**Math 4710/6710 – Graph Theory – Fall 2019**

**Basic concepts II**

Note: Many definitions are different from those in the book.

**Trees**

**Edge exchange properties:** Let $T, U$ be distinct spanning trees of a graph $G$, and $e \in E(T) - E(U)$.
(EE1) There is $e' \in E(U) - E(T)$ such that $T - e + e'$ is a spanning tree.
(EE2) There is $e'' \in E(U) - E(T)$ such that $U + e - e''$ is a spanning tree.

**Kruskal's Algorithm:** Apply Global TCM, being greedy, i.e., picking an available edge of minimum weight at each step.

**Jarník-Prim Algorithm:** Apply Local TCM, being greedy, i.e., picking an available edge of minimum weight at each step.

**Directed graphs**

*directed graph* or *digraph*: $D$ has vertex set $V(D)$, set of *arcs/* or *directed edges* $A(D)$, incidence function $\psi_D$ mapping each arc to *ordered* pair of vertices.
*strict* digraph: no loops or parallel arcs (but opposite arcs are allowed); denote arc as $uv$.
Arc from $u$ to $v$: *head* $v$, *tail* $u$, $u$ *dominates* $v$.
*outdegree* $d^+(v)$, *indegree* $d^-(v)$.
Set of *outneighbours* $N^+(v) = \{u \in V(D) \mid u \neq v,\ v \text{ dominates } u\}$; *inneighbours* $N^-(v)$.
*underlying graph*: ignore directions.
*associated digraph* of graph $G$: replace each edge by pair of opposite arcs.
*orientation* of graph $G$: replace each edge by *one* of possible arcs; *oriented graph* = orientation of simple graph.
*tournament*: orientation of complete graph $K_n$.
*source*: vertex of indegree 0; *sink*: vertex of outdegree 0.
*converse* of $D$: reverse all arcs.

**Moving around in digraphs:** Have directed versions of walks, trails, paths, cycles, euler trails and euler tours: must follow edges in correct direction. Directed $uv$-walk goes *from* $u$ *to* $v$.

*connected*: underlying graph connected.
If $X, Y \subseteq V(D)$, $A(X, Y)$ = edges with tail in $X$, head in $Y$. Let $\overline{X}$ denote $V(D) - X$. $\delta^+(X) = A(X, \overline{X})$ and $\delta^-(X) = A(\overline{X}, X)$.
*strong* or *strongly connected*: $\delta^+(X) \neq \emptyset$ for all proper nonempty subsets $X$ of $V(D)$. (Or equivalently, $\delta^-(X) \neq \emptyset$ for all such $X$.)
*reachability* in digraphs means directed reachability: $uR^+v$ if there is a directed $uv$-walk (or equivalently a directed $uv$-path); say $v$ *is reachable from* $u$.
$R_D^+(v)$ means vertices reachable from $v$; $R_D^-(v)$ means vertices that can reach $v$.
*branching* or *outbranching* or *arborescence*: rooted tree where all edges directed outward from root. Can be constructed via *Directed Local TCM*; special cases *Directed BFS* and *Directed DFS* (only consider edges going outward from root).
*acyclic* digraph (computer scientists call it a $DAG$): no directed cycles.

**Lemma:** An acyclic digraph has at least one source and at least one sink.

*distance in networks:* given digraph $D$, nonnegative weight (distance) $w(a)$ for each arc, $d(u, v)$ is minimum length (total weight) of $uv$-path.

**Flows**

*Network* $(D, c)$: digraph $D$ ($V = V(D)$, $A = A(D)$), each arc has nonnegative *capacity* $c(a)$.
$\delta^+X$, $\delta^+v$, $\delta^-X$, $\delta^-v$: arcs out of/into set of vertices $X$ or single vertex $v$.
$\overline{X} = V - X$.
A *flow* is $f \in \mathbf{R}^A$, i.e., $f$ is a function $f : A \to \mathbf{R}$.

If $S \subseteq A$ then $f(S)$ means $\sum_{a \in S} f(a)$.

If $X \subseteq V$ and $v \in V$ then $f^+(X)$, $f^+(v)$, $f^-(X)$, $f^-(v)$ mean $f(\delta^+X)$, $f(\delta^+v)$, $f(\delta^-X)$, $f(\delta^-v)$ respectively.

$\partial f(X) = f^+(X) - f^-(X)$ is *net outflow from* $X$ and $\partial f(v)$ is defined similarly.

**Proposition (Vertex additivity of net flow):** For any $f : A(D) \to \mathbf{R}$ and any $X \subseteq V(D)$, $\partial f(X) = \sum_{v \in X} \partial f(v)$.
Say $f$ *conserved* at $v$ if $f^+(v) = f^-(v)$, i.e., $\partial f(v) = 0$.
$f$ is a *circulation* if $f$ is conserved at all $v \in V$.
Given *supply vertex* $x$ and *demand vertex* $y$ an *xy-flow* (or often just *flow*) is a flow $f : A \to \mathbf{R}$ conserved at every $v \in V - \{x, y\}$.
*Feasible* flow in $(D, c)$: flow (not necessarily $xy$-flow) that satisfies $0 \le f(a) \le c(a)\ \forall\ a \in A(D)$.
The *value* of an $xy$-flow is $\operatorname{val} f = \partial f(x)$ (net flow out of $x$). (Linear function on $xy$-flows.)

**Special flows:** - if $P$ directed $xy$-path, $\chi_P(a) = 1$ if $a \in A(P)$, 0 otherwise. $\operatorname{val} \chi_P = 1$.
- if $P$ direction-insensitive $xy$-path, $\overrightarrow{\chi}_P(a) = 1$ if $P$ uses $a$ forwards, $-1$ if $P$ uses $a$ backwards, 0 otherwise. $\operatorname{val} \overrightarrow{\chi}_P = 1$.
- if $C$ directed cycle (may or may not contain $x$ or $y$), $\chi_C(a) = 1$ if $a \in A(C)$, 0 otherwise. $\operatorname{val} \chi_C = 0$.

An *xy-cut* is a set of arcs $K$ for which there exists some set of vertices $X$ with $x \in X$, $y \notin X$ and $K = \delta^+X$.
The *capacity* of the cut $K = \delta^+X$ is just $c(K) = c^+(X)$.
A *minimum xy-cut* means an $xy$-cut of minimum capacity.

**Lemma:** $\partial f(X) = \operatorname{val} f$ for any $xy$-cut $\delta^+X$.

**Observation:** For any feasible $xy$-flow $f$ and $xy$-cut $K = \delta^+X$, we have $\operatorname{val} f \le c(K)$. Moreover, equality holds if and only if $f(a) = c(a)\ \forall\ a \in \delta^+X$ and $f(a) = 0\ \forall\ a \in \delta^-X$.

*Residual network* $(D^*, c^*) = \operatorname{Res}(D, c, f)$: shows how we can modify flow $f$. Same vertex set as $D$. Up to two arcs for every arc $a$ of $D$:

if $f(a) < c(a)$ add $a^+$, copy of $a$, to $D^*$ with capacity $c^*(a) = c(a) - f(a)$ (shows we can push extra flow along $a$);

if $f(a) > 0$ add $a^-$, opposite to $a$, to $D^*$ with capacity $c^*(a) = f(a)$ (shows that we can 'push some flow backwards' along $a$, i.e., reduce flow in $a$).

*$f$-augmenting path*: directed $xy$-path in $D^*$.

**Observe:** If $P$ is an $f$-augmenting path then we can augment along $f$ to get a new feasible $xy$-flow of higher value. **Ford-Fulkerson Algorithm** repeatedly searches for $f$-augmenting path and augments; if no $f$-augmenting path, vertices $X$ reachable from $x$ in $D^*$ give minimum cut $\delta^+X$. **Edmonds-Karp Algorithm** is special version guaranteed to terminate in polynomial time.

**Note:** If all capacities integral, F-F Algorithm shows that an integer-valued maximum flow exists.

**Max Flow Min Cut Theorem:** The value of a maximum $xy$-flow equals the capacity of a minimum $xy$-cut.

**Note:** Can allow infinite capacities, MFMC Theorem still holds.

**Note:** Vertex capacities $c(v)$ implemented by splitting $v$ into $v^-$ with all in-arcs, $v^+$ with all out-arcs, and arc $v^-v^+$ of capacity $c(v)$.

*Support* of $f$, $\operatorname{supp} f = \{a | f(a) \neq 0\}$. *Acyclic* flow has acyclic support.

**Flow Decomposition Algorithm:** Given nonnegative flow $f_0$, first remove flow around directed cycles (remove circulation $f_C$) to get acyclic $f_A$, then remove flow along maximal directed paths to get 0.

**Gallai's Flow Decomposition Theorem (FDT):** Every nonnegative flow $f_0$ may be written

$$f_0 = \overbrace{\alpha_1\chi_{C_1} + \alpha_2\chi_{C_2} + \ldots + \alpha_s\chi_{C_s}}^{f_C} + \overbrace{\beta_1\chi_{P_1} + \beta_2\chi_{P_2} + \ldots + \beta_t\chi_{P_t}}^{f_A}$$

where

(i) $f_C$ is a nonnegative circulation, $s \geq 0$, $\alpha_1, \ldots, \alpha_s > 0$, and $C_1$, …, $C_s$ are directed cycles;

(ii) $f_A$ is a nonnegative acyclic flow, $t \geq 0$, $\beta_1, \ldots, \beta_t > 0$, and each $P_i$ is a directed $x_iy_i$-path with $\partial f_0(x_i) > 0$, $\partial f_0(y_i) < 0$; and

(iii) if $f_0$ is integer-valued then we may choose $\alpha_1, \ldots, \alpha_s, \beta_1, \ldots, \beta_t$ to all be integers, so that $f_C$ and $f_A$ are also integer-valued. ∎