Homework 3 for CMSC 498U/644

Due March 27

March 25, 2018

1 Problem 1

A meat packing plant produces 480 hams, 400 pork bellies, and 230 picnic hams every day; each of these products can be sold either fresh or smoked. The total number of hams, bellies, and picnics that can be smoked during a normal working day is 420; in addition, up to 250 products can be smoked on overtime at a higher cost. The *net* profit are as follows:

	Fresh	Smoked on regular time	Smoked on overtime
Hams	\$8	\$14	\$11
Bellies	\$4	\$12	\$7
Picnics	\$4	\$13	\$9

Table 1: Profit

For example, the following schedule yields a total daily net profit of \$9,965:

	Fresh	Smoked on regular time	Smoked on overtime
Hams	165	280	35
Bellies	295	70	35
Picnics	55	70	105

Table 2: Profit

The objective is to find the schedule that maximizes the total net profit. Formulate as an LP problem in the standard form.

2 Problem 2

The *bicycle problem* involves n people who have to travel a distance of ten miles, and have on a single-seat bicycle at their disposal. The data is specified by the walking speed w_j and the bicycling speed b_j of each person j ($j = 1, 2, \ldots, n$); the task is to minimize the arrival time of the last person. Can you solve the case of $n = 3$ and $w_1 = 4$, $w_2 = w_3 = 2$, $b_1 = 16$, $b_2 = b_3 = 12$

3 Problem 3

Consider the following problem: given a set of items, such that the i-th element has a weight w_i and a value v_i, determine whether to include each item in a bag so that the total weight is less than or equal to a given limit B and the total value is as large as possible. This problem is called the Knapsack Problem (or more specifically 0-1 knapsack problem).

There are many algorithms (like dynamic programming) that solve this problem, but we are going to give linear programming(and LP relaxation) a try.

3.1 Part a

Write an integer program that solves Knapsack Problem, and give its linear relaxation.

3.2 Part b

Solve (you can use `gurobi`, or by hand) the LP you gave on the following instance: size of the bag $B = 20$, and weight of items as follows

i	Weight w_i	Value v_i
1	2	3
2	3	4
3	4	5
4	5	8
5	9	10

Table 3: Item value and weight

Note you would get a fractional solution.

3.3 Part c

There is a simple greedy algorithm for fractional knapsack problem. First sort the items in decreasing order of value per unit weight, i.e. $\frac{v_i}{w_i}$. It then insert the items one by one until there is no enough space for the current item. It then takes all the remaining space and assigin a portion of the current item. Run this algorithm on the instance in Part b, and compare it with the fractional solution you get in Part b. What did you find? Could give an explanation?

4 Problem 4

Using the data from the following table. Columns represent different sets, and rows represent possible items. Add to the signatures of the columns the values of the following hash functions:

(a) $h_3(x) = 2x + 4 \mod 5$

(b) $h_4(x) = 3x - 1 \mod 5$

Below are examples for hash functions $h_1(x) = x + 1 \mod 5$, and $h_2(x) = 3x + 1 \mod 5$.

Row	S_1	S_2	S_3	S_4	$h_1(x) = x + 1 \mod 5$	$h_2(x) = 3x + 1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

	S_1	S_2	S_3	S_4
$h_1()$	1	3	0	1
$h_2()$	0	2	0	0