# Homework 7

## Joni Vrapi

## 12/5/2022

**Statement of Integrity:** I, Joni Vrapi, attempted to answer each question honestly and to the best of my abilities. I cited any, and all, help that I received in completing this assignment.

**Problem 1.** decision variables are the variables you need to make a decision on - eg how many bowls/mugs do you need to produce to have the most revenue

**Problem 2a.** If we let $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$, then the expected loss for Player 1 can be calculated as $\sum xAy^T$ resulting in:

$$x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \tag{1}$$

**Problem 2b.** Player 2 playing any strategy other than $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ will allow Player 1 to adjust his strategy to play off of Player 2's strategy, resulting in an expected gain for Player 1. For example, assume Player 2 plays $(1, 0, 0)$. Player 1 would then adjust his strategy to play $(0, 1, 0)$ resulting in an expected gain of 1 (from equation 1). Likewise, it is obvious that regardless of what strategy Player 2 chooses, other than $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, Player 1 will always be able to play off of it towards an expected gain.

**Problem 2c.** Letting $x = y = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ results in an expected outcome per (1) of 0. This result is indicative of both players winning the same amount of times, on average, resulting in a draw. The sticking point here, however, is that neither player has an incentive to change strategies. If one player were to choose an imbalanced strategy, the other player could alter their strategy to take advantage of it resulting in an expected gain for the other player. Therefore, we have reached a Nash equilibrium.

**Problem 2d.** No, it is not possible for some mixed strategy where $y' \neq (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ to form a Nash equilibrium because any mixed strategy where rock, paper, or scissors are played in an imbalanced way will allow for the other player to play a strategy where the balance was reversed to accumulate more wins. For example, if Player 2 were to choose slightly more rock than paper or scissors, Player 1 could change their strategy to favor paper by the same amount, resulting in an expected gain.

**Problem 3a.** From [1] we choose:

FACILITY-LOCATION()
1   with $R = U$ and $T = \emptyset$
2   **while** $R$ is not empty
3       $c = min_{u \in R, s \in T} d_{us}$
4       select $s \in R\ T$ and set $U_s \subset R$ that minimizes: $c' = \frac{f_s + \sum_{u \in U_s} d_{us}}{|U_s|}$
5
6       **if** $c' \leq c$
7           select the site s and set $U_s$ used to obtain $c'$ above
8           add $s$ to $T$ and delete $U_s$ from $R$
9           set $c_u = c'$ for all $u \in U_s$
10
11      **else**
12          select $s$ and $u$ obtaining the first minimum
13          add $u$ to $U_s$
14          set $c_u = c$

**Problem 3b.** From [1] we consider an optimal solution which contains a subset $T^*$ of sites, and $s \in T^*$ is used to cover a set $U_s^*$ of users. The cost of using $s$ to cover $U_s^*$ is

$$C = f_s + \sum_{u \in U_s^*} d_{us} \tag{2}$$

We will need to compare $C$ with the cost of our greedy algorithm which is $\sum_{u \in U_s^*} c_u$

**Lemma:**

$$\sum_{u \in U_s^*} c_u \leq \rho(d)(f_s + \sum_{u \in U_s^*} d_{us}) = \rho(d) \cdot C \tag{3}$$

where $d = |U_s^*|$.

The algorithm covers nodes $u_1, u_2 ... u_d$ in $U_s^*$ in order. If we consider when the algorithm covers the $i$th node, there are two cases:

- Case 1: $s \notin T$

    This implies that the cost $c_u$ is at most the cost of selecting site $s$ with set $U_s^* \cap R$ which is $c_{u_i} \leq \frac{C}{d-i+1}$

- Case 2: $s \in T$

    This implies that $c_{u_i} \leq d_{us}$

If we assume that Case 1 applies for the first $k$ nodes, and after that Case 2 applies, then we can sum all the costs in $U_s^*$

$$\sum_{u \in U_s^*} c_u \leq \frac{C}{d} + \frac{C}{d-1} + ... + \frac{C}{d-k+1} + \sum_{i>d} d_{u_i}, s \tag{4}$$

If $d = k$ then the upper bound on the cost is in fact $\rho(d) \cdot C$. If $k < d$ then the costs $\sum_{u \in U_s^*} d_{us}$ is bounded by $C$ so it is also $\rho(d) \cdot C$.

Finally, if we let $T^*$ and $U_s^*$ be the optimal solution, the total cost is

$$\sum_{s \in T^*} (f_s + \sum_{u \in U_s^*}) = \sum_{s \in T^*} \cdot C \tag{5}$$

The lemma bounds each term of the cost by $\rho(d)$ proving that this is a $\rho(n)$-approximate algorithm.

**Problem 3c.** Using the lemma to bound each term of (5) to get the optimum, we arrive at

$$OPT - \sum_{s \in T^*} (f_s + \sum_{u \in U_s^*}) \leq \sum_{s \in T^*} \rho(n) \sum_{u \in U_s^*} c_u - \rho(n) \sum_{u \in U} c_u \tag{6}$$

which would return a solution within a factor of $O(log(max_i |U_s^*|))$

# References

[1] "Lecture 9." `https://cs.gmu.edu/~lifei/teaching/cs684spring17/lec9.pdf`. Accessed on 2022-11-25.