Problem set 6

This problem set is due in class on May 6th, 2015. You can skip one of the questions, of your choosing.

1. Given a matroid $M = (E, \mathcal{I})$ with rank function r, define $\mathcal{I}^* = \{F \subset E | r(E \setminus F) = r(E)\}$, i.e. $F \in \mathcal{I}^*$ if the complement of F still contains a base.

   (a) Show that $M^* = (E, \mathcal{I}^*)$ is a matroid (it is called the dual matroid). What are the bases of M^* (in terms of teh bases of M)?

   (b) Let r^* be the rank function of M^*. Give an expression for $r^*(U)$ (for any $U \subseteq E$) in terms of the rank function r of M.

2. Consider the minimum cost arborescence problem we discussed in lecture. Given a digraph $D = (V, A)$, a root vertex $r \in V$, costs $c : A \to \mathbb{R}_+$, the problem is to find an r-arborescence of minimum total cost, where an r-arborescence is a spanning tree (when the directions of the arcs are discarded) in which all vertices are reachable from r by a directed path. Consider the following greedy algorithm (generalizing the undirected setting):

   Let $F = \emptyset$, $S = V \setminus \{r\}$
   while $S \neq \emptyset$ **do**
   Among all arcs in $\bigcup_{v \in S} \delta^-(v)$ whose addition to F does not create a directed circuit, let (u, v) be the one of minimum cost
   $F = F \cup \{a\}$
   $S = S \setminus \{v\}$
   end while
   Return F

   Show that this greedy type algorithm does not necessarily return the optimum arborescence.

3. In lecture and in the notes, we have seen an algorithm to compute the minimum cost r-arborescence for a given vertex r (in a directed graoh D with costs c). Suppose we would like to find the minimum cost r-arborescence for *every* vertex $r \in V$. We could use the algorithm seen in class $|V|$ times, but show how to modify it to be able to return a minimum cost r-arborescence for every r (in time much less than running the algorithm $|V|$ times).

   Hint: The modified algorithm should first construct a strongly connected graph F (independently of r) such that it contains an optimum r-arborescence for every r and such that this optimum r-arborescence can be obtained by a reverse delete procedure (as in the original algorithm).

4. Let A be an invertible $n \times n$ matrix (thus $\text{rank}(A) = n$). Let $[n] = \{1, 2, \cdots, n\}$ denote the indices of rows and columns of A. For any subset $X \subseteq [n]$ and any subset $Y \subseteq [n]$, let $A_{X,Y}$ denote the submatrix with rows indexed by X and columns indexed by Y. The following linear algebra/matrix theory result can be derived in several ways:

   Theorem 0.1 *Let A be an invertible $n \times n$ matrix. For any partition of the set of columns into $[n] = Y_1 \cup Y_2$, there exists a partition of the set of rows into $[n] = X_1 \cup X_2$ such that A_{X_1,Y_1} and A_{X_2,Y_2} are invertible.*

   Prove this theorem by deriving it from the minmax relation for matroid intersection. (First question to ask yourself is what would be the two matroids to use in this case; one option (which you do not have to necessarily follow) is to take $M_1 = ([n], \mathcal{I}_1)$ and $M_2 = ([n], \mathcal{I}_2)$ where $\mathcal{I}_1 = \{I : A_{I,Y_1} \text{ has rank } |I|\}$ and $\mathcal{I}_2^* = \{I : A_{I,Y_2} \text{ has rank } |I|\}$ where the * denotes the dual operation in exercise 1.)

5. While discussing the matroid polytope, we derived that the spanning tree polytope of a graph $G = (V, E)$ is given by (read again the notes on matroid optimization):

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(E) = |V| - 1 & \\ x(E(S)) \leq |S| - 1 & S \subset V \\ 0 \leq x_e & e \in E\}, \end{array}$$

   where as usual $x(F) = \sum_{e \in F} x_e$.

   If we wanted to use the ellipsoid to optimize[1] over P, we would need to solve the separation problem: Given x, is $x \in P$ and if not, provide a valid inequality for P violated by x. We can easily check whether $x(E) = |V| - 1$ so in what follows we assume that the given x already satisfies $x(E) = |V| - 1$. We are now going to show that the separation problem can be solved by solving $|V|$ maximum flow problems. Consider the following directed graph $D = (V \cup \{s, t\}, A)$. A has three types of arcs:

   - For any $v \in V$, D has an arc (s, v) with capacity $u((s, v)) = x(\delta(v))$,
   - for any $v \in V$, D has an arc (v, t) with capacity $u((v, t)) = 2$,
   - for any $(u, v) \in E$, D has the arcs (u, v) and (v, u) both of capacity x_{uv}.

   (a) In D, what is the capacity of the cut induced by $\{s\}$ and by $\{s\} \cup V$?

   (b) Show that there exists a violated inequality for x (satisfying $x(E) = |V| - 1$) if and only if the value of the following cut problem is less than $2|V|$:

$$\min_{S \subseteq V, S \neq \emptyset} u(\{s\} \cup S).$$

   (Notice that we are not allowing the cut to be induced by $\{s\}$.)

[1] One could say this would be inefficient since we can simply use the greedy algorithm over P; however, if we have additional constraints then we couldn't use the greedy algorithm and instead separate both over P and over these additional constraints.

(c) Show that the separation problem can be solved by $|V|$ maximum flow problems (in $|V|$ networks obtained by slightly modifying D).