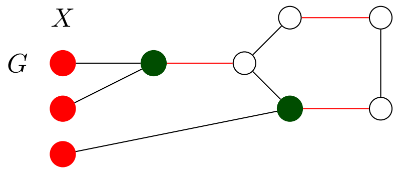
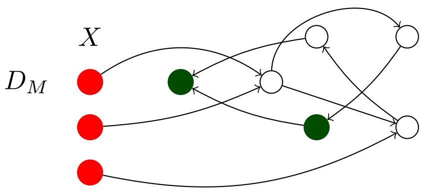
In this second lecture we will show how to compute a maximum matching in arbitrary graphs in polynomial time.

1 Edmond's Algorithm for non-bipartite matchings

We have already seen in the last lecture that a matching is maximum if and only if there is no augmenting path. This even holds in non-bipartite graphs. Hence, we just start with a matching, find augmenting paths and update the matching to obtain a larger matching. We can do this until we find a maximum matching. For bipartite graphs it was quite easy to find an augmenting path. But how do we do this for non-bipartite graphs?

Let M be a matching of a graph $G = (V, E)$. Let X be the set of uncovered vertices of G. We define a new auxiliary directed graph $D_M = (V, A)$ that we use to find M-augmenting paths: D_M also has the vertex set V, but arc set $A = \{(u, v) \mid \exists x \in V$ such that $ux \in E \setminus M$ and $xv \in M\}$. See figure below.



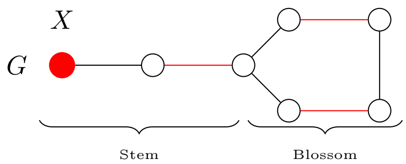


Observe that a directed path in D_M corresponds to an (even length) alternating path in G. Furthermore, if there is an M-augmenting path in G, then there is a directed path in D_M starting at a vertex in X and ending at a neighbor of X. Unfortunately, the converse does not necessarily hold: There may be a directed path in D_M starting at a vertex in X and ending at a neighbor of X that does *not* correspond to an augmenting path. Such a path must necessarily have a prefix that is a *flower*.

Definition 1. *An M-flower is an M-alternating walk $v_0, v_1, v_2, ..., v_t$ (numbered so that we have $v_{2k-1}v_{2k} \in M$ and $v_{2k}v_{2k+1} \notin M$) satisfying:*

1. *$v_0 \in X$.*
2. *$v_0, v_1, v_2, ..., v_{t-1}$ are distinct.*
3. *t is odd.*
4. *$v_t = v_i$ for an even i.*

The portion of the flower from v_0 to v_i is called the stem, *while the portion from v_i to v_t is called the* blossom.



The next lemma shows that alternating walks between exposed vertices either correspond to an augmenting path or contain a flower.

Lemma 1. *Let M be a matching in G, and let $P = (v_0, v_1, ..., v_t)$ be a shortest alternating walk from X to X. Then either P is an M-augmenting path, or $v_0, v_1, ..., v_j$ is an M-flower for some $j < t$.*

Proof. If $v_0, v_1, ..., v_t$ are all distinct, P is an M-augmenting path. Otherwise, assume $v_i = v_j, i < j$, and let j be as small as possible so that $v_0, v_1, ..., v_{j-1}$ are all distinct. We will prove that $v_0, v_1, ..., v_j$ is an M-flower. The first two properties of a flower are clearly satisfied, by construction. It cannot be the case that j is even, since then $v_{j-1}v_j \in M$, which gives a contradiction in both of the following cases:

- if $i = 0$, then $v_{j-1}v_j \in M$ contradicts $v_0 \in X$.
- if $0 < i < j - 1$, then $v_{j-1}v_j \in M$ contradicts the fact that M is a matching, since v_i is already matched to a vertex other than v_{j-1}.

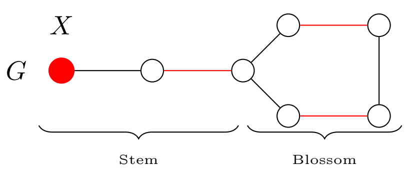
This proves that j is odd. It remains to show that i is even. Assume for a contradiction that i is odd. This means that $v_i v_{i+1}$ and $v_j v_{j+1}$ are both edges in M. Then $v_{j+1} = v_{i+1}$ (since both are equal to the other endpoint of the unique matching edge containing $v_j = v_i$), and we may delete the cycle from P to obtain a shorter alternating path from X to X. □

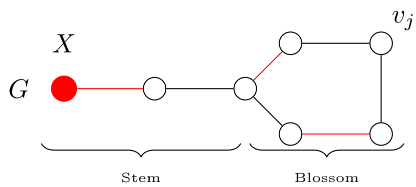
So, if we find a shortest such path, we immediately know that we either have an M-augmenting path (in which case we can increase the size of the matching) or we have an M-flower. So what do we do when we have an M-flower? In this case, our goal is simplify the graph by *shrinking* the blossom of the flower. To do that, we first do something else.

Given a flower $F = (v_0, v_1, ..., v_t)$ with blossom B, observe that for any vertex $v_j \in B$ it is possible to modify M to a matching M' satisfying:

1. Every vertex of F is the endpoint of an edge of M', except v_j.
2. M' agrees with M outside of F, i.e., $M \Delta M' \subseteq F$.
3. $|M| = |M'|$.

To do so, we take M' to consist of all the edges of the stem which do not belong to M, together with a matching in the blossom that covers every vertex, except vertex v_j, as well as all the edges in M outside of F.





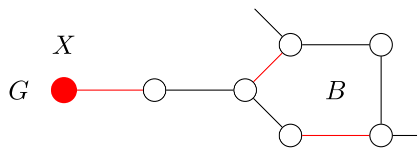
Hence, whenever a graph G with matching M contains a blossom B, we may simplify the graph by *shrinking* B. That is, we remove all vertices and edges of the blossom B and add a new vertex b which is adjacent to all vertices to which $V(B)$ was incident to. Additionally, we modify M by removing all matching edges from within B. This is formalized as follows.

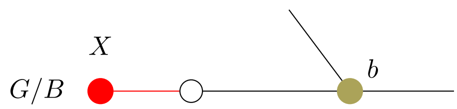
Definition 2 (Shrinking a blossom)**.** *Given a graph $G = (V, E)$ with a matching $M \subseteq E$ and a blossom B, the* shrunk *graph G/B with matching M/B is defined as follows:*

- $V(G/B) = (V \setminus B) \cup \{b\}$
- $E(G/B) = E \setminus E[B] \cup E_B$
- $M/B = M \setminus E[B]$,

where $E[B]$ denotes the set of edges within B, b is a new vertex disjoint from V, and $E_B = \{ub \mid u \in V \setminus V(B),$ and $\exists v \in V(B)$ such that $uv \in E\}$.

Observe that M/B is a matching in G, because the definition of a blossom precludes the possibility that M contains more than one edge with one but not both endpoints in B. Observe that G/B may contain parallel edges between vertices if G contains a vertex from $V \setminus V(B)$ which is connected to B by more than one edge.





The relation between matchings in G and matchings in G/B is summarized in the following theorem.

Theorem 1. *Let M be a matching of G and let B be an M-blossom. Then, M is a maximum-size matching if and only if M/B is a maximum-size matching in G/B.*

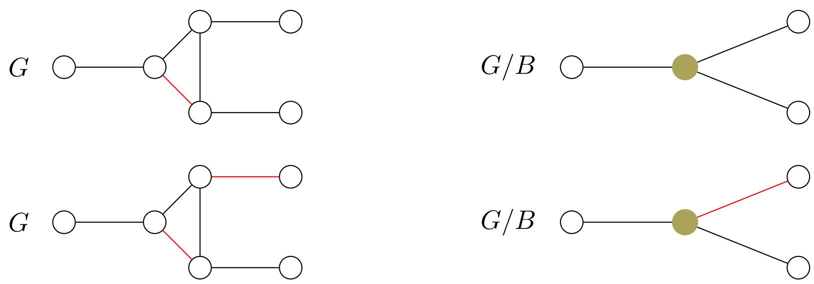
Proof. We show both directions individually.
($\Longrightarrow$) We prove this via contraposition, i.e., we suppose N is a matching in G/B larger than M/B and show that M is not maximum.

Pulling N back to a set of edges in G, it is incident to at most one vertex of B. Expand this to a matching N^+ in G by adjoining $\frac{1}{2}(|B| - 1)$ edges within B to match every other vertex in B. Then we have $|N^+| - |N| = (|B| - 1)/2$, while at the same time $|M| - |M/B| = (|B| - 1)/2$. The latter follows because B is an M-blossom, so there are $(|B| - 1)/2$ edges of M in B; then M/B contains all the corresponding edges in M except those $(|B| - 1)/2)$ many. We conclude that $|N^+|$ exceeds $|M|$ by the same amount that $|N|$ exceeds $|M/B|$.

($\Longleftarrow$) Again, we prove this via contraposition: If M is not of maximum size, we want to show that also M/B is not maximum.

First, we change M to another matching M' of equal cardinality, in which B is an entire flower, i.e., if S is the stem of the flower whose blossom is B, then we may take $M' = M\Delta S$. Note that M'/B is of the same cardinality as M/B, and b is an unmatched vertex of M'/B. Since M' is not a maximum-size matching in G, there exists an M'-augmenting path P. At least one of the endpoints of P is not in B. So number the vertices of P by $u_0, u_1, ..., u_t$ with $u_0 \notin B$, and let u_i be the first vertex on P which is in B (if there is no such vertex then $u_i = u_t$). This sub-path $u_0, u_1, ..., u_i$ is an (M'/B)-augmenting path in G/B. Hence, by Lemma 1, M' is not maximum and therefore also M. □

Note that if M is a matching in G that is not of maximum size, and B is a blossom with respect to M, then M/B is not a maximum-size matching in G/B. If we find a maximum-size matching N in G/B, then the proof gives us a way to 'unshrink' the blossom B in order to turn N into a matching N^+ of G of size larger than that of M. However, it is important to note that N^+ will not, in general, be a maximum-size matching of G, as the example shows. View the example from the top left graph, and then in clockwise direction.



The final algorithm is now at follows.

Algorithm for Maximum Matching in general Graphs:

1. Start with the empty matching $M = \emptyset$, and let X be the set of exposed vertices (initially V).
2. Construct the directed graph D_M.
3. As long as D_M contains a path $\hat{P}$ from X to $N(X)$
   Find such a path $\hat{P}$ of minimum length (number of edges)
4. Let P be the alternating path in G corresponding to $\hat{P}$.
5. If P is an M-augmenting path, set $M = M\Delta P$. Update X, construct new directed graph D_M and go to point 3.
6. Else: P contains a blossom B. Modify flower as above.
   - Recursively find max-size matching M' in G/B.
   - if $|M'| = |M/B|$, return M (M was maximum).
   - else unshrink M' (as in proof of Theorem) to obtain matching M'' in G, where $|M''| > |M|$. Update M and X, construct D_M and go to point 3.

This algorithm and our previous lemma and theorem lead to the following theorem.

Theorem 2. *We can compute a maximum matching in an arbitrary graph in time* $O(|E| \cdot |V|^2)$.

Proof. The correctness of the algorithm is established by Lemma 1 and Theorem 1. Let us focus on the running time. We can compute X and D_M in linear time (in $O(|V| + |E|)$). Shrinking a blossom also takes linear time. We can only perform $O(|V|)$ such shrinkings before terminating or increasing M. The number of times we increase $|M|$ is bounded by $O(|V|)$. Hence, the overall running time is bounded by $O((|E| + |V|) \cdot |V|^2)$, and since $|E| \geq \frac{1}{2}|V|$ (since there are no isolated vertices in G), we can bound this by $O(|E| \cdot |V|^2)$. □

One can speed up the algorithm and obtain the following theorem, which we will not prove.

Theorem 3. *We can compute a maximum matching in an arbitrary graph in time* $O(|E| \cdot \sqrt{|V|})$.

In fact, one can generalize this result to weighted graphs. We do not prove this here and only mention the result.

Theorem 4. *We can compute a maximum-weight matching in an arbitrary edge-weighted graph in time* $O(|E| \cdot |V|^2)$.

2 Edmonds-Gallai Decomposition

Next, we take a look at the structure of maximum matchings in non-bipartite graphs and try to find a certain decomposition that helps constructing *all* maximum matchings.

Consider some graph $G = (V, E)$. We let $\nu(G)$ denote the cardinality of a maximum matching in G. For some vertex set $U \subseteq V$, let $G - U$ denote the subgraph of G obtained by deleting the vertices of U and all edges incident with them. Let $o(G - U)$ denote the number of connected components of $G - U$ that contain an odd number of vertices. Let M be a matching in $G - U$ and consider a component of $G - U$ with an odd number of vertices. There must be at least one unmatched vertex v in this component, since any matching necessarily covers an even number of vertices. Treating M as a matching in G, it is possible that we could increase the size of M by matching v with some vertex in U. However, we can add at most $|U|$ edges to M in this manner, since the vertices in U will eventually all be matched. Thus, any matching in G must have at least $o(G-U)-|U|$ unmatched vertices. This shows that the maximum size of a matching is upper-bounded bu $(|V| + |U| - o(G - U))/2$, for any subset U. The following theorem strengthens this result.

Theorem 5 (Tutte-Berge Formula)**.** *Let* $G = (V, E)$ *be a graph. Then*

$$\nu(G) = \max_{M} |M| = \min_{U \subset V} |V| + |U| - o(G - U))/2 \ ,$$

where the maximization is over all matching M *in* G.

Proof. We will consider the case that G is connected. Otherwise, the result follows by adding the formulas for the individual connected components. The proof proceeds by induction on the number of vertices in G. If G has at most one vertex then the result holds trivially. Otherwise, suppose that G has at least two vertices. We consider two cases:

Case 1: G contains a vertex v that is covered by *all* maximum matchings.

The subgraph $G - v$ cannot have a matching of size $\nu(G)$, otherwise that would give a maximum matching for G that leaves v unmatched. Thus $\nu(G - v) = \nu(G) - 1$. By induction, the result holds for the graph $G - v$, so there exists a set $U' \subset V - v$ that achieves equality in the Tutte-Berge Formula. Defining $U = U' \cup \{v\}$, we see that

$$\begin{aligned}
\nu(G) &= \nu(G - v) + 1 \\
&= \Big(|V - v| + |U'| - o(G - v - U')\Big)/2 + 1 \\
&= \Big((|V| - 1) + (|U| - 1) - o(G - U)\Big)/2 + 1 \\
&= \Big(|V| + |U| - o(G - U)\Big)/2 \ .
\end{aligned}$$

Case 2: For every vertex $v \in G$, there exists a maximum matching that does not cover v.

We will prove that each maximum matching leaves exactly one vertex uncovered. Suppose the contrary, that is, each maximum matching leaves at least two vertices uncovered. We choose a maximum matching M and its two uncovered vertices u and v such that we minimize $d(u, v)$, the distance between vertices u and v. If $d(u, v) = 1$, then the edge uv can be added to M to obtain a larger matching, a contradiction.

Otherwise, $d(u, v) \geq 2$ so we may fix an intermediate vertex t on some shortest u-v path. By the assumption of the present case, there is a maximum matching N that does not cover t. Furthermore, we may choose N such that its symmetric difference with M is minimal. If N does not cover u, then (N, u, t) contradicts our choice of (M, u, v). Thus, N covers u and, by symmetry, also v. Since N and M both leave at least two vertices uncovered, there exists a second vertex $x \neq t$ that is covered by M

but not by N. Let xy be the edge in M that is incident with x. If y is also uncovered by N then $N \cup xy$ is a larger matching than N, a contradiction. Then $N \cup \{xy\} \setminus \{yz\}$ is a maximum matching that does not cover t and has smaller symmetric difference with M than N does. This contradicts our choice of N, so each maximum matching must leave exactly one vertex uncovered.

Then $\nu(G) = (|V| - 1)/2$. The Tutte-Berge Formula then follows by choosing $U = \emptyset$. □

Now that we have established the correctness of the Tutte-Berge Formula, one might ask how we can compute such a set $U \subseteq V$ giving equality in the formula?