CS 170 Algorithms
Fall 2014 David Wagner

HW 10

Due Nov. 14, 6:00pm

Instructions. This homework is due Friday, November 14th, at 6:00pm electronically. Same rules as for prior homeworks. See `http://www-inst.cs.berkeley.edu/~cs170/fa14/hws/instruct.pdf` for the required format for writing up algorithms.

1. (25 pts.) Linear programming practice

A cargo plane can carry a maximum weight of 100 tons and a maximum volume of 60 cubic meters. There are three materials to be transported, xantalum, ytterbium, and zastatine. The cargo company may choose to carry any amount of each, up to the maximum available limits given below.

- Xantalum has density 2 tons/cubic meter, maximum available amount 40 cubic meters, and revenue \$1,000 per cubic meter.
- Ytterbium has density 1 ton/cubic meter, maximum available amount 30 cubic meters, and revenue \$1,200 per cubic meter.
- Zastatine has density 3 tons/cubic meter, maximum available amount 20 cubic meters, and revenue \$12,000 per cubic meter.

Write a linear program that optimizes revenue for a single plane flight, within the constraints. List all the variables you used and the intuitive meaning of each.

2. (40 pts.) Henron

You've been hired by Henron, a new startup with this great idea to make a killing by creating a market for chicken futures. Henron has relationships with a set of n suppliers and a set of m purchasers. The i-th supplier can supply up to $s[i]$ chickens this year, and the j-th purchaser would like to buy up to $b[j]$ chickens this year. Henron is the middleman and makes \$1 off each chicken that is sold. Thus, the more chickens that are sold, the more money you make!

However, there's a complication. Due to federal restrictions on interstate trafficking in avian life forms, supplier i can only sell chickens to a purchaser j if they are situated at most 100 miles apart. Assume that you're given a list L of all the pairs (i, j) such that supplier i is within 100 miles of purchaser j. You will be given $n, m, s[1..n], b[1..m], L$ as input. Your job will be to compute the maximum number of chickens that can be sold this year. The running time of your algorithm should be polynomial in n and m.

For parts (a) and (b), assume it's OK to sell a fraction of a chicken (don't worry if answer involves you selling a number of chickens that is not an integer).

(a) Formulate this as a network flow problem. In other words, show how to solve this problem, using a network flow algorithm as a subroutine. Show or describe the graph you have in mind, and explain why the output from the network flow algorithm gives a valid solution to this problem.

(b) Formulate this as a linear programming problem. In other words, show how to solve this problem, using a linear programming solver as a subroutine. Explain why this correctly solves the problem.

(c) Now let's assume you can't sell a fraction of a chicken. In other words, the number of chickens sold by each supplier to each purchaser must be an integer. Which formulation would be better, network flow or linear programming? Explain your answer.

3. (35 pts.) 4-coloring

The 3-COLORING problem is as follows:

Input: an undirected graph $G = (V, E)$

Output: a valid 3-coloring $c : V \to \{\text{Red}, \text{Green}, \text{Blue}\}$, or "NO" if none exists

The 4-COLORING problem is as follows:

Input: an undirected graph $G = (V, E)$

Output: a valid 4-coloring $c : V \to \{\text{Red}, \text{Green}, \text{Blue}, \text{Yellow}\}$, or "NO" if none exists

These are both search problems. (Recall that c is a valid coloring if $c(u) \neq c(v)$, for every edge $(u, v) \in E$.)

(a) Find a reduction from 3-COLORING to 4-COLORING.

(b) Prove that if there is a polynomial-time algorithm for the 4-COLORING problem, then there is a polynomial-time algorithm for 3-COLORING, too.

4. (5 pts.) Optional bonus problem

You want to analyze the flow of traffic in the city. You've modeled the city as a simple $N \times M$ grid. The grid cell at row i and column j can support a capacity of $c_{i,j}$ cars. After the workday is over, rush hour starts, and cars start at grid cell $(1, 1)$ and all simultaneously try to move to grid cell (N, M). From a single cell, a car can drive to the (at most) four cells that it shares a side with (i.e., north, east, south, or west). Each car can take a different path from $(1, 1)$ from (N, M). However, the number of cars that pass through cell (i, j) on their path must not exceed $c_{i,j}$.

Design an efficient algorithm to find the maximum number of cars that can go from cell $(1, 1)$ to cell (N, M), given the NM capacities $c_{i,j}$. Your algorithm should run in $O(N^{1.1}M^{1.1})$ time or less ($O(NMC)$, where C is the largest capacity, is too slow). Make sure you describe the graph you are using.