**Ling Li**, `ling@cs.caltech.edu`
March 5, 2001

## 15.1 Matching, Components, and Edge cover (Collaborate with Xin Yu)

First show $\ell = c$ by proving $\ell \leq c$ and $c \leq \ell$.

- For a maximum matching $M$ in $G$, let $V'$ be the set of vertices covered by $M$. Since any vertex in $V'$ is incident on some edge in $M$ and $M \subseteq E$, $G' = (V', M)$ is a subgraph of $G$. And the number of connected components in $G'$ is just $\ell$, the size of $M$. Thus $\ell \leq c$.
- Let $G'$ be a subgraph of $G$ with $c$ connected components. Choose an arbitrary edge out from each connected components in $G'$. Those edges form a matching in $G$ since no two edges share a vertex (those components are disjoint from each other). The size of the matching is $c$. Hence $c \leq \ell$.

Then prove $\ell = c = n - k$.

- A maximum matching $M$ in $G$ covers $2\ell$ vertices. Since there are no isolated vertices in $G$, we can always cover the rest $n - 2\ell$ vertices with at most $n - 2\ell$ edges. Together with the $\ell$ edges in $M$, we can construct an edge cover with at most $n - \ell$ edges. Thus $k \leq n - \ell$, or $\ell \leq n - k$.
- Let $E''$ be a minimum edge cover of $G$. Since every vertices are on some edge in $E''$, so $(V, E'')$ is a subgraph of $G$. And the number of connected components in this subgraph is at least $|V| - |E''| = n - k$. (We can start from a graph $G'' = (V, \emptyset)$, with $n$ disjoint vertices as $n$ connected components. Add edges in $E''$ one by one into $G''$. Each edge reduces the number of components in $G''$ at most 1. Thus for graph $(V, E'')$, there are at least $n - k$ connected components.) Thus $c \geq n - k$.

Together with $\ell = c$, we've got $\ell = c = n - k$.

**Ling Li**, ling@cs.caltech.edu


### 15.2 Tutte matrix

$\Rightarrow$ Suppose $G$ has a perfect matching $M$. For $(u,v) \in M$ and $u < v$, set $x_{u,v} = 1$, and set all other variables to 0. Thus each row in $T(G)$ has exactly one non-zero entry with value 1 or $-1$, and each column in $T(G)$ also has exactly one non-zero entry with value 1 or $-1$, since in a perfect matching each vertex is incident on exactly one edge. Thus we know the determinant of $T(G)$ is either 1 or $-1$, not zero. That is, the polynomial $\det(T(G))$ is not zero for some assignment of $x_{u,v}$. So $\det(T(G)) \neq 0$.

$\Leftarrow$ Assume $\det(T(G)) \neq 0$. From the analysis below ($T(G)$ is that kind of matrix $A$), there must be some permutation which consists of only even-length cycles such that

$$\prod_{u \in V} T(G)_{u,p(u)} \neq 0.$$

Since $T(G)_{u,p(u)} \neq 0$ iff $(u, p(u)) \in E$, there are several even-length cycles

$$(u_{1,1}, u_{1,2}, \dots, u_{1,\ell_1}), (u_{2,1}, u_{2,2}, \dots, u_{2,\ell_2}), \dots, (u_{m,1}, u_{m,2}, \dots, u_{m,\ell_m}),$$

such that $u_{j,k}$ are distinct vertices in $V$ and $(u_{j,k}, u_{j,(k+1) \bmod \ell_j}) \in E$. So there are several vertex-disjoint even-length circles in graph $G$ and those circles covers all the vertices in $G$ (since $\ell_1 + \ell_2 + \cdots + \ell_m = n = |V|$). Thus $G$ has a perfect matching. (One example is the matching consists of edges $(u_{j,2k-1}, u_{j,2k})$ where $j = 1, 2, \dots, \ell_m$ and $k = 1, 2, \dots, \frac{\ell_m}{2}$.) $\square$

Below is the analysis used in the above proof.

Let's explore the condition for $|A| \neq 0$, where $A$ is a $n \times n$ matrix with $A_{u,v} = -A_{v,u}$, and $|A| = \det(A)$ is the determinant of $A$. Essentially,

$$|A| = \sum_{p \in S_n} \sigma(p) \prod_{i=1}^{n} A_{i,p(i)}, \tag{1}$$

where $S_n$ is the set of all permutations of $1, 2, \dots, n$ and $\sigma(p)$ is the sign of the permutation $p$. And we call $\sigma(p) \prod_{i=1}^{n} A_{i,p(i)}$ a term of $|A|$.

Any permutation $p$ can be uniquely divided into several cycles,

$$p = (i_{1,1}, i_{1,2}, \dots, i_{1,\ell_1}), (i_{2,1}, i_{2,2}, \dots, i_{2,\ell_2}), \dots, (i_{m,1}, i_{m,2}, \dots, i_{m,\ell_m}),$$

That is, $i_{j,k}$ are distinct numbers and, $p(i_{j,k}) = i_{j,(k+1) \bmod \ell_j}$ for $j = 1, 2, \dots, m$ and $k = 1, 2, \dots, \ell_j$, and $\ell_1 + \ell_2 + \cdots + \ell_m = n$. (Here we may define $\bmod \ell_j$ takes value 1 to $\ell_j$.) For another permutation

$$p' = (i_{1,1}, i_{1,2}, \dots, i_{1,\ell_1}), \dots, (i_{j,\ell_j}, \dots, i_{j,2}, i_{j,1}), \dots, (i_{m,1}, i_{m,2}, \dots, i_{m,\ell_m}), \tag{2}$$

where $p' = p$ except for some $j$, $p'(i_{j,k}) = i_{j,(k-1) \bmod \ell_j}$. Thus $\sigma(p') = \sigma(p)$ since any rotation of the $j^{\text{th}}$ circle changes the sign by $(-1)^{\ell_j - 1}$ and it takes 2 rotations from $p$ to $p'$.

However, since $A_{u,v} = -A_{v,u}$, we have

$$\sigma(p) \prod_{i=1}^{n} A_{i,p(i)} = (-1)^{\ell_j} \sigma(p') \prod_{i=1}^{n} A_{i,p'(i)}.$$

**Ling Li**, ling@cs.caltech.edu


Thus if $p$ has a cycle of odd length, we can find another permutation $p'$ by (2) and those terms associated with $p$ and $p'$ cancel each other. More generally, we can pair those permutations with odd-length cycle by using (2) for the smallest odd-length cycle. (For each cycle $(i_1, i_2, \ldots, i_\ell)$ we can assign a value $\min\{i_1, i_2, \ldots, i_\ell\}$ to it, and the smallest odd-length cycle is the one with smallest value and odd length.) So after cancellation, (1) becomes

$$|A| = \sum_{p \in E_n} \sigma(p) \prod_{i=1}^{n} A_{i,p(i)},$$

where $E_n$ is the set of all permutations consists of only even-length cycles.

**Ling Li**, ling@cs.caltech.edu


### 15.3 Jobs scheduling

Collect some notations here: dependency graph $P = (J, A)$ such that if $(J_1, J_2) \in A$ then $J_1$ must be executed before $J_2$; a legal schedule $s : J \to \{1, \ldots, T\}$; and a graph $G_P = (J, E)$ where $(J_1, J_2) \in E$ if there is no path from $J_1$ to $J_2$ or from $J_2$ to $J_1$ in $P$.

(a) For any legal schedule $s$, if there is a path from $J_1$ to $J_2$ in $P$, $J_1$ must be executed before $J_2$. That is, $s(J_1) < s(J_2)$. Symmetrically, if there is a path from $J_2$ to $J_1$, $s(J_2) < s(J_1)$. Thus a necessary condition for $s(J_1) = s(J_2)$ is that there is no path from $J_1$ to $J_2$ or from $J_2$ to $J_1$ in $P$, or equivalently, $(J_1, J_2) \in E$. (The symmetry in $J_1$ and $J_2$ also implies that $G_P$ is an undirected graph.)

let $M' = \{(J_1, J_2) : s(J_1) = s(J_2)\}$. From above discussion, $M' \subseteq E$. Since a job is executed only once, no two edges in $M'$ share a vertex. Thus $M'$ is a matching in $G_P$. Since $M$ is the maximum matching in $G_P$, $|M'| \leq |M|$. It is easy to calculate that the schedule $s$ takes time $T = |J| - |M'|$. Thus $T \geq |J| - |M|$.

(b) We want to construct a legal schedule from the maximal matching $M$ in $G_P$. First, construct a mapping $f$ so that for each edge $(J_1, J_2) \in M$, $f(J_1) = f(J_2) = J_{1,2}$; and for other vertices $J_i$, $f(J_i) = J_i$. (We just created some new vertices such as $J_{1,2}$ trying to combine those vertices in one pair in a matching.) Convert $P$ to $P' = (J', A')$ by converting $J$ to $J'$ by $f$ and $A$ to $A'$ correspondingly (replacing every edge $(J_i, J_j) \in A$ with $(f(J_i), f(J_j))$ in $A'$). Since one combination reduces the number of vertices by 1, $|J'| = |J| - |M|$. From $P$ is a directed acyclic graph (dag) and for each edge $(J_1, J_2) \in M$ there is no path from $J_1$ to $J_2$ or from $J_2$ to $J_1$ in $P$, $P'$ is also a dag. Then $P'$ has a topological sort $s' : J' \to \{1, 2, \ldots, |J| - |M|\}$ such that if there is a path from $v_i$ to $v_j$ in $P'$ then $s'(v_i) < s'(v_j)$. Let $s(J_i) = s'(f(J_i))$. This is a legal schedule since

- $\left|s^{-1}(i)\right| = \left|f^{-1}(i)\right| \leq 2 \forall i \in \{1, 2, \ldots, |J| - |M|\}$.
- If $(J_1, J_2) \in A$, then $(J_1, J_2) \notin M$ and then $f(J_1) \neq f(J_2)$ and $(f(J_1), f(J_2)) \in A'$. Thus $s(J_1) < s(J_2)$.

And the time is $T = |J| - |M|$. From (a), this is an optimal schedule.

(c) The idea of the algorithm to find an optimal schedule is in (b):

- Create graph $G_P$ and find a maximum matching $M$ in $G_P$. The time complexity is $O(|A|\,|J|)^*$ plus $O(|E|\sqrt{|J|})$
- Convert $P$ into $P'$ by using mapping $f$. This takes $O(|J| + |A|)$ time.
- Find the topology sort $s'$ in $P'$, taking time $O(|J'| + |A'|) \subseteq O(|J| + |A|)$.
- Assign $s$ from $s'$ and $f$. This takes $O(|J|)$ time.

Totally, the time complexity is $O(|J|\,|A| + |E|\sqrt{|J|})$. Since there is no edge $(J_1, J_2)$ in $G_P$ if there is a path from $J_1$ to $J_2$ or $J_2$ to $J_1$ in $P$, $|E| + |A| \leq 2\,|J|^2$. Thus the time complexity is also $O(|J|^3)$.

*For a dag $P$, we can first make a topology sort for $P$ and then use the topology sort to calculate $G_P$, which is simply a tracking down along the sorted edges.

### 15.4 Spanning tree with some degree

For any edge set $E$, define $E_v = \{e : e \in E \wedge v \text{ is incident on } e\}$ and $E_v^c = E - E_v$. Let $d_E(v) = |E_v|$. That is, $d_E(v)$ is the degree of $v$ in a graph with $E$ as the edge set. And it is obviously $E_v \subseteq E$ and $d_E(v) + |E_v^c| = |E|$.

Given $G = (V, E)$, let $M_k = \{E' : E' \subseteq E \wedge d_{E'}(v) \leq k\}$. We want to prove that $M_k$ is a matroid.

- If $E'' \subseteq E'$ and $E' \in M_k$, we have $E''_v \subseteq E'_v$ and $E' \subseteq E$. Thus $d_{E''}(v) \leq d_{E'}(v) \leq k$ and $E'' \subseteq E$. So $E'' \in M_k$.
- For $E', E'' \in M_k$ and $|E'| < |E''|$, if $d_{E'}(v) \leq k - 1$, select an arbitrary edge $e \in E'' - E'$. We have $d_{E' \cup \{e\}}(v) \leq d_{E'}(v) + 1 \leq k$ and $E' \cup \{e\} \subseteq E' \cup E'' \subseteq E$. Thus $E' \cup \{e\} \in M_k$. If $d_{E'}(v) = k$,
$$\left|(E'_v)^c\right| = \left|E'\right| - d_{E'}(v) < \left|E''\right| - k \leq \left|E''\right| - d_{E''}(v) = \left|(E''_v)^c\right|,$$
since $d_{E''}(v) \leq k$. Select an arbitrary edge $e \in (E''_v)^c - (E'_v)^c$. Since $v$ is not incident on $e$, $e \notin E'_v$, so $e \notin E'$. Thus $e \in E'' - E'$, and $d_{E' \cup \{e\}}(v) = d_{E'}(v) = k$. So again we have $E' \cup \{e\} \in M_k$.

Thus $M_k$ is a matroid. Since we already know that the collection of forests of $G$ form another matroid, say $M_f$. Then $M_f \cap M_k$ gives a set of forests with the degree of $v$ in those forests no more than $k$.

Let $T$ be a set with maximum size in $M_f \cap M_k$. If $|T| = |V| - 1$, then $T$ must be a tree (since $T$ is a forest with $|V|$ vertices). Then $T$ is a spanning tree of $G$ with $d_T(v) \leq k$. If there is a spanning tree $T$ of $G$ with $d_T(v) \leq k$, then $T \in M_k$ and $T \in M_f$ and thus $T \in M_k \cap M_f$. And $T$ must be a maximum size set in $M_f \cap M_k$ since no forest with $|V|$ vertices can have more than $|V| - 1$ edges. So, the problem of whether there is a spanning tree $T$ of $G$ with $d_T(v) \leq k$, is equivalent to the problem of whether a maximum size set in $M_f \cap M_k$ has the size $|V| - 1$.

**Ling Li**, ling@cs.caltech.edu


### 15.5 Weighted matching in bipartite graph

Let $U_S = \{u : u \in U \wedge \exists v \in V, (u,v) \in S\}$ which is the set of all $u$'s appearing in $S$. Consider the mapping $S \to U_S : f_{U,S}((u,v)) = u$ with $u \in U$. Obviously $f_{U,S}$ is map-on and $|U_S| \leq |S|$. $f_{U,S}$ is a 1-1 map iff $S \in F_U$, since when $S \in F_U$, $(u_1,v_1) \neq (u_2,v_2)$ in $S$, where $u_1,u_2 \in U$ and $v_1,v_2 \in V$, gives $u_1 \neq u_2$, and when $f_{U,S}$ is a 1-1 map, $(u_1,v_1) \neq (u_2,v_2)$ in $S$ gives $f_{U,S}((u_1,v_1)) \neq f_{U,S}((u_2,v_2))$, i.e., $u_1 \neq u_2$. Thus $|U_S| = |S|$ iff $S \in F_U$. We can similarly define $V_S = \{v : v \in V \wedge \exists u \in U, (u,v) \in S\}$ and $|V_S| = |S|$ iff $S \in F_V$.

From reasons below, $F_U$ is a matroid.

- If $S' \subseteq S$ and $S \in F_U$, $(u_1,v_1),(u_2,v_2) \in S' \Rightarrow (u_1,v_1),(u_2,v_2) \in S \Rightarrow u_1 \neq u_2$. Thus $S' \in F_U$.
- For $S, S' \in F_U$ and $|S'| < |S|$, we have $|U_{S'}| < |U_S|$. Select an arbitrary $u \in U_S - U_{S'}$ and let $(u,v) = f_{U,S}^{-1}(u) \in S$ be the pre-image of $u$. Since $u \notin U_{S'}$, we have $(u,v) \notin S'$ and there is no $(u',v') \in S'$ with $u' = u$. Hence $S' \cup (u,v) \in F_U$.

Similarly, $F_V$ is also a matroid. The intersection of these two matroids, $F_U \cap F_V$, consists of all subsets $S$ of $E$ satisfying that if $(u_1,v_1),(u_2,v_2) \in S$ and $(u_1,v_1) \neq (u_2,v_2)$, then $u_1 \neq u_2$ and $v_1 \neq v_2$. That is, $F_U \cap F_V$ consists of all matchings in $G = (U,V,E)$. Thus the problem of finding a matching in $G$ with the maximum weight is equivalent to the one of finding the maximum weight set in $F_U \cap F_V$.

Without loss of generality, let $p = v_0, u_1, v_1, u_2, \ldots, u_k, v_k, u_{k+1}$ be an augmenting path with respect to a matching $M$ in $G$. Then $(u_i,v_i) \in M$ for $i = 1,2,\ldots,k$, and $v_0$, $u_{k+1}$ are free in $M$, i.e., $u_{k+1} \notin U_M$. Apply $p$ to $M$, we get $M' = M \oplus p$. Then $M'$ consists of $(v_{i-1},u_i)$ (or $(u_i,v_{i-1})$), for $i = 1,2,\ldots,k+1$, and all edges other than $(u_i,v_i)$ in $M$. Thus $|M'| = |M| + 1$ and $U_{M'} = U_M \cup \{u_{k+1}\}$, $|U_{M'}| = |U_M| + 1$. Since $M \in F_U$, we have $|M| = |U_M|$. So $|M'| = |U_{M'}|$. Thus $M' \in F_U$. Similarly we can prove $M' \in F_V$. That is, an augmenting path preserves membership in each of the matroids.

**Ling Li**, ling@cs.caltech.edu


### 15.6 Not a matroid (Collaborate with Xin Yu)

It is easy to show that $F$ is an ideal: If $I \subseteq J$ and $J \in F$, then each vertex in $J$ has degree at most $k$ and thus each vertex in $I$ also has degree no more than $k$. So $I \in F$. Let $d_{\max}$ be the maximum degree of vertices in $G$. We want to show that for $k < d_{\max}$, $F$ is not a matroid, or, the exchange axiom doesn't hold for $F$.

First, define a procedure to construct subgraphs in $G$:

**Saturate:** Given an initial subgraph $G' \in F$, add an edge $e$ in $G$ to $G'$ if this keeps $G' \cup \{e\} \in F$. Continue adding edges in some order, or, random order, until no edge can be added and we get $G''$.

The **Saturate** procedure gives a 'maximal' subgraph $G'' \in F$ and $G' \subseteq G''$, where $G'$ is the initial subgraph.

Let $v$ be a vertex in $G$ with degree $d_{\max}$. Let $G'_1$ be a subgraph of $G$ with only $k$ edges with $v$ as one endpoint. Since $k < d_{\max}$, we can find another subgraph $G'_2$ that has only $k$ edges with $v$ as one endpoint, but different from $G'_1$. Applying **Saturate** to $G'_1$ and $G'_2$, we get $G_1$ and $G_2$ respectively. Since degrees of $v$ in both $G'_1$ and $G'_2$ are $k$, no edge with $v$ as one endpoint will be added during the **Saturate** procedures and thus $G_1 \neq G_2$.

If $|G_1| < |G_2|$, we've done since no edge in $G_2 - G_1$ can be added into $G_1$ while keeping $G_1 \in F$. By symmetry, if $|G_1| > |G_2|$, we've also done. Only when for all such initial graphs $G'_1$ and $G'_2$ we have $|G_1| = |G_2|$, there may be some chance that $F$ is a matroid.

How about $k \geq d_{\max}$? When $k \geq d_{\max}$, $F$ is basically the collection of all subgraphs of $G$, and it is always a matroid. This is because for $I, J \in F$ and $|I| < |J|$, choosing an arbitrary $e \in J - I$ gives $I \cup \{e\} \in F$.

So, for any $k$, $F$ is generally not a matroid for graphs with $d_{\max} > k$.