Massachusetts Institute of Technology
18.433: Combinatorial Optimization
Michel X. Goemans

Handout 11
October 18th, 2005

Problem set 4

This problem set is due in class on October 27th, 2005.

1. (From p. 59 of book.) We are given n jobs that each take one unit of processing time. All jobs are available at time 0, and job j has a profit of c_j and a deadline d_j. The profit for job j will only be earned if the job completes by time d_j. The problem is to find an ordering of the jobs that maximizes the total profit. First, prove that if a subset of the jobs can be completed on time, then they can also be completed on time if they are scheduled in the order of their deadlines. Now, let $E(M) = \{1, 2, \cdots, n\}$ and let $\mathcal{I}(M) = \{J \subseteq E(M) : J \text{ can be completed on time }\}$. Prove that M is a matroid and describe how to find an optimal ordering for the jobs.

2. Recall the spanning tree game from lecture 1. It is a 2-player game played on a given graph $G = (V, E)$. Each player in turn selects an edge. Player 1 starts by deleting an edge, and then player 2 fixes an edge (which has not been deleted yet); an edge fixed cannot be deleted later on by the other player. Player 2 wins if he succeeds in constructing a spanning tree of the graph; otherwise, player 1 wins.

   In this problem, you need to describe a winning strategy for player 2 *if* the graph contains two edge-disjoint spanning trees. (This is actually an *if and only if* condition; if G does not have two edge-disjoint spanning trees, then player 1 has a winning strategy. You do not need to show the other direction.)

   HINT. If player 1 first selects edge e_1 and player 2 first selects edge f_1, player 2 would then need in subsequent rounds to select edges F such that $F \cup \{f_1\}$ forms a spanning tree. This means that F forms a spanning tree in the graph G' obtained from G by first deleting e_1 and then contracting f_1 (this is denoted by $G \setminus \{e_1\}/\{f_1\}$); contracting means identifying both endpoints of f_1. But, by induction, we know when player 2 has a winning strategy in G'. To be continued...

3. Consider a graph $G = (V, E)$. Let $E(M) = E$ and $\mathcal{I}(M) = \{F_1 \cup F_2 : F_1, F_2 \text{ are forests in } G\}$.

   (a) Show that M is a matroid.

   (b) This implies that the greedy algorithm can find the maximum weight set of edges which can be partitioned into two forests. However, implementing the greedy algorithm is not at all easy (yet), as it is unclear how to test if a given set F of edges is independent (i.e. can be partitioned into 2 forests). [If you have time to kill (!) or are curious, try the analogous question for linear matroids: given a matrix A, when can the columns be partitioned into two sets of linearly independent columns?]

Give an example which shows that the following algorithm does *not* find a maximumw weight collection of edges which can be partitioned into 2 forests: find a maximum weight forest F in G, delete the edges of F, and find again a maximum weight forest in the remaining graph.

(c) If we consider the union of k forests as the independent sets, would the resulting independence system still be a matroid?

4. Given a family $A_1, A_2, \cdots, A_n$ of sets (they are not necessarily disjoint), a *transversal* is a set T such that $T = \{a_1, a_2, \cdots, a_n\}$, the a_i's are distinct, and $a_i \in A_i$ for all i. A partial transversal is a transversal for $A_{i_1}, A_{i_2}, \cdots, A_{i_k}$ for some subfamily of the A_i's.

   Show that the family of all partial transversals forms a matroid (on the ground set $E = \cup A_i$).

   Hint: Think of bipartite matchings.