**Spring 2011, CS 598CSC: Approximation Algorithms**

**Homework 4**
Due: 04/13/2011 in class

**Instructions and Policy:** Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.

Solve as many problems as you can. I expect at least three.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary. Your job is to convince me that you know the solution, as quickly as possible.

**Problem 1** Recall the congestion minimization problem in directed graphs that we discussed in lecture. We discussed a variant in which the path chosen for each pair $(s_i, t_i)$ has to have at most $h$ edges where $h$ is a given parameter. We discussed a path-based LP relaxation with an exponential number of variables but a polynomial number of constraints and how the dual of the LP can be solved via the ellipsoid method. In this problem we will consider writing a polynomial-sized primal formulation via flow variables and how it suffices to solve a slightly relaxed problem.

- Write an LP relaxation using flow variables $f(e, i)$ where $f(e, i)$ is the flow for pair $(s_i, t_i)$ on edge $e$ (assume the input graph is directed). To enforce the constraint that the number of edges used in a path for $(s_i, t_i)$ is at most $h$ write a a total cost constraint on the flow for each pair.
- Let $\lambda$ be the optimum congestion for the relaxation above. Use flow-decomposition and Markov's inequality to show that a feasible solution to the above LP can be used to obtain a feasible and polynomial-sized solution for the path-based formulation such that the length of each path is at most $2h$ and congestion of the solution is at most $2\lambda$. More generally, argue that for any fixed $\epsilon > 0$, the paths can be chosen to be of length at most $(1 + \epsilon)h$ with the congestion value at most $\lambda/\epsilon$.

**Problem 2** Problem 7.3 from Shmoys-Williamson book.

**Problem 3** Problem 7.5 from Shmoys-Williamson book.

**Problem 4** Recall that in the Generalized Steiner Network Problem (also called Survivable Network Design Problem), the input is an undirected graph $G = (V, E)$ with edge costs $c : E \to R^+$, and a *requirement* $r_{uv}$ for every (unordered) pair of vertices $u, v \in V(G)$;

the goal is to find a minimum-cost set of edges $E'$ such that for each $u, v$, there are $r_{uv}$ edge-disjoint paths between $u$ and $v$ in $E'$.

In class, we saw a cut-based Linear Program for this problem with an exponential number of constraints. Give a polynomial-sized flow-based LP formulation. (Though the input graph is undirected, you will need to create an appropriate directed graph for your LP.)

**Problem 5** Prove that each of the following classes of functions is skew supermodular:

1. Proper functions
2. Downward monotone functions
3. The residual functions of skew supermodular functions.

Can you think of an example of a residual function of a proper function that is not proper? *Hint:* Consider the Steiner network problem with connectivity at most 2.

**Problem 6** (Harder) Let $G = (V, E)$ be an undirected graph and let $f$ be an integer valued requirement function (not necessarily a $\{0, 1\}$ function) on the vertex set. Recall that the primal-dual algorithms require the ability to do answer the following questions. Given $F \subseteq E$, is $F$ a feasible solution for $f$? Given $F \subseteq E$ what are the minimal violated sets with respect to $F$? Also recall that if $f$ is skew-supermodular (or proper) the minimial violated sets are disjoint.

1. Suppose $f$ is a proper function and it is accessible as an oracle which when given a set $S \subset V$ returns the value $f(S)$; such an oracle is called a value oracle. Show that there is a polynomial time algorithm to determine if $F$ is a feasible solution.
   *Hint:* Consider the cuts in the Gomory-Hu tree $T$ for the graph $G[F]$.
2. Now suppose $f$ is a skew-supermodular function. We do not know a polynomial time algorithm to test if $F$ is a feasible solution by simply using the value oracle for $f$. However, suppose you have an oracle that given $F \subseteq E$ returns whether $F$ is feasible or not. Show how you can use such an oracle to compute in polynomial time the minimal violated sets with respect to a collection of edges $A$. First prove that if $S \subset V$ is a *maximal* set such that $A \cup \{(i, j) : i, j \in S\}$ is not feasible then $V \setminus S$ is a minimal violated set for $A$. Then deduce that the set of minimial violated sets can be obtained by less than $|V|^2$ calls to the feasibility oracle.