Assignment 1

Mohammad Mahdi Rahimi (20208244)
mahi@kaist.ac.kr

September 30, 2020

Exercise 1

(a) Formulation of Game
Players are P1 and P2

$$n = 2$$

.

Action for each Player define a Real number in

$$A = \{a | a \in [0, 100]\}$$

Payoff Function for Players:

$$U_{P1} = \begin{cases} a_1 & \text{if } a_1 + a_2 \leq 100 \text{ or } a_1 < a_2 \\ 100 - a_2 & \text{else if } a_1 > a_2 \\ 50 & \text{otherwise} \end{cases} \quad U_{P2} = \begin{cases} a_2 & \text{if } a_1 + a_2 \leq 100 \text{ or } a_2 < a_1 \\ 100 - a_1 & \text{else if } a_2 > a_1 \\ 50 & \text{otherwise} \end{cases}$$

(b) Proof (50, 50) is a Nash Equilibrium

Assume we are currently in (50, 50) situation and this is not a Nash, so there is exist a player that can improve its own payoff by changing action. As situation is symmetry regards to players, we assume P1 will change the action.

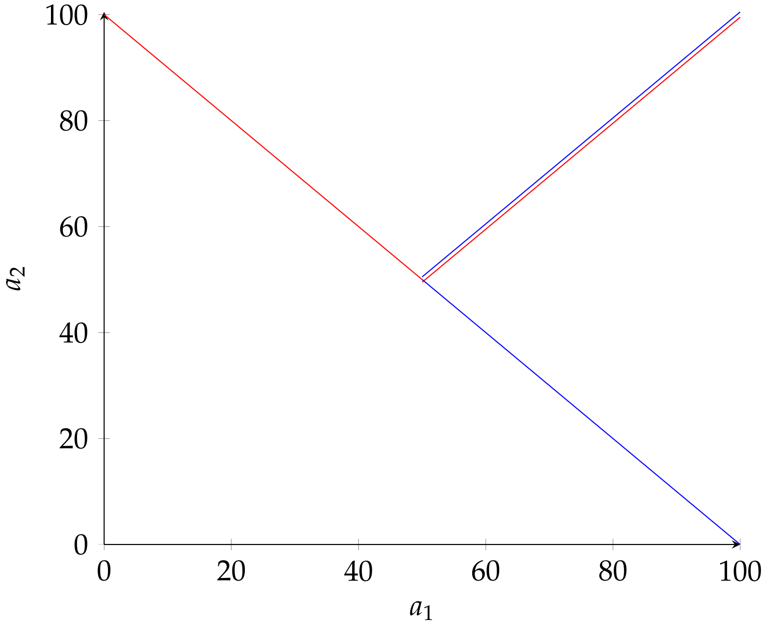
If P1 increase the amount, it leads to situation where summation of amount pass 100 limit and P1 receive 50 again.

If P1 decrease the amount, it leads to situation where summation of amount not passed 100 limit and P1 receive less than 50.

In either way nor P1 or P2 can increase the payoff by changing actions, so it's Nash equilibrium.

(c) Proof (50, 50) is the Only Nash Equilibrium We can show Nash Equilibrium as points both players best-response actions meet. The formula is as following:

$$BR_{P1} = \begin{cases} 100 - a_2 & \text{if } a_2 \leq 50 \\ a_2 - \epsilon & \text{otherwise} \end{cases} \quad BR_{P2} = \begin{cases} 100 - a_1 & \text{if } a_1 \leq 50 \\ a_1 - \epsilon & \text{otherwise} \end{cases}$$



The Only Point that two Best-Response meet as on (50, 50).

Exercise 2

(a) Formulate of Game

$N = 4000$(number of Player)

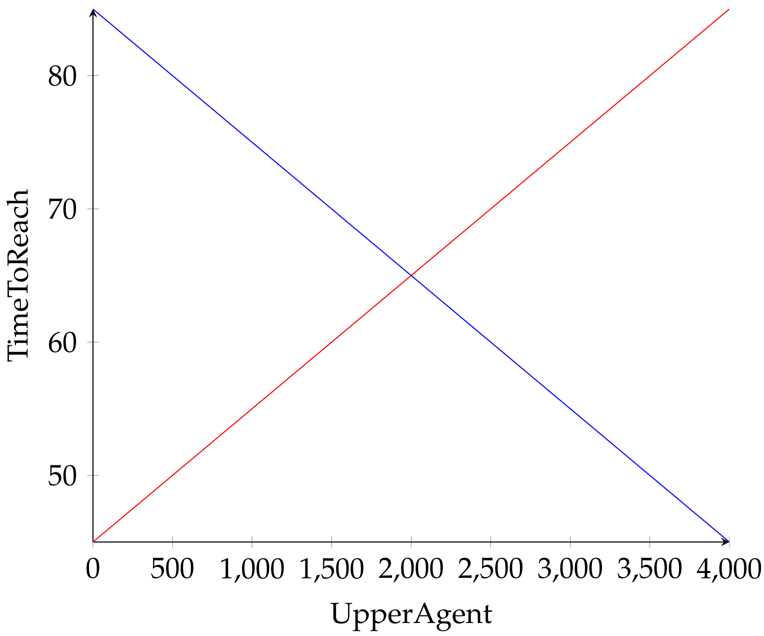
$A = \{lower, upper\} a_i$: Action Agent i

A_{-i} : Action Agents other than i

$$U_x = \begin{cases} \frac{X}{100} + 45 & \text{if Choose Upper} \\ 45 + \frac{4000-x}{100} & \text{otherwise} \end{cases}$$

(b) Proof (X = 2000) is the Only Nash Equilibrium

Nash Happen when equal people choose upper and lower road. X = 2000.
We show in X = 2000 state, neither upper road agents nor lower road agents wish to change their decision.
In state X = 2000 cost of road for all agents equals to 65 minutes any agent who changes the road will increases it's own cost by 0.01 minutes so it's not better score and nothing will change.



Red is upper agents and blue is lower agents.
The Only Point that two Best-Response meet as on (2000, 2000).

(c)

$$U_x = \begin{cases} [1].\ \frac{x}{100} + 45 & \text{if x Agent choose Upper; } X \leq 4000 \\ [2].\ 40 + \frac{k}{100} & \text{if k Agent from X choose C-D; } K \leq X \\ [3].\ 45 + \frac{4000-x+k}{100} & \text{otherwise} \end{cases}$$

$$\begin{aligned} K \leq X &\Rightarrow 40 + \frac{k}{100} < \frac{x}{100} + 45 \Rightarrow [2] < [1] \\ &\Rightarrow \text{Every Agent In Upper Road can decrease the cost by choosing the C-D,} \\ K = X &\Rightarrow 45 + \frac{4000 - x + k}{1000} = 85 > 40 + \frac{k}{100} \\ &\Rightarrow \text{Every Agent In Lower Road can decrease the cost by choosing the C-D,} \\ &\Rightarrow \text{Eventually All agent choose C-D road and X = K = 4000} \end{aligned} \tag{1}$$

The Nash happen when all agents take A-C-D-B road and cost is 80.
Surprisingly adding a new road with cost 0 moved Nash from 65 to 80.

Exercise 3

(a) 1. $(B, R, F) \rightarrow (0, 0, 0)$

(b) $Max(Min(U_1)) \Rightarrow Max([0, -1]) \Rightarrow [0]$

(c)

	L	R
T	4.4, 4.4	1.4, 5.4
B	1.4, 5.4	2.4, 2.4

(d) 1. $(B, R) \rightarrow (2.4, 2.4)$

Exercise 4

(a) Find Nash Equilibrium

n : Number of Player

q_i : Quantity of produce from i-th Firm

Q : Action for all Firms

Q^{-i} : Action Firms without i-th

Finding Best Response for i-th Agent by assuming fix action for the rest:

$$U_i(q_1, ..., q_n) = U_i(q_i) = \begin{cases} (a - Q^{-i} + q_i)q_i - cq_i & \text{if } q_i \leq a - Q^{-i} \\ -cq_i & \text{if } q_i > a - Q^{-i} \end{cases} \tag{2}$$

$$K = a - Q^{-1} - c \tag{3}$$

$$\begin{aligned} BR_i(q_i) = argmax(U_i(q_i)) &= \begin{cases} argmax(q_i(K - q_i)) & \text{if } q_i \leq K + c \\ argmax(-cq_i) & \text{if } q_i > K + c \end{cases} \\ &= \begin{cases} \frac{K}{2} & \text{if } q_i \leq K + c \\ 0 & \text{if } q_i > K + c \end{cases} \end{aligned} \tag{4}$$

Assume we are in situation $q_i > K + c$ and everyone will follow BR and set quantity to 0 then we have:

$$\begin{aligned} 0 &> K + c \\ 0 &> a - Q^{-1} - c + c \\ 0 &> a \end{aligned} \tag{5}$$

It means when $'a'$ is less than zero, all firms should stop producing.

Now, Assume we are in situation $q_i \leq K + c$ and everyone will follow BR and set quantity to $\frac{K}{2}$ then we have:

$$\begin{aligned} Q &= n(\frac{K}{2}) \\ K &= a - Q^{-i} - c \\ K &= a - (n-1)(\frac{K}{2}) - c \\ (n+1)K &= 2(a - c) \\ K &= \frac{2(a-c)}{(n+1)} \end{aligned} \tag{6}$$

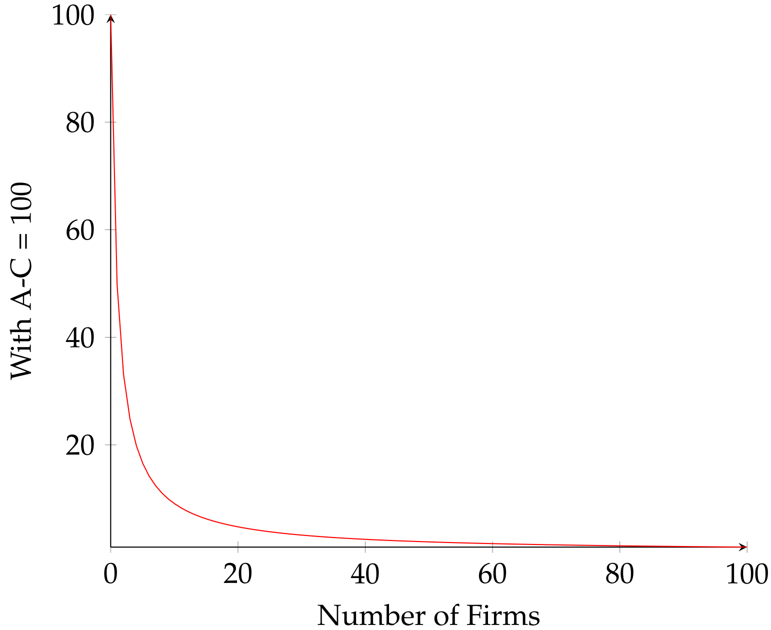
This is hold when $q_i \leq K + c$ it means:

$$\begin{aligned} \frac{K}{2} &\leq K + c \\ 0 &\leq \frac{K}{2} + c \\ 0 &\leq \frac{(a-c)}{(n+1)} + c \end{aligned} \tag{7}$$

It means this Best-Response hold when $'c'$ is not negative.
In a reasonable world both 'a' and 'c' are not negative.
Final Price if Firms use BR follow this equation: $P = \frac{K}{2} = \frac{a-c}{n+1}$

(b) Change in Price by adding new Firm

Price follow this equation: $P = \frac{a-c}{n+1}$, It is plotted by assuming 'a - c' equals to 100, for n moving between 0 to 100.



Exercise 5

(a) Pure Nash and Payoffs

$$\begin{aligned} (B, L) &\rightarrow (8, 2) \\ (T, R) &\rightarrow (2, 8) \end{aligned} \tag{8}$$

(b) Mixed Nash and Payoffs

$$(\frac{2}{3}B + \frac{1}{3}T, \frac{2}{3}R + \frac{1}{3}L) \tag{9}$$

$$\frac{4}{9}(B, R) + \frac{2}{9}(B, L) + \frac{2}{9}(T, R) + \frac{1}{9}(T, L) = (\frac{44}{9}, \frac{44}{9}) \tag{10}$$

(c) Correlated Equilibrium and Inequalities

General Inequalities:

$$\sum P(R = a|R_i = a_i)u_i(a_i, a_{-i}) \geq \sum P(R = a|R_i \neq a_i)u_i(a'_i, a_{-i}) \tag{11}$$

Correlated answer set are below:

$$C \Rightarrow \begin{cases} P_1 \Rightarrow \begin{cases} T : P(R = L|R_1 = T)u_1(L,T) + P(R = R|R_1 = T)u_1(R,T) \geq \\ P(R = L|R_1 = B)u_1(L,B) + P(R = R|R_1 = B)u_1(R,B) \\ B : P(R = L|R_1 = B)u_1(L,B) + P(R = R|R_1 = B)u_1(R,B) \geq \\ P(R = L|R_1 = T)u_1(L,T) + P(R = R|R_1 = T)u_1(R,T) \end{cases} \\ P_2 \Rightarrow \begin{cases} L : P(R = T|R_2 = L)u_2(T,L) + P(R = L|R_1 = B)u_2(L,B) \geq \\ P(R = R|R_2 = T)u_2(R,T) + P(R = R|R_2 = B)u_2(R,B) \\ R : P(R = T|R_2 = R)u_2(T,R) + P(R = R|R_1 = B)u_2(R,B) \geq \\ P(R = L|R_2 = T)u_2(L,T) + P(R = L|R_2 = B)u_2(L,B) \end{cases} \end{cases} \tag{12}$$

$$\begin{aligned} P_{LT}u_1(LT) + P_{RT}u_1(RT) &\geq P_{LB}u_1(LB) + P_{RB}u_1(RB) \\ P_{LB}u_1(LB) + P_{RB}u_1(RB) &\geq P_{LT}u_1(LT) + P_{RT}u_1(RT) \\ P_{TL}u_2(LT) + P_{BL}u_2(BL) &\geq P_{TR}u_2(TR) + P_{BR}u_2(BR) \\ P_{TR}u_2(TR) + P_{BR}u_2(BR) &\geq P_{TL}u_2(LT) + P_{BL}u_2(BL) \end{aligned} \tag{13}$$

$$\begin{aligned} P_{LT}6 + P_{RT}2 &\geq P_{LB}8 + P_{RB}0 \\ P_{LB}8 + P_{RB}0 &\geq P_{LT}6 + P_{RT}2 \\ P_{TL}6 + P_{BL}2 &\geq P_{TR}8 + P_{BR}0 \\ P_{TR}8 + P_{BR}0 &\geq P_{TL}6 + P_{BL}2 \end{aligned} \tag{14}$$

$$\begin{aligned} 6P_{LT} + 2P_{RT} &= 8P_{LB} \\ 6P_{TL} + 2P_{BL} &= 8P_{TR} \\ P_{TR} + P_{BR} + P_{TL} + P_{BL} &= 1 \end{aligned} \tag{15}$$

After solving equations

$$\begin{aligned} P_{LT} = P_{RT} = P_{LB} &= P_c \\ 3P_c + P_{BR} = 1 &\Rightarrow P_{BR} = 1 - 3P_c \\ u_2(P_c) = u_1(P_c) &= P_c(2 + 8 + 6) = 16P_c, P_c \in [0, 1/3] \end{aligned} \tag{16}$$

All values that satisfies above equations is a correlated equilibrium.

(d) Correlated Equilibrium with Maximum sum of payoffs

Obviously the maximum sum of payoff happen when P_c equals to $\frac{1}{3}$, which means all $P_{LT} = P_{RT} = P_{LB} = \frac{1}{3}$ and $P_{BR} = 0$

Exercise 6

(a) Proof

First we direct proof $\overline{V}(A) = \underline{V}(A)$ and $a_{i'j'}$ is Nash $\Rightarrow i' \in i_{sec}, j' \in j_{sec}$ and then use indirect proof for $\overline{V}(A) = \underline{V}(A)$ and $i^* \in i_{sec}, j^* \in j_{sec} \Rightarrow a_{i^*j^*}$ is Nash.

Assume $i^* \in i_{sec}$ and $j^* \in j_{sec}$

$$\begin{gathered} a_{i'j'}\text{is Nash} \Rightarrow \begin{cases} a_{i'j'} = \min_i a_{j'} \Rightarrow a_{i'j'} = a_{i^*j'} \\ a_{i'j'} = \max_j a_{i'} \Rightarrow a_{i'j'} = a_{i'j^*} \end{cases} \\ \overline{V}(A) = \underline{V}(A) \Rightarrow \max_j a_{i^*j} = \min_i a_{ij^*} = a_{i^*j^*} \\ \begin{cases} a_{i^*j^*} \geq a_{i^*j'} \\ a_{i^*j^*} \leq a_{i'j^*} \end{cases} \Rightarrow \begin{cases} a_{i^*j^*} \geq a_{i'j'} \\ a_{i^*j^*} \leq a_{i'j'} \end{cases} \Rightarrow a_{i^*,j^*} = a_{i'j'} \end{gathered} \tag{17}$$

$$\begin{aligned} a_{i'j'} &= a_{i^*,j^*} \\ \min_i a_{ij'} &= min_i a_{ij^*} \\ \Rightarrow \min_i a_{ij'} &\geq \min_i a_{ij} \\ \Rightarrow j\prime &\in j_{sec} \end{aligned} \tag{18}$$

$$\begin{aligned} a_{i'j'} &= a_{i^*,j^*} \\ \max_j a_{i\prime i} &= max_j a_{i^*j} \\ \Rightarrow \max_j a_{i'j} &\leq \max_j a_{ij} \\ \Rightarrow i\prime &\in i_{sec} \end{aligned} \tag{19}$$

For Proving [$\overline{V}(A) = \underline{V}(A)$ and $i^* \in i_{sec}, j^* \in j_{sec} \Rightarrow a_{i^*j^*}$ is Nash], let's assume that there are such i^* and j^* which satisfies player security strategy but are not Nash equilibrium. In this case if a Nash equilibrium exists, we assume in (i', j') then $a_{i'j'}$ should be less than $a_{i^*,j'}$ and more than $a_{i'j^*}$, Which leads to i^* and j^* are no longer a security strategies and it contrast with our primary assumptions, so (i^*, j^*) should be a Nash equilibrium.

(b) Proof

$$\begin{gathered} a_{i_1j_1} \Rightarrow \begin{cases} a_{i_1j_1} \leq min_i a_{ij_1} \Rightarrow a_{i_1j_1} \leq a_{i_2j_1} \\ a_{i_1j_1} \geq max_j a_{i_1j} \Rightarrow a_{i_1j_1} \geq a_{i_1j_2} \end{cases} \\ a_{i_2j_2} \Rightarrow \begin{cases} a_{i_2j_2} \leq min_i a_{ij_2} \Rightarrow a_{i_2j_2} \leq a_{i_1j_2} \\ a_{i_2j_2} \geq max_j a_{i_2j} \Rightarrow a_{i_2j_2} \geq a_{i_2j_1} \end{cases} \end{gathered} \tag{20}$$

$$\begin{aligned} & a_{i_1j_1} \leq a_{i_2j_1} \leq a_{i_2j_2} \leq a_{i_1j_2} \leq a_{i_1j_1} \\ & \Rightarrow a_{i_1j_1} = a_{i_2j_1} = a_{i_2j_2} = a_{i_1j_2} \\ & \Rightarrow \text{Therefore, all four points are Nash equilibrium} \end{aligned} \tag{21}$$

Exercise 7

(a) Disproof. It might fall into a loop depending of initial actions and size of the game. (For 2 by 2 it works fine)
Example is below table when non of start actions are 1.

	0	1	2
0	2 , 0	1 , 1	0 , 2
1	1 , 1	3 , 3	1 , 1
2	0 , 2	1 , 1	2 , 0

(b) New Algorithm.

Assume we have a two-player normal form game in which players has n_i actions. We can use dynamic programming approach and split the game of $n_1 * n_2$ to a 4 sub-game of $\lceil \frac{n_1}{2} \rceil * \lceil \frac{n_2}{2} \rceil$ and after solving each sub-game we might reach a Nash for each of them with two corresponding actions, then we reduce our four answers to a new $2 * 2$ game and find the Nash for $2 * 2$ with BestResponce-Algorithm which shown in last part.

```
def find_nash_final(G):
    actions = []  # Two corresponding action which leads to Nash
    value = 0.0  # value of the Nash
    # Find Nash by given algorithm using BestResponse
    return actions, value
def find_nash(G):
    n1 = len(G); n2 = len(G[0])
    if n1 < 3 or n2 < 3:
        return find_nash_final(G)
    else:
        half_n1 = floor(n1 / 2); half_n2 = floor(n2 / 2)
        # Create four sub-game
        sub_game = [
            G[:half_n1, :half_n2],
            G[:half_n1, half_n2:],
            G[half_n1:, :half_n2],
            G[half_n1:, half_n2:]
        ]
        # Find Nash of each sub-game
        actions = []; values = []
        for game in sub_game:
            act, val = find_nash(game)
            actions.append(act)
            values.append(val)
        # Create the final game with four state
        final_game = [
            [values[1], values[2]],
            [values[3], values[4]]
        ]
        # Solve the final game and return Nash Equilibrium actions and
        act, val = find_nash_final(final_game)
        return actions[act[0] * 2 + act[1]], val
```

Proof. Assume a pure Nash Equilibrium exists for this game, so it will remains a Nash in any sub-game of this game which contain that state. This algorithm have two phase of divide and merge, which will be discussed in below.

(1) In divide phase, after we recursively divide the game to sub-games with size less than 3, we have the opportunity to solve them. In this phase the general pure Nash will surely be selected as Nash of its own sub-game and send to next level. (Definition of pure Nash)

(2) In merge phase, we create a new sub-game of with four winner state of child sub-games and the general Nash from previous step will be shown in a new sub-game and continue to winning till reach the final game.

Therefore, at the end the general pure Nash will not get lost, and we find it's values and corresponding actions.

Calculation Time-Complexity
Assume player 1 has N and player 2 has M actions and $N \geq M$.

Sub Game(Case 1): $T_{nash}(0 * 0) = 0, T_{nash}(1 * 1) = 0, T_{nash}(2 * 2) = 4$
Sub Game(Case 2): $T_{nash}(0 * N) = 0, T_{nash}(1 * N) = N, T_{nash}(2 * N) = 2 * N$

Sub Game Order Worst: $O(SubGame) = O(N)$
Sub Game Order Best: $O(SubGame) = constant$

Divide: $T_{divide}(MN) = 4T_{divede}(MN/4) = 4^{\log_4 MN} * T_{subgame} = MN * T_{subgame}$

Total Order Worst: $O(Game) = O(MN) * O(N) = O(MN^2)$
Total Order Best: $O(Game) = O(MN) * constant = O(MN)$

Total Order $M = N$: $O(Game) = O(N^2) * constant = O(N^2)$