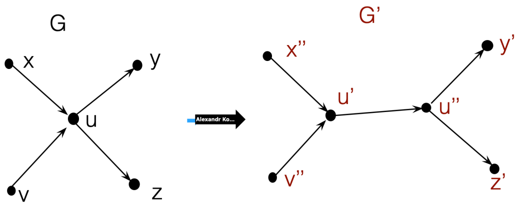
Flows: algorightms and applications

Lecture 29

Theorem 4.13: $\kappa'_G(x,y) = \lambda'_G(x,y)$ $\forall x, y \in V(G)$ $\forall$ digraph G.

Theorem 4.12: $\kappa_G(x,y) = \lambda_G(x,y)$ $\forall x, y \in V(G)$ 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

$$\kappa_G(x, y) \geq \kappa'_{G'}(x'', y'). \quad (1)$$

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

$$\kappa_G(x, y) \geq \kappa'_{G'}(x'', y'). \qquad (1)$$

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 (1),

$$\kappa_G(x, y) = \kappa'_{G'}(x'', y'). \qquad (2)$$

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 $\lambda_G(x, y) \le \lambda'_{G'}(x'', y')$.

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

$$\lambda_G(x, y) = \lambda'_{G'}(x'', y'). \tag{3}$$

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

$$\lambda_G(x,y) = \lambda'_{G'}(x'',y'). \qquad (3)$$

Now $(*)$, (2), and (3) 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

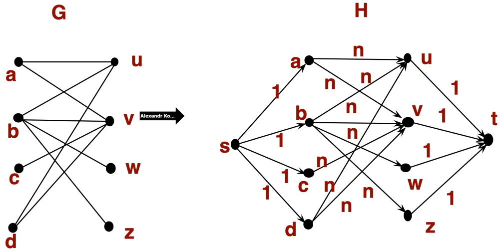
$$\lambda_G(x, y) = \lambda'_{G'}(x'', y'). \tag{3}$$

Now $(*)$, (2), and (3) 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.

Matchings in bipartite graphs: using flows

Let G be a bipartite graph with parts X and Y. Construct an auxiliary network $H = (V, E, s, t, \{\mathbf{c}(e)\}_{e \in E})$ as follows.



We take $V = V(G) \cup \{s, t\}$, orient each edge of G from X to Y and make the capacity of each such edge equal $n = |X| + |Y|$, add the set of edges $\{sx : x \in X\} \cup \{yt : y \in Y\}$, each of capacity 1.

Theorem 4.19: For each bipartite graph G, $\alpha'(G) = M(H)$.

Theorem 4.19: For each bipartite graph G, $\alpha'(G) = M(H)$.

Proof. Let $L = \{x_i y_i : 1 \leq i \leq k\}$ be a matching in G with $|L| = k = \alpha'(G)$. For $i = 1, \ldots, k$, let P_i denote the path s, x_i, y_i, t in H. Then the value of the flow

$$\sum_{i=1}^{k} \phi(P_i, 1)$$

is $k = \alpha'(G)$. This proves $\alpha'(G) \leq M(H)$.

Theorem 4.19: For each bipartite graph G, $\alpha'(G) = M(H)$.

Proof. Let $L = \{x_i y_i : 1 \leq i \leq k\}$ be a matching in G with $|L| = k = \alpha'(G)$. For $i = 1, \ldots, k$, let P_i denote the path s, x_i, y_i, t in H. Then the value of the flow

$$\sum_{i=1}^{k} \phi(P_i, 1)$$

is $k = \alpha'(G)$. This proves $\alpha'(G) \leq M(H)$.

Now consider a flow f in H with $M(f) = M(H)$ obtained using FF-algorithm. By Theorem 4.16, $f = \sum_{i=1}^{k} \phi(P_i, \rho_i)$, where each ρ_i is a positive integer. Since each of these paths contains two edges of capacity 1, $\rho_1 = \ldots = \rho_k = 1$.

Theorem 4.19: For each bipartite graph G, $\alpha'(G) = M(H)$.

Proof. Let $L = \{x_i y_i : 1 \le i \le k\}$ be a matching in G with $|L| = k = \alpha'(G)$. For $i = 1, \ldots, k$, let P_i denote the path s, x_i, y_i, t in H. Then the value of the flow

$$\sum_{i=1}^{k} \phi(P_i, 1)$$

is $k = \alpha'(G)$. This proves $\alpha'(G) \le M(H)$.

Now consider a flow f in H with $M(f) = M(H)$ obtained using FF-algorithm. By Theorem 4.16, $f = \sum_{i=1}^{k} \phi(P_i, \rho_i)$, where each ρ_i is a positive integer. Since each of these paths contains two edges of capacity 1, $\rho_1 = \ldots = \rho_k = 1$.

Moreover, since each $x \in X$ has only one in-neighbor, and each $y \in Y$ has only one out-neighbor, each of the paths P_i has the form s, x_i, y_i, t and all edges $x_i y_i$ are disjoint. This proves $\alpha'(G) \ge M(H)$ and hence the theorem.

Main results in Chapter 4

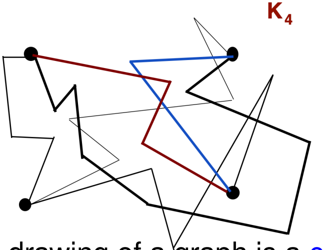
1. Characterization theorem for 2-connected graphs. (Theorem 4.6) (Theorem 4.2.4 in the book).
2. Max-flow Min-cut Theorem (Theorem 4.18).

Main results in Chapter 4

1. Characterization theorem for 2-connected graphs. (Theorem 4.6) (Theorem 4.2.4 in the book).
2. Max-flow Min-cut Theorem (Theorem 4.18).
3. Menger Theorems (Theorems 4.8, 4.10, 4.11, 4.12 and 4.13)

A polygonal curve is a curve composed of finitely many line segments.

A drawing of a graph G is a function $\varphi : V(G) \cup E(G) \to \mathbf{R}^2$ s.t.
(a) $\varphi(v) \in \mathbf{R}^2$ for every $v \in V(G)$;
(b) $\varphi(v) \neq \varphi(v')$ if $v, v' \in V(G)$ and $v \neq v'$;
(c) $\varphi(xy)$ is a polygonal curve connecting $\varphi(x)$ with $\varphi(y)$.



A crossing in a drawing of a graph is a common point in the images of two edges that is not the image of their common end.

A graph G is planar if it has a drawing φ without crossings.

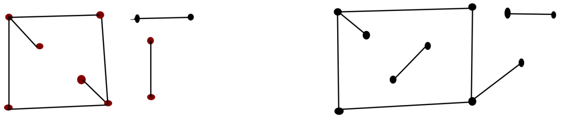
A graph G is planar if it has a drawing φ without crossings.

A plane graph is a pair (G, φ) where φ is a drawing of G without crossings.

A graph G is planar if it has a drawing φ without crossings.

A plane graph is a pair (G, φ) where φ is a drawing of G without crossings.

Two distinct plane graphs.



Remind me about Gas-Water-Electricity Problem.

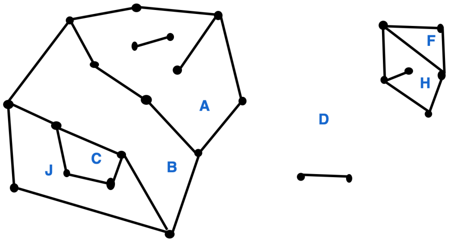
A **face** of a plane graph (G, φ) is a connected component of $\mathbf{R^2} - \varphi(V(G) \cup E(G))$.

A **face** of a plane graph (G, φ) is a connected component of $\mathbf{R^2} - \varphi(V(G) \cup E(G))$.

The length, $\ell(F_i)$, of a face F_i in a plane graph (G, φ) is the total length of the closed walk(s) bounding F_i.

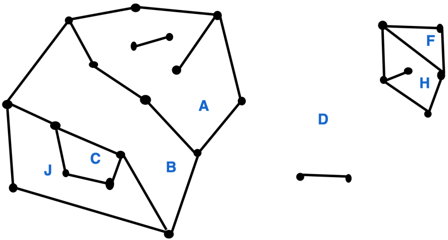
A **face** of a plane graph (G, φ) is a connected component of $\mathbf{R}^2 - \varphi(V(G) \cup E(G))$.

The length, $\ell(F_i)$, of a face F_i in a plane graph (G, φ) is the total length of the closed walk(s) bounding F_i.



A **face** of a plane graph (G, φ) is a connected component of $\mathbf{R}^2 - \varphi(V(G) \cup E(G))$.

The length, $\ell(F_i)$, of a face F_i in a plane graph (G, φ) is the total length of the closed walk(s) bounding F_i.



Definition of dual graphs: given in class (and book).