Algorithmic Game Theory – Assignment 2019/2020

$Ci = cost$

$Commuters \in \{A, B, C, D, E\}$

$Ca = 1, Cb = 3, Cc = 5, Cd = 7, Ce = 9$

$Bus \in \{\mathrm{k1}, \mathrm{k2}, \mathrm{k3}, \mathrm{k4}, \mathrm{k5}\}$

$k1 = 10, k2 = 8, k3 = 6, k4 = 4, k5 = 2$

Each player wants to minimise their cost.

Task 1

a. [7%]

Yes, there is a sequence of dominated strategies in this game; the logical choice is for each player to choose their own car.

Each car has a fixed cost $Ca = 1, Cb = 3, Cc = 5, Cd = 7, Ce = 9$ is less than the potential cost of $k1 = 10$, since we do not know if any car players will be taking the bus. Let's start with the first player $Ca = 1$ and explain how from his decision to not take the bus, a chain of events unfolds where each player acts in their own self-interest.

$Ca = 1$. This player will always take the car, as they have a fixed cost of 1. The other option is to take the bus, which even in the best-case scenario, will always be more at cost 2. Therefore, the choice is clear for Ca, in that taking the car will always maximise his utility, so it never makes sense to take the bus. His car always has the best payoff value.

$Cb = 3$. This player will also always take the car, as they have a fixed cost of 3. Cb can assume Ca will take the car, since there is no benefit of him taking the bus, and he would like to maximise his own utility. This means the lowest assumed cost of the bus is now 4, since we can also assume that $Ca\ and\ Cb$ will always take their own cars to maximise their utility.

$Cc\ =\ 5$. This player will always take the car, as they have fixed cost of 5. Cc can make the same assumption as the previous players Cb and Ca, in that those players won't take the bus (since they are maximising their own personal utility). It can be assumed this is repeated now, to the point where nobody boards the bus, because there is no reason to; each player will act in their own selfish interest. Meaning the bus cost will always 10. Cc is cost 5, so their close will also to be to take the car.

$Cd = 7$. By this same logic (previous players always maximising their own personal utility with a selfish choice), it also makes the most sense for Cd to take the car since we can now assume that even in the best case scenario of $Ce\ and\ Cd$ both taking the bus, this would result in a best case scenario of having cost 8.

$Ce\ =\ 9$. To re-iterate, the same principle applies for Ce; there is no logic to taking the bus now that the assumption can be made that everyone has taken their own car. If Ce got the bus, it would have a cost of 10, which is greater than his fixed cost of 9.

The resulting game is every player takes their own car, with nobody taking the bus.

$$Ca = 1, Cb = 3, Cc = 5, Cd = 7, Ce = 9$$

b. [8%]

The Price of Stability asks how bad the best Nash equilibrium is compared to the optimal solution.

$$PoS = \frac{value\ of\ best\ Nash\ equilibrium}{value\ of\ optimal\ solution}$$

In this case, the value of the best Nash would be the sum of each player taking their respective vehicle with each fixed cost. So, the sum of all vehicle costs for each player is equal to the value of the best Nash equilibrium. The reason this is a Nash equilibrium, is because each car player has no incentive to take the bus, and so will always choose the car strategy as this serves in their own respective best interests. The bus cost is always assumed to be 1 cost higher than what each players car's fixed cost has to offer them.

This would then be divided by the value of the optimal solution, which is the alternate scenario where over the sum of each iteration, where the players all took the bus.

$$PoS = \frac{(1+3+5+7+9)}{(2+2+2+2+2)} = \frac{(25)}{(10)} = 2.5$$

Task 2

		x_R	$1 - x_R$
		A	**B**
x_C	**X**	4	7
$1 - x_C$	**Y**	8	6

a. [3%] No, there is no pure Nash Equilibrium; there doesn't exist a strategy for either player where a row or column strictly dominates another. Dominance would give us a set rather than a point of strategies. Thus table presented is a game in its simplest form where a mixed equilibrium can be found.

b. [12%] Mixed equilibrium using indifference conditions

- Let Player 1 be the row player with probability x_R
- Let Player 2 be the column player with probability x_C

This implies once a player chooses a strategy (x_C or x_R), the chance of using their respective alternative strategy is probability $1 - x_C, x_R$

To choose the correct values for their strategies, the players need to make each other indifferent about each other's choices.

$$Player\ 1\ =$$

$$E[U_R(X)] = E[U_R(Y)]$$

$$E[U_R(X)] = 4x_C + 7\ (1 - x_C)$$

$$E[U_R(Y)] = 8x_C + 6\ (1 - x_C)$$

So…

$$E[U_R(X)]4x_C + 7\ (1 - x_C) = \ E[U_R(Y)]8x_C + 6\ (1 - x_C)$$

Equivalently…

$$-3x_C + 7 - 7x_C = 2x_C + 6 - 6x_C$$

$$-3x_C + 7 = 2x_C + 6$$

$$5x_C = 1$$

$$x_C = \frac{1}{2}$$

$$Player\ 2\ =$$

$$E[U_c(A)] = E[U_c(B)]$$

$$E[U_c(A)] = 4x_R + 8\ (1 - x_R)$$

$$E[U_c(B)] = 7x_R + 6 = 6\ (1 - x_R)$$

So…

$$E[U_c(A)] = 4x_R + 8(1 - x_R) = E[U_c(B)]7x_R + 6(1 - x_R)$$

Equivalently…

$$4x_R + 8 - 8x_R = \ 7x_R + 6 - 6x_R$$

$$-4x_R + 8 = 1x_R + 6$$

$$5x_R = 2$$

$$x_R = \frac{2}{5}$$

So all of the equivalent strategies for $A, B\ , X\ , Y$ are:

$$x_c\ (A) = \frac{\mathbf{1}}{\mathbf{5}}\ \ x_c\ (B) = 1 - x_c = \frac{\mathbf{4}}{\mathbf{5}} \quad x_R\ (X) = \frac{\mathbf{2}}{\mathbf{5}} \quad x_R\ (Y) = 1 - x_R = \frac{\mathbf{3}}{\mathbf{5}}$$

Task 3

	A	B	C
X	3,3	0,4	0,0
Y	4,0	1,1	0,0
Z	0,0	0,0	0.5,0.5

	A	B	C
X	$P_{X,A}$	$P_{X,B}$	$P_{X,C}$
Y	$P_{Y,A}$	$P_{Y,B}$	$P_{Y,C}$
Z	$P_{Z,A}$	$P_{Z,B}$	$P_{Z,C}$

Using the table above, I have created the above inequalities. Because it's a 3x3, it will result in 12 total inequalities, 6 for row, 6 for columns.

$$\frac{PXA}{PXA+PXB+PXC}(3-4)+\frac{PXB}{PXA+\ PXB+PXC}(0-1)+\frac{PXC}{PXA+PXB+PXC}+(0-0)\ \geq 0$$

$$\frac{PXA}{PXA+PXB+PXC}(3-0)+\frac{PXB}{PXA+\ PXB+PXC}(0-0)+\frac{PXC}{PXA+PXB+PXC}+(0-0.5)\ \geq 0$$

$$\frac{PYA}{PYA+PYB+PYC}(4-3)+\frac{PYB}{PYA+\ PYB+PYC}(1-0)+\frac{PYC}{PYA+PYB+PYC}+(0-0)\ \geq 0$$

$$\frac{PYA}{PYA+PYB+PYC}(4-0)+\frac{PYB}{PYA+\ PYB+PYC}(1-0)+\frac{PYC}{PYA+PYB+PYC}+(0-0.5)\ \geq 0$$

$$\frac{PZA}{PZA+PZB+PZC}(0-3)+\frac{PZB}{PZA+\ PZB+PZC}(0-0)+\frac{PZC}{PZA+PZB+PZC}+(0.5-0)\ \geq 0$$

$$\frac{PZA}{PZA+PZB+PZC}(0-4)+\frac{PZB}{PZA+\ PZB+PZC}(0-1)+\frac{PZC}{PZA+PZB+PZC}+(0.5-0)\ \geq 0$$

- All denominators in each expression are equal so they cancel out (multiply each term by the denominator)
- Converting ≥ to ≤ means changing the signs to all terms
- So, the previous inequalities (for the ROW player) are equivalent to:

$$1PXA + \ 1PXB + 0PXC + 0PYA + \ 0PYB + \ 0PYC + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$-3PXA + 0PXB + \ 0.5PXC \ + 0PYA + \ 0PYB + \ 0PYC + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$0PXA + \ 0PXB + 0PXC - 1PYA - 1PYB + \ 0PYC \ + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$0PXA + \ 0PXB + 0PXC - 4PYA - 1PYB + 0.5PYC \ + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$0PXA + \ 0PXB + 0PXC + 0PYA + 0PYB + \ 0PYC \ + 3PZA + \ 0PZB - 0.5PZC \leq 0$$

$$0PXA + \ 0PXB + 0PXC + 0PYA + 0PYB + \ 0PYC + \ 4PZA + 1PZB - 0.5PZC \leq 0$$

This gives us a matrix for the coefficients for ROW player.

$$\frac{PXA}{PXA + PYA + PZA}(3-4) + \frac{PYA}{PXA + PYA + PZA}(0-1) + \frac{PZA}{PXA + PYA + PZA} + (0-0) \ \geq 0$$

$$\frac{PXB}{PXA + PYA + PZA}(3-0) + \frac{PYB}{PXA + PYA + PZA}(0-0) + \frac{PZB}{PXA + PYA + PZA} + (0-0.5) \ \geq 0$$

$$\frac{PXB}{PYA + PYB + PYC}(4-3) + \frac{PYB}{PYA + PYB + PYC}(1-0) + \frac{PYC}{PYA + PYB + PYC} + (0-0) \ \geq 0$$

$$\frac{PXB}{PYA + PYB + PYC}(4-0) + \frac{PYB}{PYA + PYB + PYC}(1-0) + \frac{PYC}{PYA + PYB + PYC} + (0-0.5) \ \geq 0$$

$$\frac{PXC}{PZA + PZB + PZC}(0-3) + \frac{PXC}{PZA + PZB + PZC}(0-0) + \frac{PXC}{PZA + PZB + PZC} + (0.5-0) \ \geq 0$$

$$\frac{PXC}{PZA + PZB + PZC}(0-4) + \frac{PXC}{PZA + PZB + PZC}(0-1) + \frac{PXC}{PZA + PZB + PZC} + (0.5-0) \ \geq 0$$

- So the previous inequalities (for the COLUMN player) are equivalent to:

$$1PXA + \ 0PXB + 0PXC + 1PYA + \ 0PYB + \ 0PYC + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$-3PXA + 0PXB + \ 0PXC + 0PYA + \ 0PYB + \ 0PYC + \ 0.5PZA + \ 0PZB + \ 0PZC \leq 0$$

$$0PXA - \ 1PXB + 0PXC + 0PYA - 1PYB + \ 0PYC \ + \ 0PZA + \ 0PZB + \ 0PZC \leq 0$$

$$0PXA + \ -4PXB + 0PXC + 0PYA - 1PYB + 0PYC \ + \ 0PZA + \ 0.5PZB + \ 0PZC \leq 0$$

$$0PXA + \ 0PXB + 3PXC + 0PYA + 0PYB + \ 0PYC \ + 0PZA + \ 0PZB - 0.5PZC \leq 0$$

$$0PXA + \ 0PXB + 4PXC + 0PYA + 0PYB + \ 1PYC + \ 0PZA + 0PZB - 0.5PZC \leq 0$$

This gives us a matrix for the coefficients for COLUMN player.

Input into Matlab - I will use the following Matlab command:

$$[x, fval] = \ linprog(f, A, b, Aeq, beq, lb, ub)$$

f - Is a vector that specifies the coefficients of the objective function, so f is the sum of all strategies:

```
f = [-6 -4 0 -4 -2 0 0 0 -1];
```

A - Is a matrix that specifies the coefficients of the unknowns in each inequality constraint

```
 A = [[1 1 0 0 0 0 0 0 0];
[-3 0 0.5 0 0 0 0 0 0];
[0 0 0 -1 -1 0 0 0 0];
[0 0 0 -4 -1 0.5 0 0 0];
[0 0 0 0 0 0 3 0 -0.5];
[0 0 0 0 0 0 4 1 -0.5];

[1 0 0 1 0 0 0 0 0];
[-3 0 0 0 0 0 0.5 0 0];
[0 -1 0 0 -1 0 0 0 0];
[0 -4 0 0 -1 0 0 0.5 0];
[0 0 3 0 0 0 0 0 -0.5];
[0 0 4 0 0 1 0 0 -0.5]];
```

b is a vector that specifies the constant term in each inequality constraint

```
b = [0;0;0;0;0;0;0;0;0;0;0;0];
```

Aeq is a matrix that specifies the coefficients of the unknowns in each equality constraint $PXA + PXB + PXC + PYA + PYB + PYC + PZA + PZB + PZC = 1$ therefore

```
Aeq = [1 1 1 1 1 1 1 1 1];
```

beq is a vector that specifies the constant term in each equality constraint

```
beq = 1;
```

lb is a vector that specifies a lower bound on the allowed values each variable can be assigned

```
lb = [0 0 0 0 0 0 0 0 0];
```

ub is a vector that specifies an upper bound on the allowed values each variable can be assigned

```
ub = [1 1 1 1 1 1 1 1 1];
```

These inequalities with the above constraints give us the output:

```
Command Window
New to MATLAB? See resources for Getting Started.

>> disp (x)
     0
     0
     0
     0
     1
     0
     0
     0
     0

>>
```

```
>> disp(fval)
    -2

>> |
```

	A	B	C
X	0	0	0
Y	0	1	0
Z	0	0	0

Task 4

$$3\ slots$$

$$Top\ slot - CTR\ ctr_1\ = 1$$

$$Middle\ slot -\ CTR\ ctr_2 = 0.55071$$

$$Bottom\ slot - CTR\ ctr_2 = 0.4704$$

$$3\ advertisers$$

$$Advertiser\ 1\ value/bid\ per\ click = 1\ million$$

$$Advertiser\ 2\ value/bid\ per\ click = 555710$$

$$Advertiser\ 3\ value/bid\ per\ click = 470400$$

$$Payoff\ of\ bidder\ i\ in\ slot\ j\ is\ ctr_j(v_i - p_j)\ where\ p_j is\ the\ price\ charged\ per - click\ in\ slot\ j.$$

a. [4%] Optimal allocation (maximising the social welfare)

$$OPT_{SW} = (1 * 1000000) + (0.55071 * 555710) + (0.4704 * 470400) = \ 1{,}527{,}311.2141$$

b. [26%]

So, my initial Matlab code is as follows:

```
value1 = 1:100;
%value should be range 1-million but won't compute
value2 = 1:55;
%value should be range 1-555710 but won't compute
value3 = 1:47;
%value should be range 1-470400 but won't compute
ctrl = 1;
ctrl2 = 0.55071;
ctrl3 = 0.47041;
bid = 1000000;
bid2 = 555710;
bid3 = 470400;

 for n1 = 1:length(value1)
     for n2 = 1:length(value2)
         for n3 = 1:length(value3)
             temp_eq=1;
             strategy1 = n1;
             strategy2 = n2;
             strategy3 = n3;
             utility1 = 0;
             utility2 = 0;
             utility3 = 0;
            if (strategy1 > strategy2)
                if(strategy1 > strategy3)
                    temp_high = strategy1;
                    if(strategy2 > strategy3)
                        temp_second_high = strategy2;
                    else
                        temp_second_high = strategy3;
                    end
                else
                    temp_high=strategy3;
                    temp_second_high=strategy1;
                end
            elseif(strategy2>strategy3)
                temp_high=strategy2;
                if(strategy3>strategy1)
                    temp_second_high=strategy3;
                else
                    temp_second_high=strategy1;
                end
            else
                temp_high=strategy3;
                temp_second_high=strategy2;
            end
            utility1 = ctrl*(temp_high-temp_second_high);
            utility2 = ctrl2*(temp_high-temp_second_high);
            utility3 = ctrl3*(temp_high-temp_second_high);

            for temp_n1 = 1:length(value1)
                if (temp_n1 ~= strategy1)
                    if (temp_n1 > strategy2)
                        if(temp_n1 > strategy3)
                            temp_high = temp_n1;
                            if(strategy2 > strategy3)
```

```
                    temp_second_high = strategy2;
                else
                    temp_second_high = strategy3;
                end
            else
                temp_high=strategy3;
                temp_second_high=temp_n1;
            end
        elseif(strategy2>strategy3)
            temp_high=strategy2;
            if(strategy3>temp_n1)
                temp_second_high=strategy3;
            else
                temp_second_high=temp_n1;
            end
        else
            temp_high=strategy3;
            temp_second_high=strategy2;
        end
    end
    temp_utility1 = ctrl*(temp_high-temp_second_high);
    if(utility1 < temp_utility1)
        temp_eq=0;
    end
end

for temp_n2 = 1:length(value2)
    if (strategy1 > temp_n2)
        if(strategy1 > strategy3)
            temp_high = strategy1;
            if(temp_n2 > strategy3)
                temp_second_high = temp_n2;
            else
                temp_second_high = strategy3;
            end
        else
            temp_high=strategy3;
            temp_second_high=strategy1;
        end
    elseif(temp_n2>strategy3)
        temp_high=temp_n2;
        if(strategy3>strategy1)
            temp_second_high=strategy3;
        else
            temp_second_high=strategy1;
        end
    else
        temp_high=strategy3;
        temp_second_high=temp_n2;
    end
    temp_utility2 = ctrl*(temp_high-temp_second_high);
    if(utility2 < temp_utility2)
        temp_eq=0;
    end
end
```

```
            for temp_n3 = 1:length(value3)
                if (temp_n3 ~= strategy3)
                    if (strategy1 > strategy2)
                        if(strategy1 > temp_n3)
                            temp_high = strategy1;
                            if(temp_n2 > temp_n3)
                                temp_second_high = strategy2;
                            else
                                temp_second_high = temp_n3;
                            end
                        else
                            temp_high=temp_n3;
                            temp_second_high=strategy1;
                        end
                    elseif(strategy2>temp_n3)
                        temp_high=strategy2;
                        if(temp_n3>strategy1)
                            temp_second_high=temp_n3;
                        else
                            temp_second_high=strategy1;
                        end
                    else
                        temp_high=temp_n3;
                        temp_second_high=strategy2;
                    end

                        temp_utility3 = ctrl3*(temp_high-
temp_second_high);
                    if(utility3 < temp_utility3)
                        temp_eq=0;
                    end

                end

            end


            if temp_eq==1
                eq_1 = strategy1;
                eq_2 = strategy2;
                eq_3 = strategy3;
            end

        end
    end
end
```

```
value1 = 1:100;
%value should be range 1-million but won't compute
value2 = 1:55;
%value should be range 1-555710 but won't compute
value3 = 1:47;
```

```
>> disp(eq_1)
   100

>> disp(eq_2)
    47

>> disp(eq_3)
    47
```

Advertiser 1 Social Welfare = 100, Advertiser 2 Social Welfare = (55), Advertiser 3 Social Welfare = 47.

Price of Anarchy = $PoA = \frac{\max_s Welf(s)}{\min_s Equil\ Welf(s)}$