Fall 2013, CS 583: Approximation Algorithms

Homework 3
Due: 10/21/2013

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 four.

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 Let $G = (V, A)$ be a directed graph with arc weights $c : A \to \mathcal{R}^+$. Define the density of a directed cycle C as $\sum_{a \in C} c(a)/|V(C)|$ where $V(C)$ is set of vertices in C.

1. A cycle with the minimum density is called a minimum mean cycle and such a cycle can be computed in polynomial time. How?

   Hint 1: Given density λ, give a polynomial-time algorithm to test if G contains a cycle of density $< \lambda$. Now use binary search.

   Hint 2: There is a polynomial time algorithm to detect if a graph has a negative cycle (a cycle with sum of arc lengths negative).

2. Consider the following algorithm for ATSP. Given G (with c satisfying asymmetric triangle inequality), compute a minimum mean cycle C. Pick an arbitrary vertex v from C and recurse on the graph $G' = G[V - C \cup \{v\}]$. A solution to the problem on G can be computed by patching C with a tour in the graph G'. Prove that the approximation ratio for this heuristic is at most $2H_n$ where $H_n = 1 + 1/2 + \ldots + 1/n$ is the nth harmonic number.

Problem 2 For Metric-TSP consider the nearest neighbour heuristic discussed in class. Prove that the heuristic yields an $O(\log n)$ approximation. (Hint: use the basic idea in the online greedy algorithm for the Steiner tree problem from Lecture 1 (Spring 2011)). **Extra Credit:** Give an example to show that there is no constant c such that the heuristic is a c-approximation algorithm.

Problem 3 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 λ 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 2λ. 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 $(1+\epsilon)\lambda/\epsilon$.

Problem 4 Consider the set cover problem and the randomized rounding that we discussed in class. Here we consider a small variant. Recall the LP relaxation. There is a variable x_j for each set S_j in the given instance and the constraints ensure that $\sum_{j:i\in S_j} x_j \geq 1$ for each element $i \in \mathcal{U}$.

Given a feasible solution $\bar{x}$ define a new solution $\bar{y}$ where $y_j = \min\{1, (2\ln n)x_i\}$. We then independently pick each S_j with probability y_j. The chosen sets may not form a set cover in that some elements may not be covered. To obtain a feasible set cover we do the following: for each uncovered element i we simply pick the cheapest set that contains it and add it to the solution. Clearly, the algorithm outputs a feasible set cover. Prove that the expected cost of the set cover output by this algorithm is $O(lnn)\sum_j c_j x_j$. In other words this gives, with high probability, a randomized $O(\ln n)$ approximation for set cover.

Hint: Use Chernoff bounds to bound the probability that an element is not covered in the first step, and that the solution satisifies the given bound with high probability.

Problem 5 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 6 Problem 7.3 from Shmoys-Williamson book.

Problem 7 Problem 7.5 from Shmoys-Williamson book.

Problem 8 Problem 9.2 from Shmoys-Williamson book. However, there is a slight typo in the data given for the problem. Assume, that for all facilities, the opening cost is 2. The distances c_{ij} remain as given in the problem.