University of Konstanz
Department of Computer & Information Science
Sabine Cornelsen / Jan Christoph Athenstädt

Combinatorial Optimization
Summer 2014

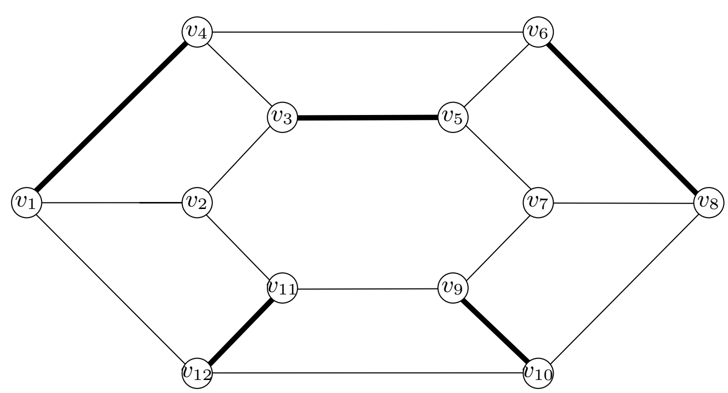
Assignment 9

Post Date: 20 June 2014 **Due Date:** 27 June 2014 **Tutorial:** 2 July 2014
You are permitted and encouraged to work in groups of two.

Problem 1: Alternating and Augmenting Paths 5 Points

Find the following in the graph below. Matched edges are drawn thicker.

- **(a)** an alternating path of length 10
- **(b)** an alternating circle of length 10
- **(c)** an augmenting path of length 5
- **(d)** an augmenting path of length 9
- **(d)** an augmenting tree with v_2 as root



Please turn over

Problem 2: Max Weighted Matching **7 Points**

Given a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, we want to find a matching M, with maximum weight $w(M) := \sum_{e \in M} w(e)$.

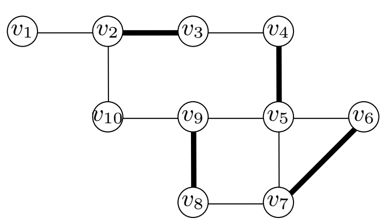
Consider the following transformation of an instance of max weighted matching into an instance of min perfect matching:

We construct a graph $G^* = (V^*, E^*)$ from G as follows. For every node $v \in V$, we create two nodes v' and v'' in V^*. For every edge $e = \{u, v\} \in E$, we create two edges in E^*: one edge $e' = \{u', v'\}$ and one edge $e'' = \{u'', v''\}$. We negate the weights for those edges, i.e. $w(e') = w(e'') = -w(e)$. Now G^* contains two copies of G with negated edge-weights. In a last step, for every node $v \in V$, we add an edge $e_v = \{v', v''\}$ to E^* and set its weight to 0.

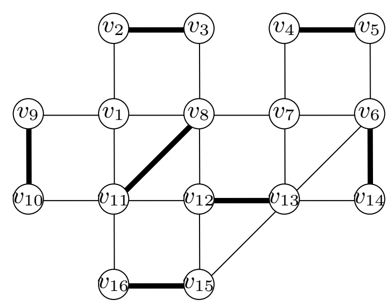
Show that a minimum perfect matching M^* in G^* induces a maximum weighted matching M in G and vice versa.

Problem 3: Blossom Algorithm **8 Points**

Apply Edmonds' blossom algorithm to augment the given matchings in the graphs below. Use v_1 as starting node on your search for blossoms. Draw the network after each contraction-step.



(a)



(b)