Tufts University
Department of Mathematics
Math 87 Homework 3

Due: Thursday, October 3, at 10:30 a.m. (in class).

1. For the following linear programs, first sketch the feasible region and, then, find the values of x and y where the objective function obtains its maximum value. What is the maximum function value?

   (a) (20 points)

   $$\begin{aligned} \text{Maximize } & 3x + y \\ \text{subject to } & 2x + y \leq 6 \\ & x + 3y \leq 9 \\ & x, y \geq 0 \end{aligned}$$

   (b) (20 points)

   $$\begin{aligned} \text{Maximize } & 7x + 3y \\ \text{subject to } & 2x + y \leq 8 \\ & x + y \geq 4 \\ & x, y \geq 0 \end{aligned}$$

   (c) (20 points)

   $$\begin{aligned} \text{Maximize } & 2x + 3y \\ \text{subject to } & x + y \geq 1 \\ & y - x \leq 3 \\ & 2x + y \leq 9 \\ & x - y \leq 3 \\ & x, y \geq 0 \end{aligned}$$

2. (10 points) Find a division of a cake between two people that is fair and envy-free, but neither pareto-optimal nor equitable.

3. (30 points) Download the file `cake_division.m` from the course webpage; this file takes as input a "taste matrix", T, and returns a "division matrix", D, that is both equitable and Pareto-optimal, based on the linear programming formulation discussed in class. In this problem, we will compare that approach with a "brute force" approach that tests a large number of possibilities.

   Note that the following code generates a random $M \times N$ taste matrix, T:

```
% Set M,N
M = 3;
N = 2;

% generate random taste matrix
```

```
T = rand(M,N);
for i=1:M,
  T(i,:) = T(i,:)/sum(T(i,:));
end;
```

For the case of two people sharing a cake composed of two distinct parts, it is easy to generate a potential division, by taking

$$D = \begin{bmatrix} d_1 & d_2 \\ 1-d_1 & 1-d_2 \end{bmatrix},$$

where $0 \leq d_1 \leq 1$ and $0 \leq d_2 \leq 1$ are the amounts of the two parts of the cake that person 1 receives.

Write an algorithm that loops over reasonable sets of values of d_1 and d_2, to find a division that is nearly equitable and nearly Pareto-optimal. Your algorithm should take a taste matrix as input, as well as two parameters, one to control how many values of d_1 and d_2 are sampled, and one to control how close v_{11} and v_{22} should be to consider a division "nearly" equitable. Your algorithm should return the division matrix that corresponds to the maximum average value of v_{11} and v_{22} among nearly equitable divisions.

Compare the performance of your code and the code given in `cake_division.m` for finding division matrices for random taste matrices. Be sure to comment on how small you need to take the parameters described above in order to generate "good" results, and be sure to average over several runs.