Combinatorial optimization - Structures and Algorithms, *GeorgiaTech, Fall 2011* Lectures 1 – 4: Aug 23 – Sep 1

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 Elementary connectivity properties

An undirected/directed graph $G = (V, E)$ is called **k-edge-connected (k-EC)**, if for every subset $\emptyset \neq X \subsetneq V$, $d(X) \geq k$ for undirected graphs and $\rho(X) \geq k$ for directed graphs. A 1-EC directed graph is called **strongly connected**.

1.0.1 2-edge-connected undirected graphs

We start by giving a simple characterization of 2-edge-connected undirected graphs.

Theorem 1.1 (**F** Thm 2.1.1)**.** *For an undirected graph $G = (V, E)$, the following are equivalent:*

(i) G is 2-EC.

(ii) There are 2 edge-disjoint paths connecting any two nodes of G

(iii) G has an **ear decomposition***, that is, there exists a sequence of graphs $G_0, G_1, \ldots, G_q = G$, so that G_0 is a cycle, and G_{i+1} can be obtained from G_i by adding a path P_i connecting two (possibly identical) nodes of G_i.*

(iv) G can be built up from a single node by iteratively applying the following two operations: (a) add a new edge (possibly loops); (b) subdivide an existing edge by a new node.

For a 1-EC graph, a minimum cost 1-EC graph is a minimum cost spanning tree, and can be found by the greedy algorithm. For 2-EC graphs, the situation changes:

Claim 1.2. *Finding a minimum cost 2-EC subgraph of a 2-EC graph is NP-complete, even for the case when all edge costs are identical.*

Proof. If $|V| = n$, and each edge cost is 1, the cost of a 2-EC subgraph is at least n. Furthermore, it can attain the value n if and only if it has a Hamiltonian cycle. □

Still, from the ear decompositions it is easy to derive the following bound:

Claim 1.3. *Every minimally 2-EC graph on n nodes have at most $2n - 2$ edges.*

An inverse question is connectivity augmentation: if G is not k-EC, what is the minimum cost of an edge set whose addition makes G k-EC? For $k = 1$, this can again be solved by the greedy algorithm, and for $k = 2$, it is again NP-complete by reducing Hamiltonian cycle.

Surprisingly, if we want to find a minimum cardinality set of new edges to be added, the problem is polynomially solvable. Let us call a set $\emptyset \neq U \subsetneq V$ **solid**, if $d(U) < 2$, but for any $\emptyset \neq X \subsetneq U$, $d(X) \geq 2$. It is easy to see that any two solid sets are disjoint. Let t_0 and t_1 denote the number of solid sets with $d(X) = 0$ and $d(X) = 1$, respectively.

Theorem 1.4 (Eswaran, Tarjan; **F** Thm 2.1.5)**.** *The minimum number γ of new edges whose addition makes an undirected graph 2-EC equals $t_0 + \lceil t_1/2 \rceil$.*

In the proof in **F**, the first step should be shrinking cycles (instead of 2-solid sets).

1.1 Strongly connected digraphs

A similar characterization can be given for strongly connected digraphs.

Theorem 1.5 (**F** Thm. 2.2.1)**.** *For a directed graph $G = (V, E)$, the following are equivalent:*

(i) G is strongly connected.

(ii) There is a directed path connecting any node to any other node.

(iii) G has an **ear decomposition**, *that is, there exists a sequence of graphs $G_0, G_1, \dots, G_q = G$, so that G_0 is a directed cycle, and G_{i+1} can be obtained from G_i by adding a directed path P_i connecting two (possibly identical) nodes of G_i.*

(iv) G can be built up from a single node by iteratively applying the following two operations: (a) add a new arc (possibly loops); (b) subdivide an existing arc by a new node.

For directed graphs, the k-EC subgraph problem is already NP-complete for $k = 1$, since one can reduce the directed Hamiltonian cycle problem. Also, making a digraph strongly connected by adding a minimum cost set of new arcs is NP-complete, while it is polynomially solvable for adding a minimum number of new arcs.

For this, we need the following structural observations. Let us define an equivalence relation *sim* on the nodes of an undirected graph G, so that $u \sim v$ if there is a directed path from u to v and one from v to u. If we contract the equivalence classes of this relation to single nodes, we obtain an acyclic graph. Let c_{source} and c_{sink} denote the number of source and sink nodes of the contracted graph (those having in/outdegree 0, respectively).

Theorem 1.6 (Eswaran, Tarjan; **F** Thm 2.2.7)**.** *The minimum number of new edges needed to make a digraph strongly connected is equal to* $\max\{c_{source}, c_{sink}\}$.

Another possible ways of making a not strongly connected digraph strongly connected is contracting or reorienting arcs. These shall be described later in the Lucchesi-Younger theorem.

Another fundamental area we shall investigate are graph orientations. Here the first question is: when does an undirected graph have a strongly connected orientation?

Theorem 1.7 (Robbins, **F** Thm 2.2.8)**.** *An undirected graph has a strongly connected orientation if and only if it is 2-EC.*

Proof. 2-EC is trivially necessary. For sufficiency, observe that an undirected ear decomposition immediately yields a directed ear decomposition. □

2 Orientations with degree constraints

A directed graph is called **eulerian**, if $\rho(v) = \delta(v)$ for any node $v \in V$. An undirected graph is called **eulerian**, if $d(v)$ is even for all nodes. It is easy to see the following.

Claim 2.1. *An undirected graph has a eulerian orientation if and only if it is eulerian.*

As an extension, a directed graph is **near-eulerian**, if $|\rho(v) - \delta(v)| \leq 1$.

Claim 2.2. *Every undirected graph has a near-eulerian orientation.*

Proof. Add an arbitrary perfect matching on the set of odd nodes, and find a eulerian orientation. □

An oriented complete graph is called a tournament.

Theorem 2.3 (Landau, **F** Thm 2.3.1)**.** *There exists a tournament on n nodes with in-degrees $m_1 \geq m_2 \geq \ldots \geq m_n$ if and only if*

$$\sum_{i=1}^{n} m_i = \binom{n}{2}$$

and

$$\sum_{i=1}^{k} m_i \leq \binom{k}{2} + k(n-k) \quad \forall k = 1, \ldots, n.$$

Exercise 2.4. *In a chess tournament, if there is a winner, he gets 2 points and the loser 0 points. If it is draw, both players get 1 point. Decide whether $m_1, \ldots, m_n$ can be the final scores of the players assuming everyone played with everyone.*

Open problem 2.5. *In a soccer tournament, the difference is that the winner gets 3 points, the loser 0 points, and both get 1-1 by draw. Decide whether $m_1, \ldots, m_n$ can be the final scores of the players assuming everyone played with everyone.*

If not every pair of teams play, but only those corresponding to the edges of a graph, then Pálvölgyi showed that the problem is NP-complete. However, complexity for the complete graph is still open.

Theroem 2.3 will be a consequence of the following general augmentation theorem. For a set $X \subseteq V$, $e(X)$ denotes the number of edges incident to X by at least one endnode, and $i(X)$ is the number of edges spanned in X. That is, $e(X) = i(X) + d(X)$.

Theorem 2.6 (Hakimi, **F** Thm 2.3.2)**.** *Let $G = (V, E)$ be an undirected graph, and $m : V \to \mathbb{Z}_+$. There exists an orientation of G with $\rho(v) = m(v)$ for every node if and only if $m(V) = |E|$ and $e(X) \geq m(X)$ for every subset $X \subseteq V$.*

The proof is based on the submodularity of $e(X)$. Observe how Landau's theorem can be derived.

More generally, we can ask for upper and lower bounds on the in-degrees.

Theorem 2.7 (**F** Thm 2.3.5)**.** *Let $G = (V, E)$ be an undirected graph and $f, g : V \to \mathbb{Z}_+$.*

(A) There exists an orientation with $\rho(v) \geq f(v)$ for every $v \in V$ if and only if $e(X) \geq f(X)$ for every $X \subseteq V$.

(B) There exists an orientation with $\rho(v) \leq g(v)$ for every $v \in V$ if and only if $i(X) \leq g(X)$ for every $X \subseteq V$.

(C) There exist an orientation with $f(v) \leq \rho(v) \leq g(v)$ if and only if both conditions in (A) and (B) hold.

For part (A), we have seen to algorithmic proofs. In the first, we improve the current orientaiton by reversing the orientation of certain $u - -v$ paths with $\rho(u) < f(u)$ and $\rho(v) > f(v)$. Namely, let X be the set of nodes reachable from the set $\{u \in V : \rho(u) < f(u)\}$. If this set contains a v with $\rho(v) > f(v)$, we can get a better orientation by reorienting a $u - -v$ path. Otherwise it is easy to see that $e(X) < f(X)$.

The second is a preflow-push type algorithm. We maintain an orientation in each step and distance labels $\Theta : V \to \{0, 1, \dots, n\}$ with the following properties: (i) If $\rho(v) > f(v)$ then $\Theta(v) = 0$; and (ii) If uv is a directed edge, then $\Theta(u) \leq \Theta(v) + 1$.

As long as there exists a node u with $\rho(u) < f(u)$, we perform the following. If there exists a v with $\Theta(u) = \Theta(v) + 1$, we reorient the edge uv. If there exists none and $\Theta(u) < n$, then we increase $\Theta(u)$ by one. If there exists a level $0 < k < n$ so that no node has level k while there are some nodes above k, the set $X = \{u : \Theta(u) > k\}$ will violate the condition.

Both algorithms have natural counterparts for part (B). For part (C), we observed that if we start any of the algorithms for part (B) with an orientation already satisfying the lower bounds, then this property will be maintained.

Part (C) means that whenever we have an orientation with the lower bounds and another one with the upper bounds, then there should also exists one satisfying all these bounds simultaneously. This is called the **linking property**, which we shall also see in other contexts.

2.1 Application to subgraphs with forbidden degrees

A general subgraph problem is the following. For an undirected graph $G = (V, E)$, let us be given a set $F(v) \subseteq \{0, 1, \dots, d(v)\}$ for every node $v \in V$. A subgraph $G' = (V, E')$ is $F - avoiding$, if $d_{G'}(v) \notin F(v)$ for any $v \in V$. For example, we obtain a perfect matching if we forbid every degree except for 1. The general problem is however NP-complete, as it is easy to formulate the 3-dimensional matching problem. The following theorem gives a sufficient condition.

Theorem 2.8 (Shirazi, Verstraëte, **F** Thm 2.3.8.). *If $|F(v)| \leq \lfloor d_G(v)/2 \rfloor$ for every $v \in V$, then there exists an F-avoiding subgraph.*

The original proof used deep algebraic methods. Yet the following generalization by Frank, Szabó and Lau, involving orientations, can be proved in two lines.

Theorem 2.9 (**F** Thm 2.3.9.). *If G has an orientation $\vec{G}$ with $\rho(v) \geq |F(v)|$ for every node v, then there exists an F-avoiding subgraph.*

The previous theorem then follows starting from a near-eulerian orientation. Also, using Theorem 2.7, we get the following more general theorem:

Theorem 2.10 (**F** Thm 2.3.10.). *If $e_G(V) \geq \sum_{v \in X} |F(v)|$ holds for every $X \subseteq V$, then there exists an F-avoiding subgraph.*

2.2 Orientations vs bipartite matchings

We show that Hakimi's orientation thm (Thm 2.6) is essentially equivalent to bipartite matchings. Let us be given a bipartite graph $G = (A \cup B; E)$ with $|A| = |B|$. By the Kőnig-Hall theorem, there exists a perfect matching if and only if $|\Gamma(Z)| \geq |Z|$ for every $Z \subseteq A$.

Let us define $m(v) = 1$ for every $v \in B$ and $m(v) = d_G(v) - 1$ for every $v \in A$. It is easy to see that in such an orientation, the edges oriented towards B form a perfect matching. It can be verified that the condition in Hakimi's theorem is equivalent to the Kőnig-Hall condition.

For the other direction, let us take an orientation problem $G = (V, E)$ with $m : V \to \mathbb{Z}_+$. Let A consist of $m(v)$ copies of each node $v \in V$, and let B correspond to the set of edges. Let us connect the representative of $uv \in E$ to all copies of u and v. It is easy to see that a perfect matching gives a good orientation.

The algorithms presented for orientation naturally correspond to algorithms for perfect matchings. The path reorientation algorithm to the alternating path method, while the preflow-type algorithm also has a natural analogue.

Moreover, Theorem 2.7 enables to derive more general matching theorems with lower and upper bounds on the degrees. The linking property also appears.

Theorem 2.11 (Mendelsohn, Dulmage, **F** Thm 2.4.3)**.** *Let $G = (A \cup B; E)$ be a bipartite graph, $X \subseteq A$ and $Y \subseteq B$. If there exists a matching M_X covering X and a matching M_Y covering Y, then there exists one covering $X \cup Y$.*

This can be proved easily looking at the symmetric difference of M_X and M_Y. However, it also derives from the linking property in Thm 2.7. Let us define lower bounds $f(v) = d(v) - 1$ if $v \in A$, $f(v) = 1$ if $v \in Y$ and $f(v) = 0$ if $v \in B - Y$. For upper bounds, let $g(v) = d(v) - 1$ if $v \in X$, $g(v) = d(v)$ if $v \in A - X$ and $g(v) = 1$ if $v \in Y$. Orientations with such lower and upper bounds correspond to matchings covering Y and X, respectively.

3 Undirected minimum cuts

For a graph $G = (V, E)$, let $\lambda(G)$ denote the minimum size of a cut in G, that is, $\min\{d(X) : \emptyset \neq X \subsetneq V\}$. Let $\lambda_G(u, v)$ denote the maximum number of disjoint paths between u and v, equal to $\min\{d(X) : u \in X \subseteq V - v\}$.

The value $\lambda_G(u, v)$ can be determined by a maximum flow computation. Clearly, $\lambda(G) = \min_{u,v \in V} \lambda_G(u, v)$, hence $\lambda(G)$ can be computed by $\binom{n}{2}$ max flows.

Yet $n - 1$ max flows suffice: for some u, v, compute $\lambda_G(u, v)$, and then contract the set $\{u, v\}$; let $G' = G/\{u, v\}$. Then $\lambda(G) = \min\{\lambda_G(u, v), \lambda(G')\}$, and iterate the above process.

En even more efficient, yet very simple algorithm was given by Nagamochi and Ibaraki. Let $\{v_1, \ldots, v_n\}$ be an ordering of the nodes of graph $G = (V, E)$. Let $V_i = \{v_1, \ldots, v_i\}$. It is called a **max adjacency (MA)** ordering, if for any $i < j$, $d(V_{i-1}, v_i) \geq d(V_{i-1}, v_j)$. Such an ordering can be easily found in $O(m)$ time.

Theorem 3.1 (Nagamochi,Ibaraki, **F** Thm 6.3.1)**.** *If $v_1, \ldots, v_n$ is an MA-ordering of $G = (V, E)$, then $\lambda_G(v_{n-1}, v_n) = d_G(v_n)$.*

The theorem can be proved by simple induction. Then $\lambda(G) = \min\{\lambda_G(v_{n-1}, v_n), G/\{v_{n-1}, v_n\}\}$ suggests the following algorithm.

Let us compute an MA-ordering, note $d_n(v)$, contract the last two nodes, and repeat the same algorithm. The minimum cut value will be the smallest among the noted values by the above observation. This altogether gives an $O(mn)$ algorithm.

This naturally extends to the capacitated case, when the edges can have arbitrary weights $g(uv)$, and we want to compute the minimum cut $\lambda_g(G) = \min\{\sum_{u \in X, v \in V-X} g(uv) : \emptyset \neq X \subsetneq V\}$.