CSC304 Fall'22
Assignment 1
Due: October 15, 2022, by 11:59pm ET

Instructions:

1. Typed assignments are preferred (e.g., using LaTeX or Word), especially if your handwriting is possibly illegible or if you do not have access to a good quality scanner. Please submit a single PDF to MarkUs.

2. Remember our citation policy. You are free to read online material (though, if you find the exact homework problem, do not read the solution) and to discuss any problems with your peers. There are two rules you must follow: You must write the solution in your own words (it helps to not take any pictures or notes from your discussions), and you must cite any peers or online sources from where you obtained a significant hint.

3. Attempt to solve each problem, but if you have no clue how to approach a (sub)problem, remember the 20% rule: You can get 20% points for a (sub)problem by just stating "I don't know how to approach this question" (and 10% if you leave it blank but do not write such a statement). This policy does not apply to any bonus questions.

4. Bonus questions are optional. You get 100% marks for solving all the non-bonus questions, and may get more than 100% marks when solving bonus questions.

Q1 [20 Points] Nash Equilibria

Compute all (mixed) Nash equilibria in each of the following games. At the end of your solution, explicitly write the set of Nash equilibria using the following notation. A Nash equilibrium can be denoted as $((p_1, p_2, p_3), (q_1, q_2, q_3))$, where p_1, p_2, and p_3 are the probabilities of the row player playing T, M, and B respectively, and q_1, q_2, and q_3 are the probabilities of the column player playing L, C, and R, respectively.

(a) [5 Points] Game 1:

[Hint: Instead of brute-forcing, apply a trick we learned in class to reduce the size of the game without changing the set of Nash equilibria.]

	L	C	R
T	$(1,1)$	$(3,4)$	$(2,1)$
M	$(2,4)$	$(2,5)$	$(8,1)$
B	$(3,3)$	$(0,4)$	$(0,9)$

We notice that whatever which strategy player 1 play, the reward of strategy C is always greater than L for player $2(5 > 4, 4 > 1, 4 > 3)$. Thus we can say L is strictly dominated by C. By iterated elimination and $Q2(a)$ of the tutorial 2, we know that elimination of a strictly dominated action cannot add or remove Nash equilibria. Thus we can remove L from the Nash equilibria set. Similarly, we notice that the reward of strategy M is always greater than B for player $1(2 > 0, 8 > 0)$

which means B is strictly dominated by M. Thus we can remove B from the Nash equilibria set.

Then this problem is reduced to a 2×2 size problem.

We notice that whatever which strategy player 1 play, the reward of strategy C is always greater than R for player $2(5 > 4, 4 > 1)$, which means R is strictly dominated by C. Thus we can remove R from the Nash equilibria set. Similarly, we notice that the reward of strategy T is always greater than M for player $1(3 > 2)$ which means M is strictly dominated by T. Thus we can remove M from the Nash equilibria set.

Until now, the only remain strategy is T for player 1 and C for player 2. Thus the only Nash equilibrium is $((1,0,0),(0,1,0))$.

(b) [15 Points] Game 2: Note that the only differences compared to Game 1 are in the rewards to the row player in (B,C) and (M,R).

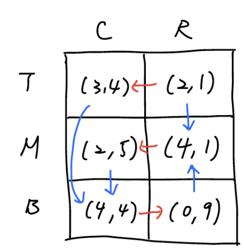
[Hint: It is still possible to reduce the size of Game 2 from 3×3 *to* 3×2*. Applying the indifference principle to the latter will still result in* $(2^3 - 1) \cdot (2^2 - 1) = 21$ *cases. Instead of writing each one separately, you can club them into four categories: neither player randomizes, only the row player randomizes, only the column player randomizes, and both players randomize. Provide a brief argument for any categories that do not lead to any Nash equilibria, and show detailed calculations only for the remaining categories.]*

	L	C	R
T	$(1,1)$	$(3,4)$	$(2,1)$
M	$(2,4)$	$(2,5)$	$(4,1)$
B	$(3,3)$	$(4,4)$	$(0,9)$

Like in 1(a), L is strictly dominated by C, $(5 > 4, 4 > 1, 4 > 3)$. Thus we can remove L from the Nash equilibria set. It now becomes a 2×3 problem.

For the remaining 21 cases, we split them into four categories.

1. Neither player randomizes(6 cases). In this category, we want to identify pure Nash equilibria. By drawing the best response diagram, we notice that every cell has outgoing arrows. Thus this categories can not yield Nash equilibra.



2. Only the row player randomizes(8 cases). It is easy to see that if column player doesn't randomize, the three actions of the row player do not yield equal rewards$(3 \neq 2 \neq 4, 2 \neq 4 \neq 0)$.

The indifference principle says that the row player can randomize between three actions only if rewards from the three actions are equal. Hence, the row player cannot randomize in a Nash equilibrium.

3. Only the column player randomizes(3 cases). It is easy to see that if row player doesn't randomize, the two actions of the row player do not yield equal rewards($4 \neq 1, 5 \neq 1, 4 \neq 9$).

The indifference principle says that the column player can randomize between two actions only if rewards from the two actions are equal. Hence, the column player cannot randomize in a Nash equilibrium.

4. Both players randomize. Suppose player 1 plays T with probability x, M with probability y and B with probability $1 - x - y$, and that player 2 plays C with probability q and R with probability $1 - q$. Because both players randomize, by the indifference principle, each player must have equal rewards for both actions given the mixed strategy of the other player.

Reward of P1: Fixing the mixed strategy of P2, the rewards of P1 for playing T, M, B must be equal. Hence, reward from $T = 3 \times q + 2 \times (1 - q)$, reward from $M = 2 \times q + 4 \times (1 - q)$, reward from $B = 4 \times q$. Solving the equations, we see $q = 2/3$.

These three reward can be equal at the same time, thus by indifference principle, we know that the mixed strategy that gives all of (T, M, B) a positive probability can exist. We can have a partially mixed strategy of $(T, M, B), (T, M), (M, B)$ and (T, B) .

Reward of P2: Fixing the mixed strategy of P1, the rewards of P2 for playing C, R must be equal. Hence, reward from $C = 4x + 5y + 4(1 - x - y)$, reward from $R = x + y + 9(1 - x - y)$. We get that $8x + 9y = 5$, this means we can not have a partially mixed strategy of (T, M) since combine it with $x + y = 1$ will yield $x = 4$ and $y = -3$ which is out of range.Thus we only consider a partially mixed strategy of (M, B) and (T, B) .

For (T, M, B), by previous calculation we get that $y = \frac{5-8x}{9}$ and $q = \frac{2}{3}$. Thus the Nash equilibria is $((x, \frac{5-8x}{9}, \frac{4-x}{9}), (0, \frac{2}{3}, \frac{1}{3}))$.

For (M, B), by previous calculation we get that $y = \frac{5}{9}$ and $q = \frac{2}{3}$. Thus the Nash equilibria is $((0, \frac{5}{9}, \frac{4}{9}), (0, \frac{2}{3}, \frac{1}{3}))$.

For (T, B), by previous calculation we get that $y = \frac{5}{8}$ and $q = \frac{2}{3}$. Thus the Nash equilibria is $((0, \frac{5}{8}, \frac{3}{8}), (0, \frac{2}{3}, \frac{1}{3}))$.

The final Nash equilibira set is $\{((x, \frac{5-8x}{9}, \frac{4-x}{9}), (0, \frac{2}{3}, \frac{1}{3})), ((0, \frac{5}{9}, \frac{4}{9}), (0, \frac{2}{3}, \frac{1}{3})), ((0, \frac{5}{8}, \frac{3}{8}), (0, \frac{2}{3}, \frac{1}{3}))\}$.

Q2 [30 Points] Strategic Computing

In a strategic computing game, there is a set of agents $N = \{1, \ldots, n\}$, and a set of machines $M = \{1, \ldots, m\}$. Each agent i has a job (referred to as job i) that requires p_i time to be processed on any single machine. A pure strategy s_i of agent i is a machine on which job i will be processed (i.e., $s_i \in M$). Given a pure strategy profile $\vec{s} = (s_1, \ldots, s_n)$, the load on machine j is the sum of processing times of the jobs assigned to machine j:

$$L_j(\vec{s}) = \textstyle\sum_{i \in N : s_i = j} p_i.$$

The cost to each agent i is the load on the machine chosen by the agent, i.e., $c_i(\vec{s}) = L_{s_i}(\vec{s})$. The social cost is defined to be the *maximum load on any machine*: $C(\vec{s}) = \max_{j \in M} L_j(\vec{s})$. (Observe that the social cost is not the sum of agent costs, but rather the maximum of agent costs.)

(a) [10 Points] Prove that the price of stability of any strategic computing game is 1. In other words, you need to show that in every strategic computing game, there is at least one socially optimal pure strategy profile that is also a pure Nash equilibrium.

[Hint: Not every socially optimal pure strategy profile will be a Nash equilibrium. First, observe that in a social optimum, an agent assigned to a machine with the highest load must have no incentive to switch. Then, think about when this would be true for the agents assigned to the other machines.]

Note that there are multiple socially pure strategy profile. In a social optimum, all agents assigned to a machine with the strictly highest load must have no incentive to switch. By contradiction we assume that the social optimum is C and one of the agent assigned to a machine with the highest load have incentive to switch to another machine with lower load, then the social optimum will be the highest of these loads, but still a strictly lower number than C. Contradiction appears so that all agents assigned to a machine with the strictly highest load must have no incentive to switch.

Out of all socially pure strategy profile, we iteratively choose the one that has the next smallest social cost (i.e. largest load) of the remaining machines. We describe the process more closely as follows:

Define the set of all social optimal pure strategy profile where we minimize the 1st, 2nd, ..., up to i-th largest load as S_i. In each step, we choose from S_i strategy profiles that minimizes the $i+1$-th largest load as S_{i+1}. We continue this procedure until we have finish iterating over all M loads, resulting in S_M. Consider strategy $\vec{s}$ in S_M.

By construction, $\vec{s}$ is socially optimal as $\vec{s} \in S_M \subseteq S_{M-1} \subseteq ... \subseteq S_1$, and S_1 is the set of all social optimal pure strategy by definition. All that is left is to show $\vec{s}$ as a Nash Equilibrium.

Consider any job scheduled by agent j on machine s_j in $\vec{s}$. Without loss of generality, assume machine $1, ..., M$ have loads $L_1(\vec{s}) >= L_2(\vec{s}) >= ... >= L_M(\vec{s})$ (i.e. we can renumber the machines in decreasing order of loads). Note that if any machine in the sequence have equal loads, we can fix either load first arbitrarily by our construction. Thus, assume further that $L_1(\vec{s}) >= ... >= L_{s_j}(\vec{s}) > ... >= L_M(\vec{s})$, i.e. loads on s_j is fixed last compared to equal loads. Observe by contradiction suppose j would benefit by switching the job from s_j to $s_j^{'}$. Note following observations:

- $s_j > s_j'$ since $L_{s_j}(\vec{s}) > L_{s_j'}(\vec{s})$ by ordering or else j would not benefit by switching

- $L_{s'_j}(s'_j, \vec{s}_j) < L_{s_j}(s_j, \vec{s}_j)$, i.e. after the switch load of s'_j would still be lesser than load of s_j before the switch

- $L_{s_j}(s'_j, \vec{s}_j) < L_{s_j}(s_j, \vec{s}_j)$, i.e. we have reduced the load on s_j

Note all the loads on other machines remains the same, but the larger load out out machine s_j and s'_j decreases strictly after the switch. However, this implies that we can decrease $L_{s_j}(\vec{s})$, contradicting the construction that in step s_j we choose to fix load on s_j such that L_{s_j} is minimized.

(b) [20 Points] Prove that the price of anarchy of any strategic computing game with respect to pure Nash equilibria is at most $2 - 2/(m+1)$.

[Hint: Let OPT be the optimum social cost. You want to prove that the social cost of any pure Nash equilibrium $\vec{s}$ is at most $(2 - 2/(m+1)) \cdot OPT$. Let j^* be a machine with the highest load under $\vec{s}$. First, prove the desired result when j^* has a single job under $\vec{s}$. Next, assume j^* has at least two jobs and i^* is a job on j^* with the minimum processing time among all jobs on j^*. To prove the desired result, derive two inequalities: the first relates OPT to the average (not maximum) load across all machines under $\vec{s}$, and the second relates the load on each machine under $\vec{s}$ to the maximum load under $\vec{s}$.]

Let OPT be the optimum social cost, $\vec{s^*}$ be a socially optimal pure strategy profile, and $\vec{s}$ be a pure Nash Equilibrium. Let $C(\vec{s})$ denotes the social cost of $\vec{s}$. We will show that $C(\vec{s}) \geq (2 - 2/(m+1)) \cdot OPT$.

Let j^* be a machine with the highest load under $\vec{s}$. Consider the two scenarios as follows:

Case 1: j^* has a single job under $\vec{s}$

Note that the socially optimal strategy also has to schedule the single job on j^* to some machine, i.e. $OPT \geq C(\vec{s})$. By definition of socially optimal strategy, we also have $OPT \leq C(\vec{s})$. Hence $OPT = C(\vec{s})$, and we also have the following inequality:

$$C(\vec{s}) = OPT \leq (2 - \frac{2}{(m+1)})OPT$$

Case 2: j^* has at least two jobs under $\vec{s}$

Let job i^* be job on j^* with the minimum processing time out of all jobs on $\vec{s}$. We note the follow inequalities:

1. $OPT \geq$ average of loads across all machines under $\vec{s}$, or $OPT \geq \sum_{k=1}^{N} L_k(\vec{s})/m$

   We note all jobs must be assigned, hence smallest maximum load occurs when jobs are assigned such that each machine has an equal load if possible, i.e. $\sum_{k=1}^{N} L_k(\vec{s})/m$. If such even assignment is not possible, we note that there must exist some pair of machines with one load greater than average, another smaller; In this case, OPT is greater than the average.

2. For all machines $m \neq j^*$, $L_m(\vec{s}) \geq L_{j^*}(\vec{s}) - p_{i^*}$.

   This follows from the fact that $\vec{s}$ is a Nash Equilibrium. Suppose for contradiction $L_m(\vec{s}) < L_{j^*}(\vec{s}) - p_{i^*}$, agent i^* would benefit by switching the job to L_m.

3. $p_{i^*} \leq L_{j^*}(\vec{s})/2$

   This follows from the fact that job i^* has the minimum processing time out of at least two jobs on j^*.

we then prove the inequality by the above facts.

$$\begin{aligned}
m \cdot OPT &\geq \sum_{k=1}^{N} L_k(\vec{s}) && \text{by (1)} \\
&= (\sum_{k \neq j^*} L_k(\vec{s})) + L_{j^*}(\vec{s}) \\
&\geq (m-1) \cdot (L_{j^*}(\vec{s}) - p_{i^*}) + L_{j^*}(\vec{s}) && \text{by (2), and there are (m-1) machines} \neq j* \\
&= mL_{j^*}(\vec{s}) - (m-1)p_{i^*} \\
&\geq mL_{j^*}(\vec{s}) - \frac{(m-1)}{2} \cdot L_{j^*}(\vec{s}) && \text{by (3)} \\
&= \frac{m+1}{2} L_{j^*}(\vec{s})
\end{aligned}$$

Then by definition of j^*, it is the machine with the highest load, i.e. $C(\vec{s}) = L_{j^*}(\vec{s})$. Rearranging, we have:

$$\frac{2m}{m+1} OPT = (2 - \frac{2}{(m+1)}) OPT \geq L_{j^*}(\vec{s}) = C(\vec{s})$$

We have shown $(2 - \frac{2}{m+1})OPT \geq C(\vec{s})$ as required.

Q3 [20 Points] Stackelberg Equilibria

Consider the following game between two players. Again, the row player is Alice, the column player is Bob, the first number in each cell denotes the payoff to Alice, and the second number in each cell denotes the payoff to Bob.

	A	B
X	$5, 4$	$0, 8$
Y	$0, 1$	$3, 2$

(a) [5 Points] Suppose both players have to choose their strategies simultaneously. Identify the unique Nash equilibrium of the game by iteratively eliminating strictly dominated actions. You only need to write down the actions removed, the order in which they are removed, and the unique Nash equilibrium; no further explanation is needed.

Perform the iterative elimination in order:

1. eliminate A for Bob since playing A strictly dominates B $(8 > 4, 2 > 1)$, resulting in a 2×1 game

2. eliminate X for Alice since Y stricly dominates X in the remaining cells where Bob plays B $(3 > 0)$

We are left with a unique Nash equilibrium of (3, 2)

(b) [5 Points] If Bob can commit to a *pure* strategy and let Alice know of his strategy, what is the best pure strategy for him to commit to? What would be his reward in the resulting outcome? Explain your answer.

Consider the two cases.
Bob commits to B Then Alice would choose Y maximizing her reward $(3 > 0)$, resulting in reward of (3, 2)
Bob commits to A Then Alice would choose X maximizing her reward $(5 > 0)$, resulting in reward of (5, 4)
Hence the best pure strategy for Bob to commit to is A, with a resulting outcome of (5, 4).

(c) [10 Points] If Bob can commit to a *mixed* strategy and let Alice know of his strategy, what is the best mixed strategy for him to commit to? What would be his reward in the resulting outcome? Explain your answer. If Alice is indifferent between her two pure strategies, assume that she plays the pure strategy that gives Bob a higher payoff.

[HINT: You do not need to solve a complicated linear program for this question. Just think about which pure strategy (action) Bob would want Alice to respond with.]

To maximize Bob's reward, Bob want Alice to commit to X. Let p be the probability of Bob choosing A, and $1-p$ be that of B. Bob's reward in this case would be $4p + 8(1-p) = 8 - 4p$, so we also wish to minimize p to give Bob the highest reward.

Then

$$\begin{aligned} E[\text{reward to Alice for choosing X}] &= 5p \\ E[\text{reward to Alice for choosing Y}] &= 3(1-p) \end{aligned}$$

Want $5p > 3(1-p)$:

$$\begin{aligned} &\implies 5p > 3 - 3p \\ &\implies 8p > 3 \\ &\implies p > \frac{3}{8} \end{aligned}$$

Notice that when Alice is indifferent between the strategies, she will play the strategy to give Bob the higher payoff. So if Bob plays A,B w.p. $(3/8, 5/8)$, then Alice will choose X. Further, $3/8$ is the minimum value p could take such that Alice will choose X. So, Bob's highest reward will be

$$\begin{aligned} 4p + 8(1-p) &= 8 - 4p \\ &= 8 - 4(\frac{3}{8}) \\ &= 8 - \frac{3}{2} \\ &= \frac{13}{2} \end{aligned}$$

Thus we conclude that the best mixed strategy for Bob to commit to is playing (A, B) with probability (3/8, 5/8). which results in highest reward of 13/2 for Bob.

Bonus Question

Q4 [20 Points] Potential Functions and Cost-Sharing Games

Recall that in the cost-sharing game, we defined the potential function $\Phi(\vec{P}) = \sum_{e \in E(\vec{P})} \sum_{k=1}^{n_e(\vec{P})} \frac{c_e}{k}$, where $E(\vec{P})$ is the set of edges taken by at least one player in $\vec{P}$, and $n_e(\vec{P})$ is the number of players taking edge e in $\vec{P}$.

Consider a cost-sharing game with n players on a network with m edges, where the cost of each edge is 1 (i.e., $c_e = 1$ for each e). Suppose we start from an arbitrary pure strategy profile $\vec{P}^1$ and follow an iterative process. The pure strategy profile at the beginning of iteration t is denoted $\vec{P}^t$. In iteration t, we find a player i who can change her strategy from P_i^t to P_i' to strictly decrease her cost (i.e., $c_i(P_i', \vec{P}_{-i}^t) < c_i(\vec{P}^t)$) and set $\vec{P}^{t+1} = (P_i', \vec{P}_{-i}^t)$. If multiple such players exist, we choose an arbitrary one, and if no such players exist, the process ends.

Our goal is to prove that this process will necessarily converge to a pure Nash equilibrium in finitely many steps, and prove an upper bound on the number of steps required.

In parts (b), (c), and (d) below, all bounds must be in terms of only n and m. Better bounds can fetch higher marks.

(a) [2 Points] Show that if this process stops, the final pure strategy profile must be a pure Nash equilibrium.

(b) [5 Points] Find an upper bound on the initial potential value, i.e., $\Phi(\vec{P}^1)$.

(c) [10 Points] Find a strictly positive lower bound on the amount by which the potential function must decrease in every iteration, i.e., on $\Phi(\vec{P}^t) - \Phi(\vec{P}^{t+1})$.

(d) [3 Points] Using parts (b) and (c), prove an upper bound on the number of steps required for the process to converge to a pure Nash equilibrium. Your bound must hold regardless of the choices made during the process (i.e., wherever we said “arbitrary” in the process description).