Fall 2015, MATH-566

Maximum Matchings in Graphs (Edmonds' Blossom Algorithm)

Source: Chapter 10.1, 10.5

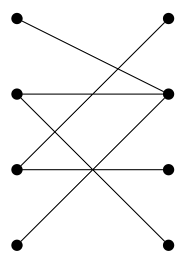
Let $G = (V, E)$ be a graph. **Matching** M a subset of E, such that graph (M, E) has maximum degree one.

Problem: For a graph G, find maximum matching M.

1: Formulate the maximum matching problem as an integer programming (IP).

2: Is there a condition on G that guarantees that a relaxation of (IP) to linear program always has an integral optimal solution? Notice that the matrix A is an incidence matrix of G.

3: Show that finding maximum matching in a bipartite graph G can be done using maximum flow algorithm. Is there a *certificate* that a matching is maximum?



Definition: Let M be a matching in graph $G = (V, E)$. A vertex $v \in V$ is **covered** if exists $e \in M$ such that $v \in e$, otherwise v is **exposed**. A matching is called a **perfect matching** if all vertices are covered.

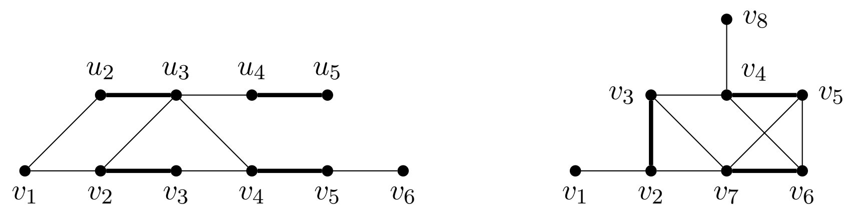
Perfect matchings are covered in the book but we are skipping them.

Notice: maximizing M is the same as minimizing exposed vertices.

Inspiration by flow: use *augmenting paths*.

A path P is M**-alternating** if $E(P) \setminus M$ is a matching. An M**-alternating** path P is M**-augmenting** if P has positive length and its endpoints are exposed in M. Augmented $M' = M \Delta E(P)$.

4: Assume there is a matching M (thick lines). Find augmenting path(s) and augment M.

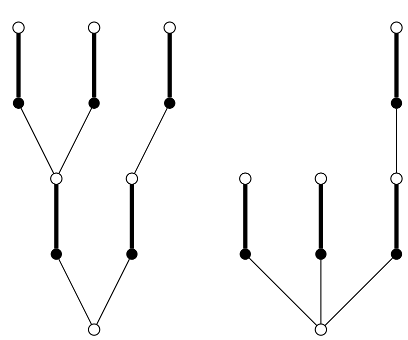


Theorem 10.7 Let G be a graph with matching M. Then M is maximum iff there is no M-augmenting path.

5: Prove Theorem 10.7

6: Can we use augmenting walks instead of paths? It particular, examine walk $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_5, v_4, v_8$ in the graph on the right-hand side.

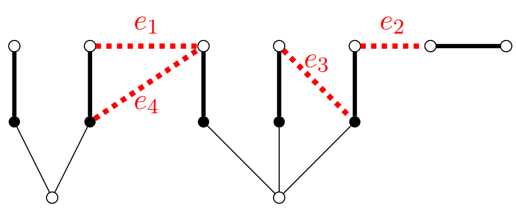
Idea: Start from exposed vertices as roots and build **alternating forest** F. Alternate non-matching edges and matching edges. This gives layers of matching edges and non-matching edges.



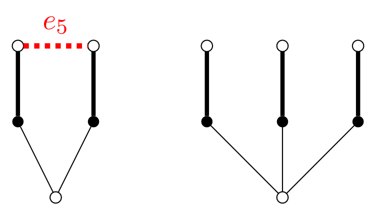
Notice that not all edges of G are present in the forest F! Call vertex **outer** vertex if it is in even distance from the root (white ones).

Assume building of the alternating forest by picking an edge adjacent to outer vertex and trying to extend the forest edge by edge.

7: What happens when we try to add any of the dotted edges?

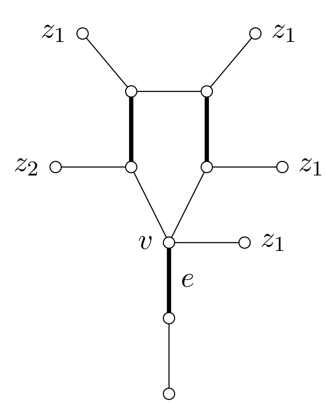


8: What happens with e_5?

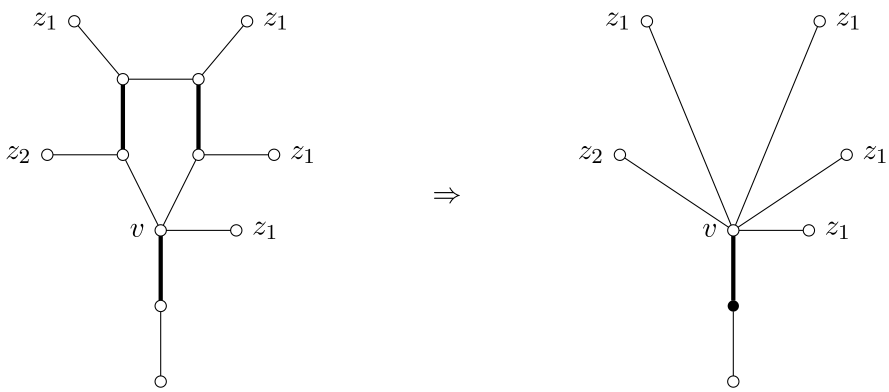


Blossom is an odd cycle on k vertices containing $\frac{k-1}{2}$ edges from M.

9: Let C be a blossom, where v is not matched with other vertex in C. Show that alternating path entering C using a matching edge e containing v can leave C using unmatched edge from any vertex of C.



Since blossom acts like a vertex that can be matched to anything, we contract the blossom.



Edmonds Blossom Algorithm (sketch)

1. $M = \emptyset$
2. $F =$ uncovered vertices, all edges unexplored
3. while exists unexplored edge e adjacent to outer vertex of F
4. if e connects two outer vertices from different components of F,
5. get M-alternating path and augment M, go to 2.
6. if e connects two outer vertices from the same components of F:
7. find a blossom and contract it
8. if e connects and outer vertex to unexplored node x,
9. add x and its neighbor in M to F.

If implemented carefully, runs in $O(\sqrt{n}m)$, where $n = |V|$ and $m = |E|$.