Combinatorial Optimization – Problem Set 5

You can hand in one of the following problems at the start of Tuesday's problem session. Please explain your solution carefully. Don't forget to put your name.

Bipartite Matchings

1. *Show that the maximum weight matching problem and the maximum weight perfect matching problem are equivalent, in the sense that if you have a polynomial algorithm for one, then you also have a polynomial algorithm for the other.*

   Given a graph G in which to find a maximum weight perfect matching, let $K = 1 + \sum_{e\in E} w_e$, and define $w'_e = w_e + K$. Now find a maximum weight matching in G with weights w'_e. Because of the size of K, this will be a matching with maximum cardinality, and maximum weight among those with maximum cardinality. If there is a perfect matching, this will be one.

   Given a graph G in which to find a maximum weight matching, define a new graph as follows. Make a copy G' of G, but set all its weights to 0. Then make a new graph H consisting of G and G', with also an edge between any $v \in V(G)$ and its copy $v' \in V(G')$; give such an edge weight 0. Find a maximum weight perfect matching in H, which certainly exists because the edges vv' form a perfect matching (albeit of weight 0). This maximum weight perfect matching will consist of a maximum weight matching in G, together with other edges of weight 0.

2. *Show in two ways that if a bipartite graph is k-regular (every vertex has degree $k \geq 1$), then it has a perfect matching: once using linear programming, and once using König's Theorem.*

   - First note that $|A| = |B|$, since

   $$|E| = k|A| = \sum_{a\in A} \deg(a) = \sum_{b\in B} \deg(b) = k|B|.$$

   - Using König: We observe that any vertex cover C has at least $|A|$ vertices, since each vertex of C covers exactly k of the $k|A|$ edges. Then König tells us that a maximum matching has at least $|A|$ edges, which is only possible if it has exactly $|A| = |B|$ edges, so is perfect.

   - Using LP: Summing up all the dual constraints $y_u + y_v \geq 1$ gives $\sum_{uv\in E}(y_u + y_v) \geq |E|$. Then

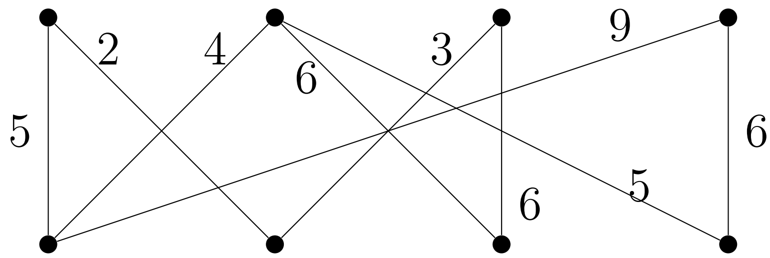
   $$k \cdot |A| = |E| \leq \sum_{uv\in E} (y_u + y_v) = k \cdot \sum_{v\in V} y_v,$$

   where the last equality holds because each y_v occurs in k different uv.
   So we have $\sum_{v\in V} y_v \geq |A|$, which implies that the dual minimum is $\geq |A|$, hence so is the primal maximum. There must be an integral optimum primal solution by the theorem from the lecture, which must have $x_e \in \{0, 1\}$, so that will correspond to a perfect matching.

3. *Prove complementary slackness for bipartite maximum weight perfect matchings (Lemma 4.4 in the notes): If we have a primal feasible x and a dual feasible y such that for all $e \in E(G)$ either $x_e = 0$ or $y_a + y_b = w_{ab}$, then both are optimal.*

   By duality we have
$$\sum_{e \in E} w_e x_e \leq \sum_{v \in V} y_v$$
   for any primal feasible x and dual feasible y. We show that the complementary slackness property gives equality, which implies that in that case x and y are optimal.
   For every edge $e = uv$ we have $w_e x_e = (y_u + y_v) x_e$, so
$$\sum_{e \in E} w_e x_e = \sum_{uv \in E} (y_u + y_v) x_{uv} = \sum_{v \in V} \left(\sum_{u : uv \in E} x_{uv} \right) y_v = \sum_{v \in V} y_v.$$

4. *Execute the primal-dual algorithm for maximum weight perfect matchings for the graph below. Give proof that the resulting matching is indeed maximum.*
   You can write it out however you like, but make sure that the different objects in every step are clear.
   This it too annoying to latex. The maximum weight is 22, uniquely achieved by the matching with weights 2,5,6,9.



5. *Let G be a graph, not necessarily bipartite. Prove that a matching M in G is maximum if and only if there is no augmenting path for M. (a path between two unmatched vertices that alternates M-edges and non-M-edges).*
   So why can't we use the augmenting path algorithm for nonbipartite graphs?

   If M is maximum there cannot be an augmenting path for M, because then there would be a larger matching.
   If M is not maximum, let M' be a larger matching, so $|M'| > |M|$. Consider H with $V(H) = V(G)$ and $E(H) = E(M) \cup E(M')$. Its vertices can only have degree 0,1, or 2, which means it consists of paths and cycles (we've seen the argument/algorithm for that before). These paths and cycles must be alternating (i.e. alternately use edges from M and M'), otherwise one of the matchings would have two touching edges. Alternating cycles have as many M-edges as M'-edges, so there must be a path with more M'-edges than M-edges, which means that it is augmenting for M. Done.

   The augmenting path algorithm doesn't work like this for nonbipartite graphs, because it's not clear how to find an augmenting path. We can't use the same directed graph trick, because there is no given direction that you could orient the matching edges in (like "from B to A").
   The problem is that without this trick we don't have an algorithm for finding alternating paths from an unmatched vertex to another unmatched one. You could just try all alternating paths from a vertex, maybe with depth-first search, but this could take exponential time. The answer to this will come in the next lecture.