Edmonds' Blossom Algorithm (Mid-Exam of AI2615)

June 2, 2022

In this problem, we together develop Jack Edmonds' celebrated "blossom" algorithm for computing a maximum matching in a (not necessarily bipartite) graph. The algorithm, invented by Edmonds in 1961, was considered as a milestone for algorithm design. It is in this work that Edmonds characterizes *feasible problem* as those "polynomial-time solvable" ones. This predates the **P** v.s **NP** problem of Stephen Cook by a decade.

Figure 1: Aviad, Tao and Jack at SJTU campus.

Recall that a matching in a simple graph $G = (V, E)$ is a collection of edges $M \subseteq E$ such that no two edges in M share a vertex. Formally, $\forall e_1, e_2 \in M, e_1 \neq e_2 \implies e_1 \cap e_2 = \emptyset$. Throughout this problem, we assume that G is given as an input with $|V| = n$ and $|E| = m$.

Problem 1

Let M be a matching. An M-*alternating path* is a simple path P in G such that the edges of P are alternatively in M and not in M. An M-*augmenting path* is an M-alternating path whose both ending vertices are not covered by M. Prove that M is a maximum matching of G if and only if no M-augmenting path exists.

Hint: Suppose M is not maximum. Let M' be a maximum matching such that $|M \cap M'|$ is maximized. How does $M \cup M'$ look like?

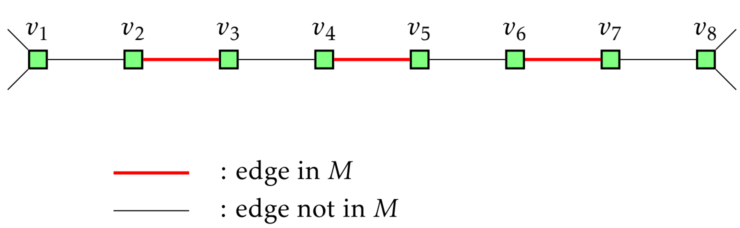

Figure 2: The path $v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8$ is an M-augmenting path.

Problem 2

Let M be a matching of G and $C \subseteq E$ be a cycle in G with $2k + 1$ vertices for some $k \geq 1$. Assume C contains exactly k edges in M and it meets no other edges in M. Let G' be the graph obtained from G by contracting C. Prove that M is a maximum matching of G if and only if $M \setminus C$ is a maximum matching of G'.

As shown in the figure, by contracting a cycle C in G we mean the following operations: (1) remove all vertices of C; (2) add a new vertex u; (3) for any remaining $v \in V$, connect v to u iff v is connected to any vertex of C in G.

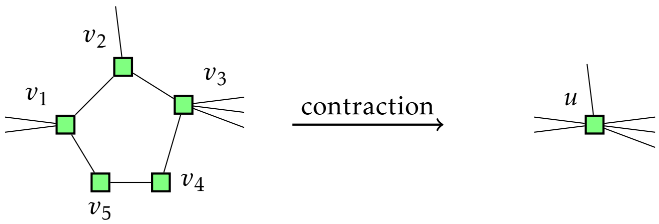

Figure 3: Contract the cycle $v_1v_2v_3v_4v_5v_1$

Problem 3

An M-alternating forest $F \subseteq E$ (shown in Figure 4) is a forest with the following properties:

1. Each component of F contains exactly one vertex not covered by M. We call it the *root* of the component.
2. If a vertex in F is at an odd distance to a root, we call it an *inner vertex*; if a vertex in F is at an even distance to a root (including root itself), we call it an *outer vertex*. Each inner vertex has two incident edges in F and exactly one of it is in M.

An M-alternating forest F is *maximal* if any $F' \supsetneq F$ is not an M-alternating forest. Prove that M-alternating forests exist and one can find a maximal one in polynomial-time.

Problem 4

Let M be a maximal matching and F be a maximal M-alternating forest. Prove that any M-augmenting path must contain two consecutive outer vertices.

Problem 5

Let M be a maximal matching and F be a maximal M-alternating forest. Assume there exist two outer vertices u and v and they are connected in G. Prove that if u and v belong to distinct components of F, then a M-augmenting path exists; and if u and v belong to the same component of F, then an odd cycle exists in G.

The cycle is called a "blossom", growing on a tree in the forest.

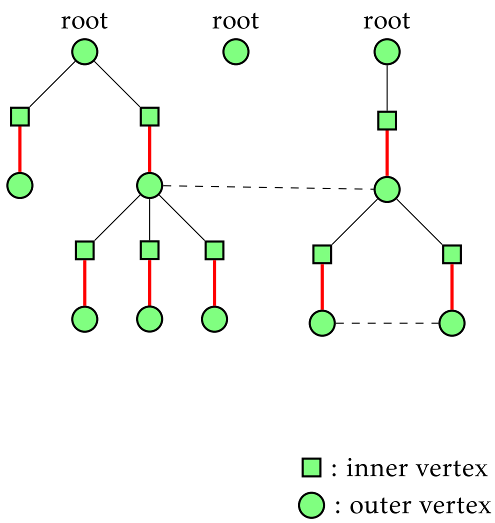

Figure 4: A M-alternating forest. Dashed edges are those edges in $E \setminus F$.

Problem 6

Based on previous observations, design a polynomial-time algorithm to find a maximum matching in G. Prove the correctness of your algorithm. What is the complexity of your algorithm?