Soham Chatterjee
Email: soham.chatterjee@tifr.res.in
Course: Algorithmic Game Theory

Assignment - 2
Dept: STCS, TIFR
Date: March 11, 2025

Problem 1

10 Marks

Consider the bimatrix game (A, B) where $A, B \in \mathbb{R}^{n\times n}$ and, further, $\text{rank}(A) = \text{rank}(B) = k$. Give an algorithm that computes a Nash equilibrium in this game in time poly $\left(n^{O(k)}, |A|, |B|\right)$ where, as before, $|x|$ is the bit-complexity of x. You may need to use Caratheodory's theorem:

Theorem 1. *Let $S = \{x_1, x_2, \dots, x_n\}$ be a set of points in $\mathbb{R}^k$, and y lie in the convex hull of S. Then there exists $\bar{S} \subseteq S$ of cardinality $k+1$, so that y lies in the convex hull of the points in $\bar{S}$.*

Solution: Let Δ_i denote the set of mixed strategies for player i. Now since $\text{rank}(R) = k$ the vector space $V := \text{Span}_{\mathbb{R}}\{Ry \mid y \in \Delta_2\}$ has dimension k. Therefore V is isomorphic to $\mathbb{R}^k$. Suppose $f : V \to \mathbb{R}^k$ is the isomorphism. Let R_i denote the i^{th} row of R for $i \in [n]$. Suppose $\tilde{R}_i = f(R_i)$. Now for any y the vector Ry is basically the y-convex combination the rows R_i. Therefore $f(Ry) = \sum_{i=1}^{n} y_i\tilde{R}_i$. Now by Caratheodory's Theorem we get that $\exists\, S \subseteq \{\tilde{R}_i \mid i \in [n]\}$with $|S| = k+1$ such that $f(Ry)$ is convex combination of $\{\tilde{R}_i \mid i \in S\}$. Hence there exists a probability distribution y' with $y'_i > 0 \iff i \in S$ such that $f(Ry)$ is y'-convex combination of $\{\tilde{R}_i \mid i \in S\}$ i.e.

$$f(Ry) = \sum_{i\in S} y'_i\tilde{R}_i \iff Ry = \sum_{i\in S} y'_i R_i$$

Therefore we obtain a new strategy y' for player 2 which is supported on at most $k+1$ pure strategies. Instead of doing this process for all $\{R_i \mid i \in [n]\}$ whose y-convex sum is giving the value Ry if we started with the set of vectors $\{R_i \mid i \in \text{Supp}(y)\}$ we would have gotten a y' such that $\text{Supp}(y') \subseteq \text{Supp}(y)$ with $Ry' = Ry$. Hence we can assume that for any strategy $y \in \Delta_2$ there exists $y' \in \Delta_2$ such that $\text{Supp}(y') \subseteq \text{Supp}(y)$ and $|\text{Supp}(y')| \leq k+1$ with $Ry = Ry'$. Similarly we can also assume if player 1 plays any mixed strategy $x \in \Delta_1$ then by the above process on the rows of C we obtain $x' \in \Delta_1$ such that $\text{Supp}(x') \subseteq \text{Supp}(x)$ and $|\text{Supp}(x')| \leq k+1$ with $C^T x = C^T x'$.

Now suppose (x^*, y^*) is an MNE . Then $\exists\, \tilde{y}^* \in \Delta_2$ such that $\text{Supp}(\tilde{y}^*) \subseteq \text{Supp}(y^*)$ and $|\text{Supp}(\tilde{y}^*)| \leq k+1$ with $Ry^* = R\tilde{y}^*$. Now since $\text{Supp}(\tilde{y}^*) \subseteq \text{Supp}(y^*)$, $\tilde{y}^*$ is a best response to x^*. And since $Ry^* = R\tilde{y}^*$ we have ${x^*}^T Ry^* = {x^*}^T R\tilde{y}^*$. Therefore x^* is also best response to $\tilde{y}^*$. Now by the same way from $(x^*, \tilde{y}^*)$ we $\exists\, \tilde{x}^* \in \Delta_1$ such that $\text{Supp}(\tilde{x}^*) \subseteq \text{Supp}(x^*)$ and $|\text{Supp}(\tilde{x}^*)| \leq k+1$ with $C^T x^* = C^T\tilde{x}^*$. Since $\text{Supp}(\tilde{x}^*) \subseteq \text{Supp}(x^*)$, $\tilde{x}^*$ is best response to $\tilde{y}^*$. And since $C^T x^* = C^T\tilde{x}^*$ we have $\tilde{y}^* C^T x^* = \tilde{y}^* C^T \tilde{x}^*$. Therefore $\tilde{y}^*$ is best response to $\tilde{x}^*$. Therefore $(\tilde{x}^*, \tilde{y}^*)$ is also MNE . Therefore from (x^*, y^*) we obtained an $(\tilde{x}^*, \tilde{y}^*)$ which is also an MNE but both of the strategies are supported on at most $k+1$ pure strategies. So we have the following algorithm:

(1): Choose all possible $S_1, S_2 \subseteq [n]$ with $|S_1|, |S_2| \leq k+1$.

(2): For each (S_1, S_2) consider the matrices $A_{S_1,S_2} = (A(i,j))_{i\in S_1, j\in S_2}$ $B_{S_1,S_2} = (B(i,j))_{i\in S_1, j\in S_2}$.

(3): Run the Lemke-Howson Algorithm on $(A_{S_1,S_2}, B_{S_1,S_2})$ find Nash Equilibrium $(x^*_{S_1,S_2}, y^*_{S_1,S_2})$ in the reduced matrices

(4): For each (S_1, S_2) check if $(x^*_{S_1,S_2}, y^*_{S_1,S_2})$ satisfies the Nash Equilibrium conditions in the full game and if it satisfies return that strategy.

The all possible choice of $S_1, S_2 \subseteq [n]$ with $|S_1|, |S_2| \leq k+1$ takes at most $2\sum_{i=1}^{k+1}\binom{n}{i} \leq poly(n^{O(k)})$ time. Now each of the reduced matrices can also be computed in $poly(|A|, |B|)$. For each reduced matrix the Lemke-Howson Algorithm runs on $(K = 1) \times (k+1)$ matrix. Therefore the algorithm takes at most $poly(k^{O(k)})$ time. Therefore the total running time is $poly(n^{O(k)}, |A|, |B|)$.

■

Problem 2 10 Marks

Recall the single-agent regret-minimization problem with n pure strategies studied in class, for which we showed that the multiplicative weight algorithm with $\epsilon = \sqrt{\ln n/T}$ has regret $2\sqrt{\ln n/T}$. Modify the algorithm to remove the assumption that T is known to the algorithm, while maintaining a bound of $O(\sqrt{\ln n}/\sqrt{T})$ on the regret.

Solution: Let $\mathcal{A}$ be the algorithm for single-agent no regret on input T and the set of n pure strategies gives a no-regret dynamics. So we will have the following algorithm which has no knowledge of T:

Algorithm 1: MULTIPLICATIVE WEIGHT WITHOUT KNOWING T

Input: A set S of $n \geq 2$ actions
Output: No regret Dynamics

begin
  $i \longleftarrow 0$
  while *True* **do**
    $i \longleftarrow i + 1$
    $t \longleftarrow 2^i$
    $p^t \longleftarrow \mathcal{A}(t, S)$
    if *It is the end* **then**
      return p^t

Basically the algorithm has $\lfloor \log T \rfloor + 1$ phases during T. In phase $k \geq 0$ it consists of steps $2^k, \dots, 2^{k+1}-1$ steps i.e. total 2^k steps. At beginning of a phase we restart the no-regret algorithm with $t = 2^k$. As the last phase may not be complete the algorithm is stopped in the last phase after the number of rounds is over. We will denote $p^t_{2^k}$ be the distribution at the t^{th} iteration in the phase k. Now before we go into the calculation of the regret we have to first handle the case of the last phase where from $2^{\lfloor \log T \rfloor}$ to $2^{\lfloor \log T \rfloor + 1}$, T can be any value between them. And for all such T's we are running the algorithm with the same $\epsilon = \sqrt{\frac{\ln n}{2^{\lfloor \log T \rfloor}}}$. We have to show that for all T this choice of ϵ gives the same regret $O\left(\sqrt{\frac{\ln n}{2^{\lfloor \log T \rfloor}}}\right)$.

Lemma 1. *For any T for the above algorithm in the last phase $\mathcal{A}$ will run for $T - 2^{\lfloor \log T \rfloor}$ many rounds with the choice of $\epsilon_1 = \sqrt{\frac{\ln n}{2^{\lfloor \log T \rfloor}}}$ instead of $\epsilon_2 = \sqrt{\frac{\ln n}{T - 2^{\lfloor \log T \rfloor}}}$. Let the regret for any ϵ is denoted R_ϵ. Then*

$$R_{\epsilon_2}^{T-2^{\lfloor \log T \rfloor}} = \Theta\left(R_{\epsilon_1}^{T-2^{\lfloor \log T \rfloor}}\right)$$

Proof: In the analysis of the algorithm $\mathcal{A}$ if it runs for t rounds for any ϵ we have

$$R_\epsilon^t \leq \frac{1}{t}\left(\epsilon t + \frac{1}{\epsilon}\ln n\right) \iff t \cdot R_\epsilon^t \leq \epsilon T + \frac{1}{\epsilon}\ln n$$

So we will show that

$$\epsilon_1\left(T - 2^{\lfloor \log T \rfloor}\right) + \frac{1}{\epsilon_1}\ln n = \Theta\left(\epsilon_2\left(T - 2^{\lfloor \log T \rfloor}\right) + \frac{1}{\epsilon_2}\ln n\right)$$

Now

$$
\begin{aligned}
&\epsilon_2\left(T-2^{\lfloor\log T\rfloor}\right)+\frac{1}{\epsilon_2}\ln n=\Theta\left(\epsilon_1\left(T-2^{\lfloor\log T\rfloor}\right)+\frac{1}{\epsilon_1}\ln n\right)\\
\iff{}&\sqrt{\frac{\ln n}{T-2^{\lfloor\log T\rfloor}}}\left(T-2^{\lfloor\log T\rfloor}\right)+\frac{1}{\sqrt{\frac{\ln n}{T-2^{\lfloor\log T\rfloor}}}}\ln n=\Theta\left(\sqrt{\frac{\ln n}{2^{\lfloor\log T\rfloor}}}\left(T-2^{\lfloor\log T\rfloor}\right)+\frac{1}{\sqrt{\frac{\ln n}{2^{\lfloor\log T\rfloor}}}}\ln n\right)\\
\iff{}&2\sqrt{T-2^{\lfloor\log T\rfloor}}=\Theta\left(\frac{1}{\sqrt{2^{\lfloor\log T\rfloor}}}\left(T-2^{\lfloor\log T\rfloor}\right)+\sqrt{2^{\lfloor\log T\rfloor}}\right)\\
\iff{}&2\sqrt{T-2^{\lfloor\log T\rfloor}}=\Theta\left(\frac{T}{\sqrt{2^{\lfloor\log T\rfloor}}}\right)\\
\iff{}&4(T-2^{\lfloor\log T\rfloor})=\Theta\left(\frac{T^2}{2^{\lfloor\log T\rfloor}}\right)\\
\iff{}&4(T-2^{\lfloor\log T\rfloor})=\Theta(T)=\Theta\left(\frac{T^2}{2^{\lfloor\log T\rfloor}}\right)
\end{aligned}
$$

Hence $R_{\epsilon_2}^{T-2^{\lfloor\log T\rfloor}}=\Theta\left(R_{\epsilon_1}^{T-2^{\lfloor\log T\rfloor}}\right)$ □

Therefore the regret in the last phase can be at most $O\left(\sqrt{\frac{\ln n}{T-2^{\lfloor\log T\rfloor}}}\right)$. Suppose regret is at most $\delta\sqrt{\frac{\ln n}{T-2^{\lfloor\log T\rfloor}}}$ for some $\delta\in\mathbb{N}$, $\delta\geq 2$. Therefore in the last phase we have

$$\sum_{t=0}^{2^m-1}\sum_{a\in S}p^t_{2^{\lfloor\log T\rfloor}}(a)c^{t+2^m}(a)-\min_{a\in S}\sum_{t=0}^{2^{\lfloor\log T\rfloor}-1}c^{t+2^{\lfloor\log T\rfloor}}(a)\leq\delta\sqrt{\left(T-2^{\lfloor\log T\rfloor}\right)\ln n}\leq\delta\sqrt{2^{\lfloor\log T\rfloor}\ln n}$$

Now for $k\in\{0,\ldots,m-2\}$ by the analysis of $\mathcal{A}$ we know

$$\sum_{t=0}^{2^k-1}\sum_{a\in S}p^t_{2^k}(a)c^{t+2^k}(a)-\min_{a\in S}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)\leq 2\sqrt{2^k\ln n}\leq\delta\sqrt{2^k\ln n}$$

Therefore we have

$$\sum_{k=0}^{\lfloor\log T\rfloor}\sum_{t=0}^{2^k-1}\sum_{a\in S}p^t_{2^k}(a)c^{t+2^k}(a)-\sum_{k=0}^{\lfloor\log T\rfloor}\min_{a\in S}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)\leq\sum_{k=0}^{m}\delta\sqrt{2^k\ln n}$$

Now we have

$$\sum_{k=0}^{\lfloor\log T\rfloor}\min_{a\in S}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)\leq\min_{a\in S}\sum_{k=0}^{\lfloor\log T\rfloor}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)$$

Therefore we have

$$
\begin{aligned}
\sum_{k=0}^{\lfloor\log T\rfloor}\sum_{t=0}^{2^k-1}\sum_{a\in S}p^t_{2^k}(a)c^{t+2^k}(a)-\min_{a\in S}\sum_{k=0}^{m}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)&\leq\sum_{k=0}^{\lfloor\log T\rfloor}\sum_{t=0}^{2^k-1}\sum_{a\in S}p^t_{2^k}(a)c^{t+2^k}(a)-\sum_{k=0}^{\lfloor\log T\rfloor}\min_{a\in S}\sum_{t=0}^{2^k-1}c^{t+2^k}(a)\\
&\leq\sum_{k=0}^{\lfloor\log T\rfloor}\delta\sqrt{2^k\ln n}=\delta\sqrt{\ln n}\sum_{k=0}^{\lfloor\log T\rfloor}\sqrt{2}^k\\
&=\delta\sqrt{\ln n}\frac{(\sqrt{2})^{\lfloor\log T\rfloor+1}-1}{\sqrt{2}-1}=O(\sqrt{T\ln n})
\end{aligned}
$$

Hence

$$\frac{1}{T}\left[\sum_{k=0}^{m}\sum_{t=1}^{2^k-1}\sum_{a\in S}p^{t+2^k}(a)c^{t+2^k}(a)-\min_{a\in S}\sum_{k=0}^{m}\sum_{t=1}^{2^k-1}c^{t+2^k}(a)\right]\leq O\left(\sqrt{\frac{\ln n}{T}}\right)$$

Therefore this algorithm gives the same external regret as the algorithm. Therefore this algorithm has no regret as $T\to\infty$. ■

Problem 3 10 Marks

We saw in class that any deterministic regret-minimization algorithm, that selects a point distribution p^t at each time t, has regret at least $1 - 1/n$. Consider the deterministic regret-minimization algorithm that at each time t, selects the pure strategy that has least cumulative cost so far. That is, $p^t(a) = 1$ for some $a \in \arg\min \sum_{\tau \le t} c^\tau(a)$. Show the regret of this algorithm (called "Follow-the-Leader") is at most

$$\frac{(n-1)\mathrm{OPT}}{T} + \frac{n}{T}$$

Solution: Let a_t denotes the action taken at time t. Let the actions are $a_1, \ldots, a_n$. Therefore $a_t \in \arg\min_{i \in [n]} \sum_{\tau \le t} c^\tau(a_i)$. Now total cost of the algorithm after T time is $\sum_{t=1}^{T} c^t(a_t)$. Let t_i denote the last time when the algorithm choose the action a_i for all $i \in [n]$. Let $c_i = \sum_{t=1}^{t_i - 1} c^t(a_i)$. Now $\mathrm{OPT} = \min_{i \in [n]} \sum_{t=1}^{T} c^t(a_i)$. If $a^* = \arg\min_{i \in [n]} \sum_{t=1}^{T} c^t(a_i)$ then define $\mathrm{OPT}_t = \sum_{\tau \le t-1} c^\tau(a^*)$.

Lemma 2. $c_i \le \mathrm{OPT}$ *for all* $i \in [n]$

Proof: For any $i \in [n]$ we have $c_i \le \mathrm{OPT}_{t_i}$ since t_i is the last time the action a_i was chosen by the algorithm and henceforth $a_i \in \arg\min_{j \in [n]} \sum_{\tau \le t_i} c^\tau(a_j)$. Since $\mathrm{OPT}_t \le \mathrm{OPT}$ for all $i \in [T]$ we have $c_i \le \mathrm{OPT}$. Since $i \in [n]$ is arbitrary we have the lemma. □

Lemma 3. $\sum_{t=1}^{T} c^t(a_t) \le \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} c^{t_i}(a_i)$.

Proof: For $i \in [n]$ denote $d_i = \sum_{\substack{t \le T \\ a_t = a_i}} c^\tau(a_i) = \sum_{\substack{t \le t_i \\ a_t = a_i}} c^\tau(a_i)$. Then we have $\forall\, i \in [n]$, $d_i - c^{t_i}(a_i) \le c_i$ since every summand in $d_i - c^{t_i}(a_i)$ i.e. every summand except the last element appears as a summand in c_i. Since $\sum_{i \in [n]} d_i = \sum_{t=1}^{T} c^t(a_t)$ we have $\sum_{t=1}^{T} c^t(a_t) \le \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} c^{t_i}(a_i)$. □

Now since $c^t(a_i) \in [0,1]$ for all $i \in [n]$, for all $t \in [T]$. Therefore

$$\sum_{t=1}^{T} c^t(a_t) \le n\,\mathrm{OPT} + n \implies \sum_{t=1}^{T} c^t(a_t) - \mathrm{OPT} \le (n-1)\,\mathrm{OPT} + n \implies \frac{1}{T}\left(\sum_{t=1}^{T} c^t(a_t) - \mathrm{OPT}\right) \le \frac{(n-1)\,\mathrm{OPT}}{T} + \frac{n}{T}$$

Hence the regret is at most $\frac{(n-1)\mathrm{OPT}}{T} + \frac{n}{T}$

■

Problem 4 10 Marks

Recall the value of a zero-sum game: this was the payoff for the row player in any Nash equilibrium of the game. Show that, in fact, this extends to CCE of zero-sum games as well: any CCE has the same payoff for the row-player.

Solution: Suppose (x^*, y^*) is an MNE in the zero sum game. And the payoff obtained by the player 1 is w^*. Now we have the lemma

Lemma 4. *Let* μ *is a* CCE *. And* μ_1 *and* μ_2 *are the marginals of* μ *for player 1 and player 2. Then*

$$\mathop{\mathbb{E}}_{s \sim \mu}[u_i(s)] = u_i(\mu_1, \mu_2)$$

Proof: Suppose for some $i \in [2]$, $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] \neq u_i(\mu_1,\mu_2)$. So either $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] < u_i(\mu_1,\mu_2)$ or $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] > u_i(\mu_1,\mu_2)$. If

$$\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] > u_i(\mu_1,\mu_2) \implies -\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] < -u_i(\mu_1,\mu_2) \implies \mathop{\mathbb{E}}_{s\sim\mu}[u_{3-i}(s)] < u_{3-i}(\mu_1,\mu_2)$$

So it is reduced to the case that $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] < u_i(\mu_1,\mu_2)$. Now there exists $s_1 \in S_1$ such that $u_i(\mu_1,\mu_2) \leq u_i(s_1,\mu_2)$. Then $\exists\, s_1 \in S_1$ such that $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] < u_i(s_1,\mu_2)$. Hence it contradicts the fact that μ is an CCE . Which is not possible. Contradiction $\lightning$ Therefore $\mathop{\mathbb{E}}_{s\sim\mu}[u_i(s)] = u_i(\mu_1,\mu_2)$. □

Since (x^*,y^*) is an MNE we have $u_1(\mu_1,\mu_2) \leq u_1(x^*,y^*) = w^* \implies \mathop{\mathbb{E}}_{s\sim\mu}[u_1(s)] \leq w^*$. Now we also have $u_2(\mu_1,\mu_2) \leq u_2(x^*,y^*) = -w^*$. Therefore

$$\mathop{\mathbb{E}}_{s\sim\mu}[u_2(s)] \leq -w^* \implies -\mathop{\mathbb{E}}_{s\sim\mu}[u_1(s)] \leq -w^* \implies \mathop{\mathbb{E}}_{s\sim\mu}[u_1(s)] \geq w^*$$

Therefore we got $\mathop{\mathbb{E}}_{s\sim\mu}[u_1(s)] = w^*$. Hence any CCE has the same payoff for the row-player. ■

Problem 5 — 5 Marks

In a 3-player zero sum game, for any pure strategy profile s, $\sum_{i=1}^{3} u_i(s) = 0$. Either give an efficient algorithm for computing an MNE in a 3-player zero-sum game, or prove that computing a MNE in a 3-player zero-sum game is PPAD-hard.

Solution: We will show a polynomial time reduction from 2NASH to 3-player zero sum. Suppose we have a bimatrix game (R,C) where $R,C \in \mathbb{R}^{m\times n}$. Let u_1 and u_2 were the payoff functions of the bimatrix game. We construct a 3-player zero sum game we add a new third player. Let the payoff functions in the new 3-player are $\tilde{u}_i$ for $i \in [3]$. For the first player and the second player for any strategy profile $s = (s_1,s_2,s_3)$ becomes

$$u_i(s_1,s_2,s_3) = u_i(s_1,s_2)\ \forall\, i \in [2]$$

Now for any strategy profile $s = (s_1,s_2,s_3)$ the payoff for the third player given by

$$u_3(s_1,s_2,s_3) = -u_1(s_1,s_2) - u_2(s_1,s_2)$$

So we can think that the player 3 has only one strategy $S_3 = \{s_3\}$. Hence the new game has payoff matrices (A,B,C) where $A,B,C \in \mathbb{R}^{m\times n\times 1}$, where for any $s = (i,j,s_3)$, $i \in [m]$, $j \in [n]$ we have

$$A(i,j,s_3) = R(i,j), \qquad B(i,j,s_3) = C(i,j) \qquad C(i,j,s_3) = -R(i,j) - C(i,j)$$

game is indeed a 3-player zero sum game. And the reduction from the 2NASH game is polynomial time. Since 2NASH is PPAD-hard we can conclude that 3-player zero sum games are also PPAD-hard. ■

Problem 6 — 10 Marks

Given a 2-player game (R,C), prove that the following problems are either in P or are NP-complete:

- Determine if there exists an MNE (x^*,y^*) where both players play each pure strategy with positive probability (i.e., $x_i^* > 0$ for all i, and $y_j^* > 0$ for all j).
- Determine if there exists an MNE (x^*,y^*) where $x_1^* = 1$ (i.e., a given pure strategy is played with probability 1).

Solution: In the following games we will assume that the row player has m strategies and the column player has n strategies.

- Here we have to find an MNE (x^*, y^*) with the property that $\sup(x^*) = [m]$ and $\mathrm{Supp}(y^*) = [n]$ i.e. they have full support. Since their support size is full we have $(Ry^*)_k = \arg\max_{i\in[m]}(Ry^*)_i$ for all $k \in [m]$. Hence $(Ry^*)_i = (Ry^*)_k$ for all $i, k \in [m]$. Similarly we have $(C^T x^*)_j = (C^T x^*)_l$ for all $j, l \in [n]$. Hence consider the following LP:

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & (Ry)_i = (Ry)_k \quad \forall\, i, k \in [m], \\ & (C^T x)_j = (C^T x)_l \ \forall\, j, l \in [n], \\ & \sum_{i=1}^{m} x_i = \sum_{j=1}^{n} y_i = 1, \\ & x > 0, y > 0 \end{aligned}$$

  Any solution of the above LP will give a MNE with full support size. Since LP can be solved in polynomial time this problem is in P.

- Here we have to find an MNE (x^*, y^*) with the property that $x_1^* = 1$. That means the row player is playing the first pure strategy with full probability. Since y^* is the best response for x^* we have

$$\mathrm{Supp}(y^*) \subseteq \arg\max_{j\in[n]} C(1, j)$$

  So suppose $T = \arg\max_{j\in[n]} C(1, j)$. Since (x^*, y^*) is MNE we have to ensure that $1 \in \arg\max_{i\in[m]}(Ry^*)_i$. So consider the following LP:

$$\begin{aligned} \text{maximize} \quad & 0 \\ \text{subject to} \quad & (Ry)_1 \geq (Ry)_i \ \forall\, i \in [m], \\ & y_j = 0 \qquad\quad \forall\, j \notin T, \\ & \sum_{j\in T} y_j = 1, \\ & y \geq 0 \end{aligned}$$

  Any solution of the above LP will give a MNE (x^*, y^*) such that $x_1^* = 1$ and the first constraint of LP will ensure x^* is best response for y^* and the second and third condition will ensure $\mathrm{Supp}(y^*) \subseteq \arg\max_{j\in[n]} C(1, j)$ i.e. y^* is best response for x^*. Since LP can be solved in polynomial time this problem is in P.

■

Problem 7 — 10 Marks

Players 1 and 2 choose an element of the set $\{1, \cdots, K\}$. If the players choose the same number, then player 2 pays 1 rupee to player 1 ; otherwise no payment is made. Find all pure and mixed strategy Nash equilibrium of this game.

Solution: If both players chooses same number then player 2 pays 1 rupee to player 1 otherwise no payment is made. So consider the matrices $R, C \in \{0, 1\}^{K\times K}$ where for any $i, j \in [K]$

$$R(i, j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad C(i, j) = \begin{cases} -1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Lemma 5. *There is no* PNE *in the bimatrix game* (R, C)

Proof: Suppose there is a PNE . Let (i, j) where $i, j \in [K]$ is the PNE . Now if $i = j$ then the payoff of the player 2 is -1. But player 2 could play $l \in [K]$, $l \neq i$ and get a pay-off of 0. Hence this is not an equilibrium. So then $i \neq j$. But then the player 1 is getting a payoff 0. But instead player 2 could have played j and gotten a pay-off of 1.. Hence this also not a PNE . Therefore for al (i, j) where $i, j \in [K]$, (i, j) is not a PNE . But we assumed that there is a PNE . Hence contradiction $\lightning$ So there is no PNE in the bimatrix game $\square$

Since this is a finite game with each player having finitely many strategies. By Nash's theorem there exists a mixed nash equilibrium.

Lemma 6. *Choosing all the numbers uniformly at random for both players is an* MNE *in the bimatrix game* (R, C).

Proof: First we will show that (x, y) where $x_i = y_j = \frac{1}{K}$ for all $i, j \in [K]$ is an MNE . Now in the matrix R it has 1's on the diagonal and the rest of the entries are zero and similarly in C it has -1's on the diagonal and the rest of the entries are zero. Then

$$Ry = \begin{bmatrix} \frac{1}{K} \\ \vdots \\ \frac{1}{K} \end{bmatrix} \qquad C^T x = -\begin{bmatrix} \frac{1}{K} \\ \vdots \\ \frac{1}{K} \end{bmatrix}$$

Therefore $\arg\max_{i \in [K]} (Ry)_i = [K]$ and $\arg\max_{j \in [K]} (C^T x)_j = [K]$. Hence x is indeed a best response to y and y is indeed best response to x. Therefore (x, y) is an MNE . $\square$

Now we will show if (x, y) is a mixed strategy of player 1 and 2 respectively. Then for all $i, j \in [K]$. Now $Ry = y$ and $C^T x = -x$ and the expected payoff $u_1(x, y) = \sum_{i \in [K]} x_i y_i = -u_2(x, y)$. Since the game is zero sum the value of the game is $= \min_{y \in \Delta(K)} \max_{i \in [K]} y_i$. Player 1 benefits when their probability distribution x is concentrated where y is highest. So in order to minimize the payoff of player 1, player 2 will try to minimize $\max_{i \in [K]} y_i$. Now for any distribution $y \in \Delta(K)$, $\max_{i \in [K]} y_i \geq \frac{1}{K}$. Hence $\max_{i \in [k]} y_i = \frac{1}{K}$. Hence $y^* = \frac{1}{K}\mathbb{1}$ where $\mathbb{1}$ is the all 1's vector minimizes the payoff of player 1 the most. So for any pure strategy $i \in [k]$ of player 1 the expected payoff is $u_1(i, y^*) = y_i^* = \frac{1}{K} = u_2(i, y^*)$. So every pure strategy gives the same payoff.

Therefore if x^* is best response to y^* then $\mathrm{Supp}(x^*) \subseteq [K]$. Now suppose $\mathrm{Supp}(x^*) \subsetneq [K]$. Let $i, j \in [K]$, $i \neq j$ such that $i \notin \mathrm{Supp}(x^*)$ but $j \in \mathrm{Supp}\, x^*$. Then $C^T x^* = -x^*$. So $j \notin \arg\max_{l \in [K]} -x_l^*$ as $-x_j^* < 0$ and $-x_i^* = 0$. Therefore the y_j^* has to be 0. But we know $y_j^* > 0$. So y^* is not best response to x^*. Therefore $\mathrm{Supp}(x^*) = [K]$. Now again with the same logic since $C^T x^* = -x^*$ and y^* is best response of x^* with $y_i^* > 0$ for all $i \in [K]$ we have $x_i^* = x_j^*$ for all $i, j \in [K]$. Therefore $x^* = \frac{1}{K}\mathbb{1}$. Therefore the only MNE is (x^*, y^*) with $x_i^* = y_j^* = \frac{1}{K}$ for all $i, j \in [K]$. $\blacksquare$