CS 473g: Algorithms, Fall 2007

Homework 5 (due Tuesday, November 27, 2007 at 11:59.99 p.m.)

Version 1.01

Name:		
Net ID:	Alias:	

Name:		
Net ID:	Alias:	

#	Score	Grader
1.		
2.		
3.		
4.		
5.		
6.		
7.		
Total		

Neatly print your name(s), NetID(s), and the alias(es) you used for Homework 0 in the boxes above. Staple this sheet to the top of your homework. If you are on campus, submit the homework by submitting it in SC 3306 (or sliding it under the door).

Note: You will be held accountable for the appropriate responses for answers (e.g. give models, proofs, analysis, etc). For NP-Complete problems you should prove everything rigorously, i.e. for showing that it is in NP, give a description of a certificate and a polynomial time algorithm to verify it, and for showing problems are NP-Hard, you must show that your reduction is polynomial time (by similarly proving something about the algorithm that does the transformation) and proving both directions of the 'if and only if' (a solution of one is a solution of the other) of the many-one reduction.

Finally: It was stated at the outset, that this system would not be here, and at once, perfected. You cannot but plainly see that I have kept my word. But I now leave my cetological System standing thus unfinished, even as the great Cathedral of Cologne was left, with the crane still standing upon the top of the uncompleted tower. For small erections may be finished by their first architects; grand ones, true ones, ever leave the copestone to posterity. God keep me from ever completing anything. This whole book is but a draft - nay, but the draft of a draft. Oh, Time, Strength, Cash, and Patience! – Moby Dick, Herman Melville

Required Problems

1. Slack form
   [10 Points]

   Let L be a linear program given in slack form, with n nonbasic variables N, and m basic variables B. Let N' and B' be a different partition of $N \cup B$, such that $|N' \cup B'| = |N \cup B|$. Show a polynomial time algorithm that computes an equivalent slack form that has N' as the nonbasic variables and b' as the basic variables. How fast is your algorithm?

2. Tedious Computations
   [20 Points]

   Provide *detailed* solutions for the following problems, showing each pivoting stage separately.

   (a) **[5 Points]**
   maximize $6x_1 + 8x_2 + 5x_3 + 9x_4$
   subject to
   $2x_1 + x_2 + x_3 + 3x_4 \leq 5$
   $x_1 + 3x_2 + x_3 + 2x_4 \leq 3$
   $x_1, x_2, x_3, x_4 \geq 0.$

   (b) **[5 Points]**
   maximize $2x_1 + x_2$
   subject to
   $2x_1 + x_2 \leq 4$
   $2x_1 + 3x_2 \leq 3$
   $4x_1 + x_2 \leq 5$
   $x_1 + 5x_2 \leq 1$
   $x_1, x_2 \geq 0.$

   (c) **[5 Points]**
   maximize $6x_1 + 8x_2 + 5x_3 + 9x_4$
   subject to
   $x_1 + x_2 + x_3 + x_4 = 1$
   $x_1, x_2, x_3, x_4 \geq 0.$

   (d) **[5 Points]**
   minimize $x_{12} + 8x_{13} + 9x_{14} + 2x_{23} + 7x_{24} + 3x_{34}$
   subject to
   $x_{12} + x_{13} + x_{14} \geq 1$
   $-x_{12} + x_{23} + x_{24} = 0$
   $-x_{13} - x_{23} + x_{34} = 0$

$x_{14} + x_{24} + x_{34} \leq 1$
$x_{12}, x_{13}, \ldots, x_{34} \geq 0.$

3. LINEAR PROGRAMMING FOR A GRAPH
[10 Points]

(a) **[3 Points]** Given a weighted, directed graph $G = (V, E)$, with weight function $w : E \rightarrow \mathcal{R}$ mapping edges to real-valued weights, a source vertex s, and a destination vertex t. Show how to compute the value $d[t]$, which is the weight of a shortest path from s to t, by linear programming.

(b) **[4 Points]**
Given a graph G as in (a), write a linear program to compute $d[v]$, which is the shortest-path weight from s to v, for each vertex $v \in V$.

(c) **[4 Points]**
In the *minimum-cost multicommodity-flow problem*, we are given a directed graph $G = (V, E)$, in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$ and a cost $\alpha(u, v)$. As in the multicommodity-flow problem (Chapter 29.2, CLRS), we are given k different commodities, K_1, K_2, $\ldots$, K_k, where commodity i is specified by the triple $K_i = (s_i, t_i, d_i)$. Here s_i is the source of commodity i, t_i is the sink of commodity i, and d_i is the demand, which is the desired flow value for commodity i from s_i to t_i. We define a flow for commodity i, denoted by f_i, (so that $f_i(u, v)$ is the flow of commodity i from vertex u to vertex v) to be a real-valued function that satisfies the flow-conservation, skew-symmetry, and capacity constraints. We now define $f(u, v)$, the ***aggregate flow*** to be sum of the various commodity flows, so that $f(u, v) = \sum_{i=1}^{k} f_i(u, v)\alpha(u, v)$. The aggregate flow on edge (u, v) must be no more than the capacity of edge (u, v).
The cost of a flow is $\sum_{u,v \in V} f(u, v)$, and the goal is to find the feasible flow of minimum cost. Express this problem as a linear program.

4. LINEAR PROGRAMMING
[20 Points]

(a) **[10 Points]** Show the following problem in NP-hard.

Problem: Integer Linear Programming

> *Instance:* A linear program in standard form, in which A and B contain only integers.
> *Question:* Is there a solution for the linear program, in which the x must take integer values?

(b) **[5 Points]** A steel company must decide how to allocate next week's time on a rolling mill, which is a machine that takes unfinished slabs of steel as input and produce either of two semi-finished products: bands and coils. The mill's two products come off the rolling line at different rates:

Bands 200 tons/hr
Coils 140 tons/hr.

They also produce different profits:

Bands \$ 25/ton
Coils \$ 30/ton.

Based on current booked orders, the following upper bounds are placed on the amount of each product to produce:

Bands 6000 tons
Coils 4000 tons.

Given that there are 40 hours of production time available this week, the problem is to decide how many tons of bands and how many tons of coils should be produced to yield the greatest profit. Formulate this problem as a linear programming problem. Can you solve this problem by inspection?

(c) **[5 Points]** A small airline, Ivy Air, flies between three cities: Ithaca (a small town in upstate New York), Newark (an eyesore in beautiful New Jersey), and Boston (a yuppie town in Massachusetts). They offer several flights but, for this problem, let us focus on the Friday afternoon flight that departs from Ithaca, stops in Newark, and continues to Boston. There are three types of passengers:

i. Those traveling from Ithaca to Newark (god only knows why).
ii. Those traveling from Newark to Boston (a very good idea).
iii. Those traveling from Ithaca to Boston (it depends on who you know).

The aircraft is a small commuter plane that seats 30 passengers. The airline offers three fare classes:

i. Y class: full coach.
ii. B class: nonrefundable.
iii. M class: nonrefundable, 3-week advanced purchase.

Ticket prices, which are largely determined by external influences (i.e., competitors), have been set and advertised as follows:

	Ithaca-Newark	Newark-Boston	Ithaca-Boston
Y	300	160	360
B	220	130	280
M	100	80	140

Based on past experience, demand forecasters at Ivy Air have determined the following upper bounds on the number of potential customers in each of the 9 possible origin-destination/fare-class combinations:

	Ithaca-Newark	Newark-Boston	Ithaca-Boston
Y	4	8	3
B	8	13	10
M	22	20	18

The goal is to decide how many tickets from each of the 9 origin/destination/fare-class combinations to sell. The constraints are that the place cannot be overbooked on either the two legs of the flight and that the number of tickets made available cannot exceed the forecasted maximum demand. The objective is to maximize the revenue. Formulate this problem as a linear programming problem.

5. Distinguishing between probabilities

[10 Points] Suppose that Y is a random variable taking on one of the n know values:

$$a_1, a_2, \ldots, a_n.$$

Suppose we know that Y either has distribution p given by

$$\mathcal{P}(Y = a_j) = p_j$$

or it has distribution q given by

$$\mathcal{P}(Y = a_j) = q_j.$$

Of course, the numbers $p_j, j = 1, 2, \ldots, n$ are nonnegative and sum to one. The same is true for the q_j's. Based on a single observation of Y, we wish to guess whether it has distribution p or distribution q. That is, for each possible outcome a_j, we will assert with probability x_j that the distribution is p and with probability $1-x_j$ that the distribution is q. We wish to determine the probabilities $x_j, j = 1, 2, \ldots, n$, such that the probability of saying the distribution is p when in fact it is q has probability no larger than β, where β is some small positive value (such as 0.05). Furthermore, given this constraint, we wish to maximize the probability that we say the distribution is p when in fact it is p. Formulate this maximization problem as a linear programming problem.

6. SOME DUALITY REQUIRED.
[10 Points]

(a) **[3 Points]** What is the dual of the following LP?

$$\begin{aligned} \text{maximize} \quad & x_1 - 2x_2 \\ \text{subject to} \quad & x_1 + 2x_2 - x_3 + x_4 \geq 0 \\ & 4x_1 + 3x_2 + 4x_3 - 2x_4 \leq 3 \\ & -x_1 - x_2 + 2x_3 + x_4 = 1 \\ & x_2, x_3 \geq 0 \end{aligned}$$

(b) **[4 Points]** Solve the above LP in detail, providing the state of the LP after each pivot step. What is the value of the target function of your LP?

(c) **[4 Points]** Solve the dual of the above LP in detail, providing the state of the LP after each pivot step.

7. STRONG DUALITY.
[20 Points]

Consider a directed graph G with source vertex s and target vertex t and associated costs $\mathsf{c}_\cdot \geq 0$ on the edges. Let $\mathcal{P}$ denote the set of all the directed (simple) paths from s to t in G.

Consider the following (very large) integer program:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E(G)} \mathsf{c}_e x_e \\ \text{subject to} \quad & x_e \in \{0, 1\} && \forall e \in E(G) \\ & \sum_{e \in \pi} x_e \geq 1 && \forall \pi \in \mathcal{P}. \end{aligned}$$

(A) **[5 Points]** What does this IP computes?

(B) **[5 Points]** Write down the relaxation of this IP into a linear program.

(C) **[5 Points]** Write down the dual of the LP from (B). What is the interpretation of this new LP? What is it computing for the graph G (prove your answer)?

(D) **[5 Points]** The strong duality theorem states the following.

> **Theorem 0.1** *If the primal LP problem has an optimal solution $x^* = (x_1^*, \ldots, x_n^*)$ then the dual also has an optimal solution, $y^* = (y_1^*, \ldots, y_m^*)$, such that*
>
> $$\sum_j c_j x_j^* = \sum_i b_i y_i^*.$$

In the context of (A)-(C) what result is implied by this theorem if we apply it to the primal LP and its dual above? (For this, you can assume that the optimal solution to the LP of (B) is integral – which is not quite true – things are slightly more complicated than that.)