Flows in networks, IV

Lecture 28

Earlier, we proved:

Theorem 4.16. Every positive flow f in a network G can be represented as the sum of at most $|E(G)|$ positive flows along cycles, along s,t-paths and along t,s-paths.

The capacity of an s,t-cut $(S,\overline{S})$ is

$$\mathbf{c}(S,\overline{S}) = \sum_{xy\in E: x\in S, y\in\overline{S}} \mathbf{c}(xy). \tag{1}$$

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,

$$M(f) \leq \mathbf{c}(S,\overline{S}). \tag{2}$$

Proof of 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 $(S, \overline{S})$.

Proof of 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 $(S, \overline{S})$.

We know that G has a maximum flow, say f. Construct the residue network G_f.

Proof of 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 $(S, \overline{S})$.

We know that G has a maximum flow, say f. Construct the residue network G_f.

If G_f has an s, t-path P and $\rho = \min\{\mathbf{c_f}(xy) : xy \in E(P)\}$, then the flow $f' = f + \phi(P, \rho)$ has $M(f') = M(f) + \rho > M(f)$, a contradiction.

Proof of 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 $(S, \overline{S})$.

We know that G has a maximum flow, say f. Construct the residue network G_f.

If G_f has an s, t-path P and $\rho = \min\{\mathbf{c_f}(xy) : xy \in E(P)\}$, then the flow $f' = f + \phi(P, \rho)$ has $M(f') = M(f) + \rho > M(f)$, a contradiction.

Let $S = \{v \in V : G_f \text{ has an } s, v\text{-path}\}$. Trivially, $s \in S$. By above, $t \notin S$. Also

$$G_f \text{ has no edges } xy \text{ with } x \in S \text{ and } y \in \overline{S}. \qquad (3)$$

By (3), (a) if $xy \in E$, $x \in S$ and $y \in \overline{S}$, then $f(xy) = \mathbf{c}(xy)$,

(b) if $yx \in E$, $x \in S$ and $y \in \overline{S}$, then $f(yx) = 0$.

Recall: (a) if $xy \in E$, $x \in S$ and $y \in \overline{S}$, then $f(xy) = \mathbf{c}(xy)$,

(b) if $yx \in E$, $x \in S$ and $y \in \overline{S}$, then $f(yx) = 0$.

Recall: (a) if $xy \in E$, $x \in S$ and $y \in \overline{S}$, then $f(xy) = \mathbf{c}(xy)$,

(b) if $yx \in E$, $x \in S$ and $y \in \overline{S}$, then $f(yx) = 0$.

Hence repeating the calculations in the proof of Claim 4.17 and using (a) and (b) in the third line below,

$$M(f) = \sum_{v \in S} div_f(v) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right)$$

Recall: (a) if $xy \in E$, $x \in S$ and $y \in \overline{S}$, then $f(xy) = \mathbf{c}(xy)$,

(b) if $yx \in E$, $x \in S$ and $y \in \overline{S}$, then $f(yx) = 0$.

Hence repeating the calculations in the proof of Claim 4.17 and using (a) and (b) in the third line below,

$$M(f) = \sum_{v \in S} div_f(v) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right)$$

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

Recall: (a) if $xy \in E$, $x \in S$ and $y \in \overline{S}$, then $f(xy) = \mathbf{c}(xy)$,

(b) if $yx \in E$, $x \in S$ and $y \in \overline{S}$, then $f(yx) = 0$.

Hence repeating the calculations in the proof of Claim 4.17 and using (a) and (b) in the third line below,

$$M(f) = \sum_{v \in S} div_f(v) = \sum_{v \in S} \left(\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) \right)$$

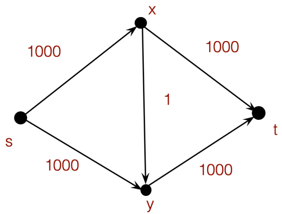
$$= \sum_{xy \in E: x \in S, y \in \overline{S}} f(xy) - \sum_{yx \in E: x \in S, y \in \overline{S}} f(yx)$$

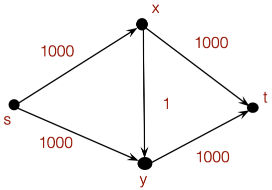
$$\sum_{xy \in E: x \in S, y \in \overline{S}} \mathbf{c}(xy) - \sum_{yx \in E: x \in S, y \in \overline{S}} 0 = \mathbf{c}(S, \overline{S}).$$

This proves Max-flow Min-cut Theorem.

For each network G, let $M(G)$ denote the maximum value of a feasible flow in G.

Consider the following simple example

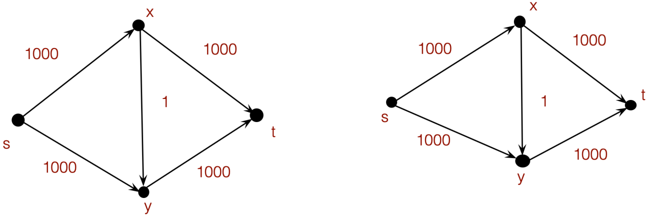




Here on the left is the original network, on the right is the residue network.

For each network G, let $M(G)$ denote the maximum value of a feasible flow in G.

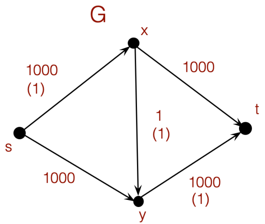
Consider the following simple example

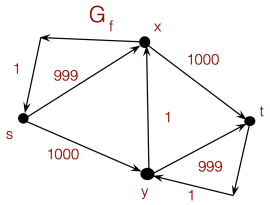


Here on the left is the original network, on the right is the residue network.

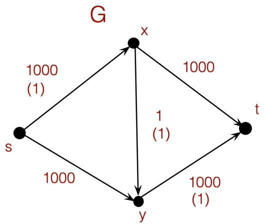
Consider the s, t-path s, x, y, t in G_f. The minimum capacity of its edges is 1. Take an iteration of FF-algorithm.

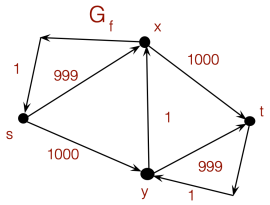
We now have



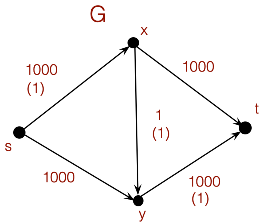


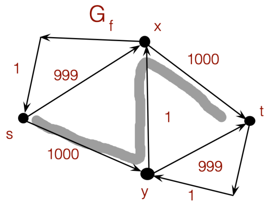
We now have



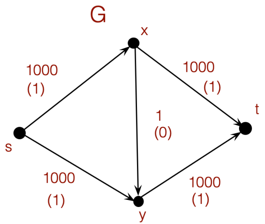


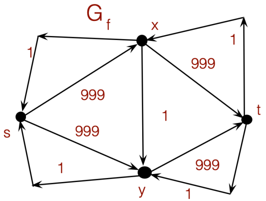
We have the s, t-path s, y, x, t in G_f. Again, the minimum capacity of its edges is 1.



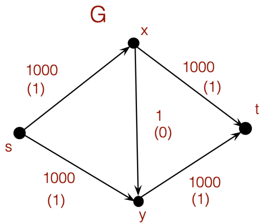


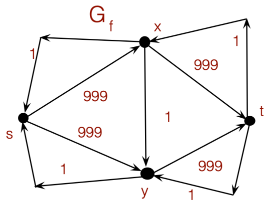
This time we have



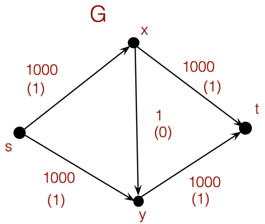


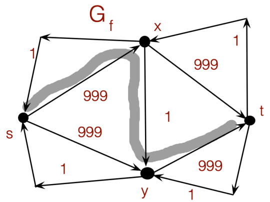
This time we have





Again, we have the s, t-path s, x, y, t in G_f.





Altogether, we will do 2000 steps in this tiny network.
Moreover, there is an example of a network with 10 vertices and irrational capacities, where if we choose the augmenting paths not wisely then we will need infinite number of iterations, and moreover, the value of the flow will not tend to the maximum flow, but only to one fourth of it.

Altogether, we will do 2000 steps in this tiny network. Moreover, there is an example of a network with 10 vertices and irrational capacities, where if we choose the augmenting paths not wisely then we will need infinite number of iterations, and moreover, the value of the flow will not tend to the maximum flow, but only to one fourth of it.

But there is a good news: Edmonds and Karp proved that if we use the Breadth-First Search for finding augmenting paths in G_f, then independently of capacities we find a maximum flow after at most n^3 iterations.

Altogether, we will do 2000 steps in this tiny network. Moreover, there is an example of a network with 10 vertices and irrational capacities, where if we choose the augmenting paths not wisely then we will need infinite number of iterations, and moreover, the value of the flow will not tend to the maximum flow, but only to one fourth of it.

But there is a good news: Edmonds and Karp proved that if we use the Breadth-First Search for finding augmenting paths in G_f, then independently of capacities we find a maximum flow after at most n^3 iterations.

Observation: If all capacities are integers, then the FF-algorithm yields an integer-valued flow.

Proof of Theorem 4.13

It says: $\kappa'_G(x, y) = \lambda'_G(x, y)$ $\forall x, y \in V(G)$ $\forall$ digraph G.

Proof of Theorem 4.13

It says: $\kappa'_G(x, y) = \lambda'_G(x, y)$ $\forall x, y \in V(G)$ $\forall$ digraph G.

Proof. Construct network H with $V(H) = V(G)$, $E(H) = E(G)$, $s = x$, $t = y$ and $\mathbf{c}(e) = 1$ for each $e \in E(H)$. Let $(S_0, \overline{S_0})$ be an x, y-cut of minimum capacity. By Max-flow Min-cut Theorem,

$$M(H) = \mathbf{c}(S_0, \overline{S_0}). \qquad (4)$$

Proof of Theorem 4.13

It says: $\kappa'_G(x, y) = \lambda'_G(x, y)$ $\forall x, y \in V(G)$ $\forall$ digraph G.

Proof. Construct network H with $V(H) = V(G)$, $E(H) = E(G)$, $s = x$, $t = y$ and $\mathbf{c}(e) = 1$ for each $e \in E(H)$. Let $(S_0, \overline{S_0})$ be an x, y-cut of minimum capacity. By Max-flow Min-cut Theorem,

$$M(H) = \mathbf{c}(S_0, \overline{S_0}). \quad (4)$$

Since $\mathbf{c}(e) = 1$ for each $e \in E(H)$, $\min \mathbf{c}(S, \overline{S}) = \kappa'_G(x, y)$. Let f_0 be a flow in H with $M(f_0) = M(H)$. By Theorem 4.16, $f_0 = \sum_{i=1}^{k} \phi(P_i, \rho_i)$, where each ρ_i is a positive integer. Hence $\rho_1 = \ldots = \rho_k = 1$. This implies that all P_i are edge-disjoint, and $M(f_0) = k$.

Proof of Theorem 4.13

It says: $\kappa'_G(x,y) = \lambda'_G(x,y)$ $\forall x, y \in V(G)$ $\forall$ digraph G.

Proof. Construct network H with $V(H) = V(G)$, $E(H) = E(G)$, $s = x$, $t = y$ and $\mathbf{c}(e) = 1$ for each $e \in E(H)$. Let $(S_0, \overline{S_0})$ be an x, y-cut of minimum capacity. By Max-flow Min-cut Theorem,

$$M(H) = \mathbf{c}(S_0, \overline{S_0}). \tag{4}$$

Since $\mathbf{c}(e) = 1$ for each $e \in E(H)$, $\min \mathbf{c}(S, \overline{S}) = \kappa'_G(x,y)$. Let f_0 be a flow in H with $M(f_0) = M(H)$. By Theorem 4.16, $f_0 = \sum_{i=1}^{k} \phi(P_i, \rho_i)$, where each ρ_i is a positive integer. Hence $\rho_1 = \ldots = \rho_k = 1$. This implies that all P_i are edge-disjoint, and $M(f_0) = k$.

Thus $M(f_0) \leq \lambda'_G(x,y)$. And since $\lambda'_G(x,y) \geq M(H)$, we get from (4) that

$$\lambda'_G(x,y) \geq M(H) = \mathbf{c}(S_0, \overline{S_0}) \geq \kappa'_G(x,y) \geq \lambda'_G(x,y).$$

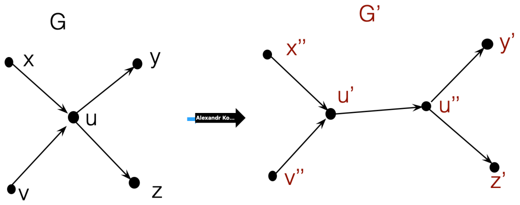
Proof of Theorem 4.12

It says: $\kappa_G(x, y) = \lambda_G(x, y)$ $\forall x, y \in V(G)$ with $xy \notin E(G)$ $\forall$ digraph G.

Proof of Theorem 4.12

It says: $\kappa_G(x, y) = \lambda_G(x, y)$ $\forall x, y \in V(G)$ with $xy \notin E(G)$ $\forall$ digraph G.

Proof. Let G a digraph. Construct another digraph G' as follows



Replace each vertex u by two vertices u' and u'' with an edge $u'u''$, and replace each edge vw with edge $v''w'$.

By Theorem 4.13, $\kappa'_{G'}(x'', y') = \lambda'_{G'}(x'', y')$. (*)

By Theorem 4.13, $\kappa'_{G'}(x'', y') = \lambda'_{G'}(x'', y')$. $(*)$

Let S be a minimum x, y-cut in G. If we delete in G' edge $w'w''$ for every $w \in S$, then the resulting subgraph of G' has no x'', y'-path. Hence

$$\kappa_G(x, y) \geq \kappa'_{G'}(x'', y'). \quad (5)$$

By Theorem 4.13, $\kappa'_{G'}(x'', y') = \lambda'_{G'}(x'', y')$. $(*)$

Let S be a minimum x, y-cut in G. If we delete in G' edge $w'w''$ for every $w \in S$, then the resulting subgraph of G' has no x'', y'-path. Hence

$$\kappa_G(x, y) \geq \kappa'_{G'}(x'', y'). \tag{5}$$

On the other hand, let L be a minimum x'', y'-edge-cut in G'. If L contains an edge of the form $v''w'$ and $w' \neq y'$, then we can replace it in L by edge $w'w''$. Similarly, if $v'' \neq x''$, then we can replace $v''w'$ in L by edge $v'v''$. Since $x''y' \notin E(G')$, we can find a minimum x'', y'-edge-cut L in G' in which each edge has the form $u'u''$. But then the set $\{u \in V(G) : u'u'' \in L\}$ is an x, y-cut in G. So, $\kappa_G(x, y) \leq \kappa'_{G'}(x'', y')$, and together with (5),

$$\kappa_G(x, y) = \kappa'_{G'}(x'', y'). \tag{6}$$

Any two int.-disjoint x, y-paths in G yield edge-disjoint x'', y'-paths in G' (with added edges of the kind $u'u''$). Hence $\lambda_G(x, y) \leq \lambda'_{G'}(x'', y')$.

Any two int.-disjoint x, y-paths in G yield edge-disjoint x'', y'-paths in G' (with added edges of the kind $u'u''$). Hence $\lambda_G(x, y) \leq \lambda'_{G'}(x'', y')$.

Any two edge-disjoint x'', y'-paths in G' are also vertex int.-disjoint, and hence correspond to int.-disjoint x, y-paths in G. Hence $\lambda_G(x, y) \geq \lambda'_{G'}(x'', y')$, which together with the previous para yields

$$\lambda_G(x, y) = \lambda'_{G'}(x'', y'). \qquad (7)$$

Any two int.-disjoint x, y-paths in G yield edge-disjoint x'', y'-paths in G' (with added edges of the kind $u'u''$). Hence $\lambda_G(x, y) \leq \lambda'_{G'}(x'', y')$.

Any two edge-disjoint x'', y'-paths in G' are also vertex int.-disjoint, and hence correspond to int.-disjoint x, y-paths in G. Hence $\lambda_G(x, y) \geq \lambda'_{G'}(x'', y')$, which together with the previous para yields

$$\lambda_G(x, y) = \lambda'_{G'}(x'', y'). \tag{7}$$

Now $(*)$, (6), and (7) together imply the theorem.

Any two int.-disjoint x, y-paths in G yield edge-disjoint x'', y'-paths in G' (with added edges of the kind $u'u''$). Hence $\lambda_G(x, y) \leq \lambda'_{G'}(x'', y')$.

Any two edge-disjoint x'', y'-paths in G' are also vertex int.-disjoint, and hence correspond to int.-disjoint x, y-paths in G. Hence $\lambda_G(x, y) \geq \lambda'_{G'}(x'', y')$, which together with the previous para yields

$$\lambda_G(x, y) = \lambda'_{G'}(x'', y'). \tag{7}$$

Now $(*)$, (6), and (7) together imply the theorem.

Remark: Since we can find maximum flows in n-vertex networks in $O(n^3)$ iterations, the last proofs yield polynomial algorithms for finding connectivity, edge connectivity and minimum separating sets in directed graphs.