Diego Escobedo April 24, 2020
6.046 Problem Set 7
Collaborators: *Temi Omitoogun*

Problem 1

(A) We want to find:

$$\underset{x_A, x_B}{\text{argmax}}\ u_A + u_B$$
$$\underset{x_A, x_B}{\text{argmax}}\ x_A(1 - x_A - x_B) + x_B(1 - x_A - x_B)$$
$$\underset{x_A, x_B}{\text{argmax}}\ x_A + x_B - x_A^2 - 2x_Ax_B - x_B^2$$

To find the max, we take the derivative with respect to x_A and x_B respectively, and set them to 0.

$$\frac{\partial f}{\partial x_A} = 1 - 2x_A - 2x_B \qquad \text{We can isolate } x_A \text{ and plug into the next equation}$$
$$x_A = \frac{1}{2} - x_B$$
$$\frac{\partial f}{\partial x_B} = 1 - 2x_B - 2x_A \qquad \text{Set equal to 0 and plug in } x_A$$
$$0 = 1 - 2x_B - 2(\frac{1}{2} - x_B)$$
$$0 = 0$$

Because we have 0=0, that means that there are infinite solutions. Any combination of x_A and x_B that satisfies $x_A + x_B = \frac{1}{2}$ is going to maximize total utility. The value of the maximum total utility is $\frac{1}{4}$

(B) We were told that a Nash equilibrium is one where neither player can improve their utility by unilaterally changing their strategy. Thus, consider for Alice the problem where

x_B is fixed at $\frac{1}{3}$.

$$
\begin{aligned}
u_A &= x_A(\frac{2}{3} - x_A) \\
\frac{\partial u_A}{\partial x_A} &= \frac{2}{3} - 2x_A \\
0 &= \frac{2}{3} - 2x_A \\
2x_A &= \frac{2}{3} \\
x_A &= \frac{1}{3}
\end{aligned}
$$

The optimal strategy for Alice, therefore, is the strategy that is currently implemented. By revealing Bob's choice, she is not able to unilaterally change her own position to increase her utility. Because the situation is symmetrical, this constitutes a Nash equilibrium. The total utility achieved is $x_A + x_B - x_A^2 - 2x_Ax_B - x_B^2 = \frac{2}{3} - 4 * \frac{1}{9}$. Thus, when both players reach Nash equilibrium at $\frac{1}{3}$, the total utility is $\frac{2}{9}$.

(C) The deterministic Nash Equilibria are any pair $(x_A, x_B) s.t. x_A + x_B = 1$ and $x_A, x_B \geq 0$. The reason for this is that if they do not add up to 1, then either player could increase their utility by adding a greater fraction of the bandwith. However, they are disincentivized from accruing more bandwith past 1 because any bandwith they get would be immediately wiped out if they go over 1. Therefore, given the other player's fraction x_O, a player's optimal strategy is to apportion to themselves $1 - x_O$ of the bandwith. So, a Nash equilibrium can only be achieved if both teh players fractions add up to 1.

Problem 2

(A)
False. Observe the following set of companies/interns and their preferences

Preferences	Interns	Companies	Preferences
[C1, C2, C3]	Intern 1	Company 1	[I3, I2, I1]
[C2, C1, C3]	Intern 2	Company 2	[I3, I1, I2]
[C3, C2, C1]	Intern 3	Company 3	[I1, I2, I3]

It is easy to see by inspection that it is impossible for any intern,company pair to have each other as first ranked, since there are none that have mutually ranked each other first.

(B) Counterexample: Imagine network A has TV shows rated 0 and 2, and network B has shows rated 1 and 3. There are two possible matchings: (0-1, 2-3), which is not stable because A would benefit from switching their shows to win 1 instead of winning 0, and (0-3, 2-1), which is not stable because B would benefit from switching their shows to win 2 instead of winning 1. Thus, there is not always necessarily a stable pair of schedules.

Problem 3

(A) An instability could arise if h_α is assigned to d_i, and h_β is not given an offer (since there are more candidates than positions), but d_i preferred h_β to h_α.

(B) Instead of trying to create a new algorithm we can simply use our existing algorithm but modify the inputs to make sense for the variations of the problem.

The way the problem is told, we have n potential hires, and m divisions d_1 through d_m each with c_1 through c_m openings, respectively. We are also told that the sum of all the openings is a number $c < n$. Additionally, we know that each division has its own set of preferences of interns. So, to make this a problem that Gale-Shapley can take, we will create c_i nodes for each d_i, named $d_{i,1}, d_{i,2}, ..., d_{i,c_i}$, each with an identical set of preferences (which was the original preferences of the division). Thus, now instead of having a bipartite graph with m on one side and n on the other, we have one with c on one side and n on the other.

To make the two sets even, we are going to create $n-c$ 'ghost' divisions (labeled $g_1, g_2, ..., g_{n-c}$, each with their own arbitrary preferences of the interns. One other thing we need to do is adjust the preferences of the interns. They originally ranked m divisions, but now they need to rank n things. The way it will work is that we will replace their ranking d_i with all the copies of its openings $d_{i,1}, d_{i,2}, ..., d_{i,c_i}$ in an arbitrary ordering. At the end of every intern's preference list is all the ghost divisions in an arbitrary ordering (since they obviously would rather get a job than not).

So, this should finally allow us to use Gale Shapley on this bipartite graph with equal set size. Because we are using an algorithm identical to the one in lecture, we refer to that proof of termination. Now, it remains to show why it is stable. We know that Gale Shapley will give us a stable matching. But what does this mean? It means that there is no division d_i that prefers a different intern AND that intern prefers d_i to its current division. In the context of the instabilities discussed in part a, that means that even if a ghost division wants another intern more, the intern (because the ghost divisions are at the end of their preference list) would rather have their current job than not have one. Further, if an intern gets matched to a ghost division, that means they are ranked lower on every division's intern preferences than the one they got matched with, meaning no division wants to trade with them and thus making the matchings stable.

In terms of the number of offers made, we will be making c offers, as we have more than enough potential hires to fill all our positions.