Combinatorial optimization - Structures and Algorithms, *GeorgiaTech, Fall 2011* Lectures 5 – 7: Sep 6 – 13

László Végh

References are from the book *Connections in Combinatorial Optimization* by András Frank (**F**). All proofs covered in the lectures not included in the notes can be found in the book.

1 Representations of minimum cuts

Let $\lambda(G)$ denote the edge-connectivity of the undirected graph G, that is, $\lambda(G) = k$ if G is k-edge-connected but not $k+1$-connected. X is called a minimum cut, if $d(X) = k$. X and $V - X$ define the same cut. If X or $V - X$ is a singleton, then it is called a trivial minimum cut, otherwise a **proper min-cut**. Minimum cuts will also be refered here as tight sets.

Two sets $X, Y \subseteq V$ are called **crossing** if all four sets $X - Y$, $Y - X$, $X \cap Y$, $V - (X \cup Y)$ are nonempty. Note that if X and Y are crossing, then so are X and $V - Y$. Let $d(X, Y)$ and $\bar{d}(X, Y)$ denote the number of edges between $X - Y$ and $Y - X$, and between $X \cap Y$ and $V - (X \cup Y)$, respectively.

Claim 1.1 (**F** Prop. 7.1.1)**.** *If X and Y are crossing tight sets, then $X \cap Y$, $X \cup Y$, $X - Y$ and $Y - X$ are also tight. Furthermore, $d(X, Y) = \bar{d}(X, Y) = 0$.*

An easy consequence is that if k is odd, then there are no crossing tight sets, and therefore all tight sets admit a simple tree representation.

Theorem 1.2 (**F** Thm. 7.1.2)**.** *If $\lambda(G)$ is odd, then there exists a tree $H = (U, F)$ and a mapping $\varphi : V \to U$ such that for any $e \in F$, the pre-images of the two components of $H - e$ define a minimum cut, and conversely, every minimum cut can be obtained in this form.*

Such a structure cannot be given for even $\lambda(G)$. For example, a cycle for $k = 2$ contains a minimum cut corresponding to every pair of edges. Cycles turn out to give a canonical example for even k.

Lemma 1.3 (**F** Lem. 7.1.3)**.** *Assume $k = \lambda(G)$ is even. Assume every proper minimum cut is crossed by another proper minimum cut. Then G can be obtained from a circuit by replacing every edge by $k/2$ parallel edges.*

The 2-edge-connected, loopless graph $H = (U, F)$ is called a **cactus**, if every 2-node-connected block is a cycle. Equivalently, H is a cactus if every edge is contained on exactly one cycle. Further equivalently, H is a cactus if it can be obtained from a cycle by a sequence of two operations: (i) add a new node, and connect it to an old node by 2 parallel edges; (ii) subdivide an edge by a new node.

Theorem 1.4 (Dinitis, Karzanov, Lomonosov, **F** 7.1.8)**.** *Let $\lambda(G) = k$ be even. Then there exists a cactus $H = (U, F)$ and a mapping $\varphi : V \to U$ such that there is a one-to-one correspondence between the minimum cuts of G and H. That is, for any $e, f \in F$ contained in the same cycle, the pre-images of the two components of $H - \{e, f\}$ form a minimum cut in G, and every minimum cut in G can be obtained this way.*

The proof in **F**, given by Frank and Fleiner, uses induction by contracting proper min-cuts not crossed by any proper min-cuts. If no such proper min-cut exists, then Lemma 1.3 can be applied.

2 Gomory-Hu trees

Let $G = (V, E)$ be a connected undirected graph, and $g : E \to \mathbb{R}_+$ a capacity function. For a pair of nodes $u, v \in V$, let $\lambda_g(u, v)$ denote the maximum number of edge-disjoint path between u and v under the capacity g. By Menger's theorem, $\lambda_g(u, v) = \min_{u \in X, v \notin X} d_g(X)$. An X giving the minimum here is called uv-critical. ($d_g(X) = \sum_{x \in X, y \notin X} g(xy)$.)

A tree $T = (V, F)$ on the same node-set is called a **Gomory-Hu tree** for G if it satisfies the following properties. For each edge $e \in F$, let $m(e) = d_g(X_e)$, where X_e is either component of $T - e$. We require that for every pair of nodes $u, v \in V$, $\lambda_g(u, v)$ is the minimum $m(e)$ value on the unique uv-path in T, and for the edge e giving the minimum, X_e is uv-critical.

Theorem 2.1 (Gomory, Hu, **F** Thm 7.2.2)**.** *Every graph $G = (V, E)$ posesses a Gomory-Hu tree, that can be found by $n - 1$ max flow computations.*

See the algorithmic proof in **F**. The existence of a Gomory-Hu tree implies that there exist a set of $n - 1$ sets containing a uv-critical set for every pair $u, v \in V$, and gives a concise representation of such sets.

2.1 Application: minimum k-cuts

Given a graph $G = (V, E)$ and a cost function $w : E \to \mathbb{R}_+$, our aim is to find a set of edges J minimizing $w(J)$ so that $G - J$ has at least k-connected components. The problem is NP-complete. The following simple approximation algorithm was given by Saran and Vazirani: construct a Gomory-Hu tree (with capacities $g \equiv w$), and pick the $k - 1$ lightest edges (w.r.t. m) of T. The corresponding edge set $J \subseteq E$ achieves a factor $2 - 2/k$.

To see this, first let us verify that $G - J$ has at least k components. Let $L \subseteq F$ be the set of the $k - 1$ lightest edges of T. Then $T - L$ has $k - 1$ components, and J is the set of all edges between these components in G.

Let J_0 be the optimal k-cut, with $G - J_0$ containing partition classes $V_1, V_2, \ldots, V_k$ (we may assume that an optimal solution contains exactly k sets). Then $2w(J_0) = \sum_{i=1}^{k} d_w(V_i)$. W.l.o.g. assume that V_k is the component maximizing $d_w(V_i)$. Then

$$(2 - \frac{1}{k}) w(J_0) \geq \sum_{i=1}^{k-1} d_w(V_i).$$

We shall proof that $w(J)$ is at most the RHS. Indeed, contract all set V_i into single nodes, and let L_0 be a spanning tree in the contraction of the Gomory-Hu tree T. Let us designate the contraction of V_k as the root node, and orient all edges of L_0 towards the root. By the minimal choice of T_0,

$w(J) = m(L) \leq m(L_0)$. We claim that $m(L_0)$ is at most the RHS. Indeed consider L_0 as a subset of the original Gomory-Hu tree. For each V_i, $i < k$, there is exactly one edge $x_i y_i \in L_0$ leaving V_i with $x_i \in V_i$, $y_i \notin V_i$. By the definition of the Gomory-Hu tree, the minimum cut value between x_i and y_i is at least $m(x_i y_i)$, and since V_i is an $x_i y_i$-cut, $m(x_i y_i) \leq w(V_i)$. Hence the claim follows.

3 Sparse certificates for connectivity

3.1 Directed edge-connectivity

The directed graph $D = (V, E)$ with root node $r \in V$ is called rooted k-edge-connected, if for every $v \in V - r$, there exists k edge-disjoint directed paths from r to v. Equivalently, $\rho(X) \geq k$ for every $r \notin X \subsetneq V$. As a generalization, D is called (k, ℓ)-edge-connected, if for every $v \in V - r$, there exists k edge-disjoint directed paths from r to v, and ℓ edge-disjoint directed paths from v to r. Equivalently, $\rho(X) \geq k$ and $\delta(X) \geq \ell$ for every $r \notin X \subsetneq V$. Observe that a graph is (k, k)-EC if and only if it is k-EC.

By the following lemma, all minimal k-EC connected graphs have precisely $k(n-1)$ edges.

Lemma 3.1 (**F** Lem 7.4.1)**.** *Let $D = (V, E)$ is minimally rooted k-EC with root node $r \in V$. Then $\rho(v) = k$ for every $v \neq r$.*

Since a minimally rooted (k, ℓ)-EC graph is a (not necessarily disjoint) union of a min. rooted k-(out-)connected and a min. rooted ℓ-(in-)connected graph, it follows that

Theorem 3.2 (**F** Thm 7.4.3)**.** *A minimally (k, ℓ)-EC directed graph has at most $(k + \ell)(n - 1)$ edges. Consequently, a minimally k-EC graph has at most $2k(n - 1)$-edges.*

The above bound is sharp for every pair (k, ℓ): take a star, with the central node connected to everyone else by k out- and ℓ in-edges.

3.2 Undirected edge-connectivity

The bounds on directed graphs immediately give bounds on minimal k-EC undirected graphs.

Claim 3.3 (**F** Thm 7.5.3)**.** *A minimally k-EC undirected graph has a most $k(n - 1)$ edges.*

The proof follows by the observation that if we direct all edges in both directions in a minimally k-EC connected undirected graph, then we get a min k-EC directed graph. As we shall see, this bound is not sharp.

By a maximal forest decomposition of the graph $G = (V, E)$, we mean the partition of E into forests $F_1, F_2, \ldots, F_t$ such that F_1 is an (arbitrary) spanning forest in G, F_2 is a spanning forest in $G - F_1$, F_i is a spanning forest in $G - \cup_{j<i} F_j$. Let $E_k = F_1 \cup \ldots \cup F_k$ denote the union of the first k forests.

Claim 3.4 (**F** Prop. 7.5.4)**.** *For any set $X \subseteq V$, $d_{E_k}(X) \geq \min\{k, d_E(X)\}$ for a maximal forest decomposition.*

This immediately implies

Theorem 3.5 (Nishizeki, Polyak, **F** Thm 7.5.5)**.** *If G is k-EC, then (V, E_k) is also k-EC.*

This gives an efficient algorithm to identify a small certificate of k-edge-connectivity. Yet it is not true that in a k-EC graph there exist k spanning trees such that the paths between u and v in the k trees yield k edge-disjoint paths.

Since the forest decomposition is arbitrary, we can start with F_1 containing all edges incident to a certain node v_1, F_2 containing all edges in $G - F_1$ of v_2, etc. This implies that $|F_i| \leq n - i$, and the following strengthening of Claim 3.3 follows.

Theorem 3.6 (Mader)**.** *A minimal k-EC graph has at most $kn - \binom{k}{2}$ edges.*

3.3 Scan First Search decompositions

Let us now define a restricted class of forest decompositions. To find a spanning forest, we apply the following generic **Scan First Search (SFS) algorithm**.

In the undirected graph $G = (V, E)$, nodes are sorted into three classes: scanned (S), marked (M) and undiscovered (U). In each step we maintain a forest F. The algorithm starts with $U = V$, $S = M = \emptyset$. We start by picking an arbitrary node $s \in U$, and move it to S. We add all neighbours v of s to M, and add all edges vs to the forest F. This step is repeated by picking an arbitrary $s \in U$ if $M = \emptyset$, and a node $s \in M$ otherwise.

The SFS algorithm always returns a spanning forest. Note that BFS is a special implementation of SFS, but DFS is not. For a running of the SFS algorithm, by the corresponding ordering we mean the order in which the nodes are being scanned. Orderings that can be obtained for some running of the SFS algorithm are called **scan-first orderings**.

In general, let $v_1, v_2, \ldots, v_n$ be an ordering of the nodes of $G = (V, E)$. We say that this ordering is **contiguous** if every component C of G is an interval in the ordering, that is, if $i < j < k$ and $v_i, v_j \in C$, then $v_k \in C$. Furthermore we require that if v_j is not the first node of a component C, then there exists an edge $v_i v_j \in E$ with $v_i < v_j$.

Observe that Max Adjacency (MA) orderings, defined in Lec. 4, are all contiguous orderings. Moreover, it can easily be seen that every scan-first ordering is contiguous.

The converse of this statement is also true. With a contiguous ordering, we can associate a forest by connecting every node v_j to the first node v_i with $v_i v_j \in E$, $i < j$ if there is any. We call this the forest decomposition belonging to the ordering.

Claim 3.7 (**F** Thm. 7.5.1)**.** *Scan-first orderings are the same as contiguous orderings.*

A useful property of MA-orderings is the following.

Claim 3.8 (**F** Prop. 7.5.2)**.** *If $v_1, \ldots, v_n$ is an MA-ordering of G, and F is the associated tree, then $v_1, \ldots, v_n$ is also an MA ordering for $G - F$.*

3.4 Node-connectivity

The undirected graph $G = (V, E)$ is called **k-node-connected (k-NC)** between $u, v \in V$, if G contains k internally node disjoint paths between u and v. The most convenient way for dual characterization is in terms of **set pairs**. Let us call (X, Y) a set pair, if $X, Y \neq \emptyset$, and $X \cap Y = \emptyset$. Let $w(X, Y) = |V - (X \cup Y)|$ denote the number of node outside both sets. Then G is k-NC between u and v, if $w(X, Y) + d(X, Y) \geq k$ for every set pair (X, Y) with $x \in X$, $y \in Y$. G is k-NC, if it is k-NC between every pair of nodes $u, v \in V$.

Exercise 3.9. *If $|V| \geq k+1$, then G is k-node-connected if and only if $w(X,Y) \geq k$ for all set pairs with $d(X,Y) = 0$. Equivalently, $|N(X)| \geq k-1$ whenever $X \neq \emptyset$, $X \cup N(X) \neq V$.*

We shall prove that an appropriate maximal forest decomposition gives a witness to k-edge-connectivity as well. By an SFS decomposition of a graph we mean a maximal forest decomposition obtained by SFS-algorithms.

Theorem 3.10 (Even, Itkis, Rajsbaum, **F** Thm 7.5.7-7.5.9)**.** *If $G = (V,E)$ is k-node-connected, and $F_1, F_2, \ldots, F_k$ is an SFS-decompostion, then (V, E_k) is also k-node-connected, where $E_k = F_1 \cup \ldots \cup F_k$.*

This will immediately prove that Theorem 3.6 is also true for node-connectivity:

Theorem 3.11 (Mader, **F** Thm 7.5.9.)**.** *A minimal k-NC graph has at most $kn - \binom{k}{2}$ edges.*

Claim 3.8 gives an efficient way to compute a wittness for k-node-connectivity: compute an MA ordering $v_1, \ldots, v_n$ (this can be done in $O(m)$ time). Let us connect every v_j to its first k neighbours v_ℓ with $\ell < j$ if there is at least k of them, and to all such neighbours if there are less then k. This gives precisely the set E_k.

Theorem 3.10 will follow from the next lemma:

Lemma 3.12 (**F** Thm 7.5.7)**.** *Let $G = (V,E)$ be an arbitrary graph with SFS decomposition $F_1, \ldots, F_t$, and let $E_k = F_1 \cup \ldots \cup F_k$. If u and v are in the same component of F_k, then (V, E_k) is k-NC between u and v.*

Proof. We shall prove by induction on k. The theorem is trivial for $k = 1$. Let $J = F_2 \cup \ldots \cup F_k$ and $G' = G - F_1$.

For $k > 2$, we shall prove $w(X,Y) + d_{E_k}(X,Y) \geq k$ for every (X,Y) with $x \in X$, $y \in Y$. If $d_{F_1}(X,Y) > 0$, then by the induction hypothesis for G' and J, $w(X,Y) + d_J \geq k-1$, hence the claim follows as $d_{E_k} \geq d_J > 1$.

Assume now $d_{F_1}(X,Y) = 0$. Let $v_1, \ldots, v_n$ be the order in which nodes were scanned in SFS algorithm giving F_1. W.l.o.g. we may assume $v_1 \notin X$. Let v_j be the first node with $v_j \in V - (X \cup Y)$. We claim that $d_{G'}(v_j, X) = 0$. Indeed, when v_j was scanned, all edges connected v_j to unprocessed nodes are added to F_1, including all edges going to X.

Let $Y' = Y \cup \{v_j\}$. By induction, $w(X,Y') + d_J(X,Y') \geq k-1$. Since $w(X,Y') = w(X,Y) - 1$, and $d_J(X,Y') = d_J(X,Y)$, the claim follows. □

Proof of Theorem 3.10. We need to prove $w(X,Y) + d_{E_k}(X,Y) \geq k$ for all set pairs (X,Y). If F_k has a component intersecting both X and Y, this is guaranteed by the lemma above. Otherwise, we claim that $d_{E_k}(X,Y) = d_G(X,Y)$, and thus the claim follows as G is k-NC. Indeed, if F_k does not connect nodes in X and Y, and there exists an edge $xy \in E - E_k$ with $x \in X$, $y \in Y$, then we could have added xy to F_k, a contradiction. □

Let $\kappa_G(u,v)$ denote the maximum number of node disjoint paths in G between u and v.

Theorem 3.13 (**F** Thm 7.5.10)**.** *If $v_1, \ldots, v_n$ is an MA ordering of G, then $\kappa(v_{n-1}, v_n) = d(v_n)$.*

An important consequence is this:

Theorem 3.14 (Mader, **F** Thm 7.5.11)**.** *Every minimally k-NC graph has a node of degree exactly k.*

3.5 Constructive characterization of chordal graphs

Let us give a different application of MA-orderings. An undirected graph $G = (V, E)$ is called **chordal**, if every cycle of length at least four has a chord, that is, an edge connecting two nodes not adjacent on a cycle. The following theorem gives a constructive characterization of chordal graphs, providing an NP-certificate of this property.

Theorem 3.15 (Dirac, **F** Thm 7.5.13)**.** *G is chordal if it can be obtained from a single node by the following operation: add a new node, and connect it to the nodes of a clique.*

Tarjan and Yannakakis proved that if G is chordal, then an MA-ordering provides such a construction sequence (**F** Thm 7.5.14).