CS 573: Graduate Algorithms, Fall 2011
HW 5 (due in class on Tuesday, November 29th)

This homework contains five problems. **Read the instructions for submitting homework on the course web page**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: For this home work students can work in groups of up to three students each. Only one copy of the homework is to be submitted for each group. Make sure to list all the names/netids clearly on each page.

Note on Proofs: Details are important in proofs but so is conciseness. Striking a good balance between them is a skill that is very useful to develop, especially at the graduate level.

1. (20 pts) Consider the following scheduling problem. There are n unit-sized jobs that need to be scheduled on m identical machines. Each job j has a release time r_j and a deadline d_j both of which are integers. A schedule assigns each job to a time slot $[t, t+1]$ for some integer t and to some machine such that no two jobs are assigned to the same slot on the same machine. If job j is completed by its deadline there is no penalty; if it completes at time $C_j > d_j$ then it incurs a penalty of $(C_j - d_j)$. A job cannot be schedule before it is release time. The goal is find a schedule for the jobs on the given machines so as to minimize the total penalty. Describe a polynomial time algorithm for this problem. Be sure to pay attention to the fact that your algorithm runs in polynomial time by examining carefully the size of the input.

2. (20 pts) Reduce the following two problems to the problem of computing a minimum-cost perfect matching in a bipartite graph.

   - Given an undirected graph $G = (V, E)$ and edge costs $c : E \to \mathbb{R}$ find the minimum-cost 2-matching in G. A 2-matching in G is an assignment $x : E \to \mathbb{Z}_+$ such that $x(\delta(v)) = 2$ for each v.
   - Given a directed graph $G = (V, E)$ and edge costs $c : E \to \mathbb{R}$ find a minimum-cost set of cycles that partition the node set V; that is these cycles are node-disjoint and together contain all the nodes. The cost of a cycle is the sum of the edges of the cycle.

3. (20 pts) Let $G = (V, E)$ be an undirected graph with non-negative edge lengths $\ell : E \to R^+$. Given nodes s, t we wish to find a shortest s-t path with an odd number of edges. Show that this problem can be solved via matching techniques by following the hint below. Suppose s, t have degree 1. Consider the reduction of the maximum weight matching problem to the maximum weight perfect matching problem. Adapt this reduction and show how a perfect matching can be used to obtain a shortest odd length path from s to t. How can you get rid of the assumption that s, t have degree 1? Extend the ideas for finding a shortest even length s-t path.

4. (20 pts) Flow is typically defined as a function on the edges but path-based flow definition and formulations are useful and necessary in various applications. We illustrate some relevant

issues via this problem. Let $D = (V, A)$ be a directed graph with non-negative capacities $c : A \to R^+$. Given nodes s, t let $P_{s,t}$ denote the set of all simple paths between s and t.

- Write the maximum s-t flow problem as a linear programming problem with one variable for each path $p \in P_{s,t}$. Note that the primal can have an exponential (in $|V|$) number of variables. Write its dual.
- What is the separation problem for the dual? Show that there is a polynomial time algorithm for the separation problem for the dual. Via the Ellipsoid method, this implies that you can solve the dual to optimality.
- Now consider the following problem. You want to find the maximum s-t flow when restricted to paths of length at most k where k is a given integer. Write this is a large linear program. Show that the separation oracle for the dual is polynomial time solvable.

5. (20 pts) The facility location problem is the following. There is a set of n facilities F and m clients C. The cost of openining a facility i is f_i. There is a cost $c(i, j)$ to connect client j to facility i. The goal is to open a susbet of the facilities and connect each client to an open facility. The objective function is to minimize the sum of the facility opening costs and the connection costs.

   - Write an integer linear programming formulation for this problem using two sets of decision variables, one set for opening facilities, and one set for assigning clients to open facilities. Prove that it is sufficient to constrain only the facility opening variables to be integer valued. Such a problem is called a mixed-integer programming problem.
   - Show that an α-approximation for the facility location problem implies an α-approximation for the set cover problem.

Questions to ponder:

- Do you understand the optimality condition for minimum-cost flow?
- Did you read up on the dynamic programming based algorithm for finding the minimum-mean cycle?
- Reduce the problem of finding a maximum weight perfect matching in a graph (the weights can be negative or positive) to the problem of finding a maximum weight matching (Does it make sense to have negative weights for this problem?). *Hint:* Create two copies of the given graph and a perfect matching between the corresponding nodes.
- Show via network flow and matching ideas that the minimum-cost vertex cover problem can be solved in bipartite graphs. Recall that we showed that this problem in NP-Hard in general graphs.
- An edge cover in an undirected graph $G = (V, E)$ is a subset of edges $E' \subseteq E$ such that each node is incident to some edge in E'. In vertex cover we cover edges by nodes while in edge cover we cover nodes by edges. Unlike vertex cover, egde cover is polynomial-time solvable due its connections to matchings. Show that in any graph G we have $\rho(G) + \nu(G) = |V|$ were $\rho(G)$ is the size of a minimum edge-cover and $\nu(G)$ is the size of a maximum matching. One

can find a minimum-cost edge cover via an algorithm for maximum weight matching. Here is a hint. Given edge weighted $c : E \to \mathbb{R}$ consider new edge weights $c' : E \to \mathbb{R}$ as follows. First, for any node v let $\alpha(v) = \min\{c(e) \mid e \in \delta(v)\}$. Now let $c'(uv) = \alpha(u) + \alpha(v) - c(uv)$ for each edge $uv \in E$.

- T-joins are intimately related to matchings and have many applications. One interesting application is to find negative length cycles and shortest paths in *undirected* graph when edges can have negative length; recall that the directed graph algorithms do not work. You can look up the notes on T-joins in Chandra's course on combinatorial optimization.

- Consider a system of linear inequation defined by $Ax = b$ where A is a $n \times n$ matrix and b is a n-dimensional vector. If the inverse A^{-1} exists then system has a unique solution x^* given by $A^{-1}b$. Using the standard definition of the formula for the inverse of a matrix and definition of the determinant using the formula $\det(A) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)}$, show that the number of bits needed to represent x^* is polynomial in n and the number of bits needed to represent the numbers in A and b. Do you see how this allows you to prove that the decision version of linear programming is in both NP and co-NP (which requires duality).

- Given a polyhedron $Ax \leq b$ and a point z can you check in polynomial time whether z is a vertex of the polyhedron?

- Given a vertex z of a polyhedron and a basis how can you check whether z is a local optimum or not?

- Do you understand how to find a starting vertex for running the Simplex algorithm by solving another linear program? To see whether you understand this consider solving the following problem: given a system $Ax < b$ with strict inequalities give an algorithm to decide if there is a feasible solution for this system.

- We say a proof sketch of one variant of Farkas lemma in class: $Ax \leq b$ is not feasible if and onle if there is a $y \geq 0$ such that $yA = 0$ and $yb < 0$. Using this prove that $Ax = b, x \geq 0$ is not feasible iff there is y such that $yA \geq 0$ and $yb < 0$ (we saw the geometric intuition for this).

- Suppose we have a linear program $\max cx, A_1x \leq b_1, A_2x = b_2, A_3x \geq b_3, x \geq 0$. What is its dual?