is either
(a) a flow along a cycle, or
(b) a flow along a t, s-path, or
(c) a flow along an s, t-path.

If some f_{i_0} satisfies (a), then $f - f_{i_0}$ has the same value $M(f - f_{i_0})$ and is feasible.
If some f_{i_0} satisfies (b), then $f - f_{i_0}$ has even larger $M(f - f_{i_0})$ and is feasible.

Thus, it is enough to seek f in the form $f = \sum_{i=1}^{k} f_i$, where each f_i is a flow along an s, t-path.

Ford-Fulkerson Algorithm with an example

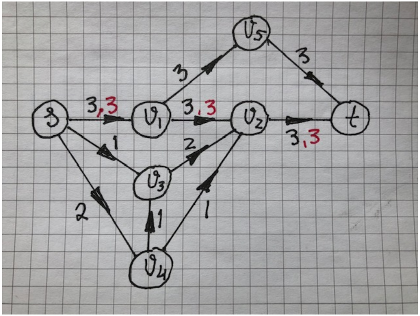

Figure: Network and a starting flow f_1; $M(f_1) = 3$.

Given a feasible flow f in a network G, the **residue network** $G(f)$ has $V(G(f)) = V$ and $xy \in E(G(f))$ iff

either $xy \in E$ and $f(xy) < \mathbf{c}(xy)$ (then $\mathbf{c_f}(xy) = \mathbf{c}(xy) - f(xy)$),

or $yx \in E$ and $f(yx) > 0$ (in this case, $\mathbf{c_f}(xy) = f(yx)$).

Ford-Fulkerson Algorithm with an example, 2

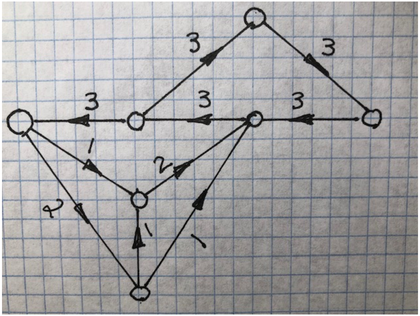

Figure: First residue network G_{f_1}.

If G_f has an s, t-path P and $\rho = \min\{\mathbf{c_f}(xy) : xy \in E(P)\}$, then instead of f, consider $f' = f + \phi(P, \rho)$ and repeat. Note $M(f') = M(f) + \rho > M(f)$.

Ford-Fulkerson Algorithm with an example, 2

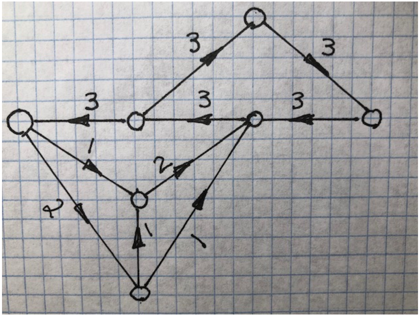

Figure: First residue network G_{f_1}.

If G_f has an s, t-path P and $\rho = \min\{\mathbf{c_f}(xy) : xy \in E(P)\}$, then instead of f, consider $f' = f + \phi(P, \rho)$ and repeat. Note $M(f') = M(f) + \rho > M(f)$.

Here we have e.g. $P_1 = sv_4v_2v_1v_5t$ and $\rho_1 = \min\{2, 1, 3, 3, 3\} = 1$. So $f_2 = f_1 + \phi(P_1, 1)$.

Second iteration

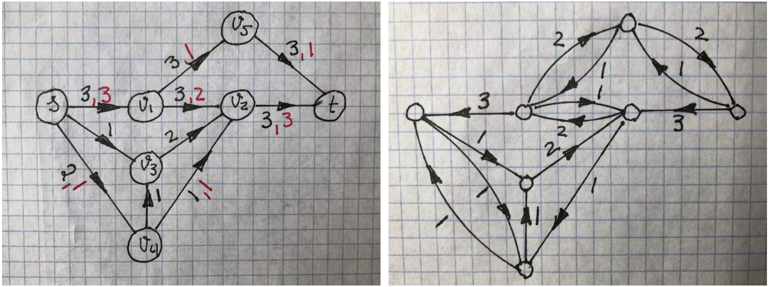

Figure: The second flow f_2 and the second residue network G_{f_2}.

We have $M(f_2) = 4$. The residue network has an s, t-path $P_2 = sv_3v_2v_1v_5t$ and $\rho_2 = \min\{1, 2, 2, 2, 2\} = 1$.

Hence $f_3 = f_2 + \phi(P_2, 1)$.

Third iteration

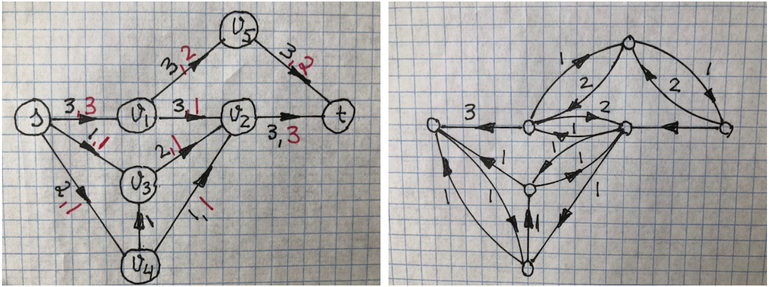

Figure: The third flow f_3 and the third residue network G_{f_3}.

We have $M(f_3) = 5$. The residue network has an s,t-path $P_3 = sv_4v_3v_2v_1v_5t$ and $\rho_3 = \min\{1,1,1,1,1\} = 1$.

Hence $f_4 = f_3 + \phi(P_3, 1)$.

Last iteration

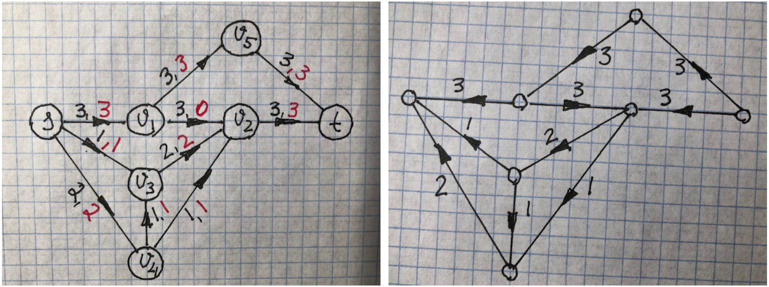

Figure: The fourth flow f_4 and the fourth residue network G_{f_4}.

We have $M(f_4) = 6$. The residue network G_{f_4} has no s, t-paths. Hence, there is a set S, namely $S = \{s\}$, such that $s \in S$, $t \notin S$ and G_{f_4} has no edges from S to $\overline{S}$.

Last iteration

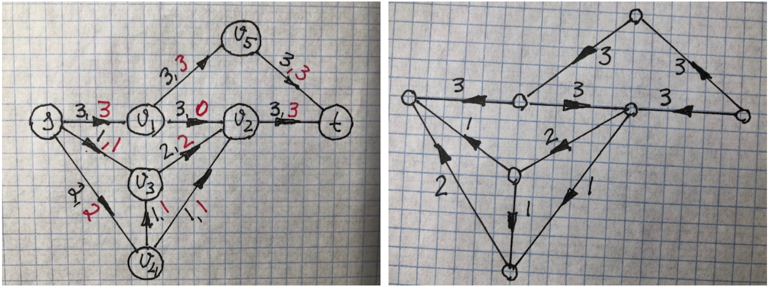

Figure: The fourth flow f_4 and the fourth residue network G_{f_4}.

We have $M(f_4) = 6$. The residue network G_{f_4} has no s, t-paths.
Hence, there is a set S, namely $S = \{s\}$, such that
$s \in S$, $t \notin S$ and G_{f_4} has no edges from S to $\overline{S}$.

We check that $\mathbf{c}(S, \overline{S}) = 6 = M(f_4)$.
This certifies that both, f_4 and $(S, \overline{S})$ are optimal.