Winter term 2014/15
Professor Dr. Stephan Held
Jannik Silvanus

Research Institute for Discrete Mathematics
University of Bonn

Combinatorial Optimization

Exercise Sheet 4

Exercise 4.1: Let $G = (V, E)$ be a graph. A set $\mathcal{H} = \{S_1, \dots, S_k, v_1, \dots v_r\}$ has property A if

- $S_i \subseteq V$ and $|S_i|$ is odd for $1 \leq i \leq k$,
- $v_i \in V$ for $1 \leq i \leq r$, and
- for each $e \in E$ either $e \subseteq S_i$ for some $i \in \{1, \dots, k\}$ or $v_i \in e$ for some $i \in \{1, \dots, r\}$.

The weight of a set $\mathcal{H}$ with property A is $w(\mathcal{H}) := r + \sum_{i=1}^{k} \frac{|S_i|-1}{2}$. Prove

$$\nu(G) = \min\{w(\mathcal{H}) \,|\, \mathcal{H} \text{ is a set with property A}\}.$$

(4 Points)

Exercise 4.2: Let G be a graph and M a matching in G that is not maximum.

(i) Show that there are $\nu(G) - |M|$ vertex-disjoint M-augmenting paths in G.
Hint: Recall the proof of Berge's Theorem.

(ii) Prove that there exists an M-augmenting path of length at most $\frac{\nu(G)+|M|}{\nu(G)-|M|}$.

(iii) Let P be a shortest M-augmenting path in G and P' an $(M \triangle E(P))$-augmenting path. Prove $|E(P')| \geq |E(P)| + 2|E(P \cap P')|$.

Consider the following algorithm: We start with the empty matching and in each iteration augment the matching along a shortest augmenting path. Let $P_1, P_2, \dots$ be the sequence of augmenting paths chosen.

(iv) Show that if $|E(P_i)| = |E(P_j)|$ for $i \neq j$, then P_i and P_j are vertex-disjoint.

(v) Conclude that the sequence $|E(P_1)|, |E(P_2)|, \dots$ contains at most $2\sqrt{\nu(G)} + 2$ different numbers.

From now on, let G be bipartite and set $n := |V(G)|$ and $m := |E(G)|$.

(vi) Prove that, given a matching M in G, the union of all shortest M-augmenting paths in G can be found in $\mathcal{O}(m+n)$ time.
Hint: Use a variant of breath-first search.

(vii) Consider a sequence of iterations of the algorithm where the length of the augmenting path remains constant. Show that the time needed for the whole sequence is no more than $\mathcal{O}(m+n)$.
Hint: Use (vi) and apply a variant of depth-first search.

(viii) Describe an algorithm with runtime $\mathcal{O}(\sqrt{n}(m+n))$ that solves the CARDINALITY MATCHING PROBLEM in bipartite graphs.

Note that the final steps are also possible for non-bipartite graphs, yet more complicated. (12 Points)

Deadline: Tuesday, November 4, 2014, before the lecture.
Information: Submissions by groups of one or two students are allowed.