Solving Normal-Form Games

Branislav Bošanský, Dominik Seitz

Czech Technical University in Prague

dominik.seitz@aic.fel.cvut.cz

October 12, 2021

Finding Nash Equilibria (NE)

	L	**R**
U	$(2, 1)$	$(0, 0)$
D	$(0, 0)$	$(1, 2)$

Finding Nash Equilibria (NE)

	L	**R**
U	$(2, 1)$	$(0, 0)$
D	$(0, 0)$	$(1, 2)$

We can also find *mixed* Nash equilibria.
Example: Column player (player 2) plays $\mathbf{L}$ with probability p and $\mathbf{R}$ with probability $(1-p)$. Then

$$\mathbb{E}u_1(\mathbf{U}) = \mathbb{E}u_1(\mathbf{D})$$
$$2p + 0(1-p) = 0p + 1(1-p)$$
$$p = \frac{1}{3}$$

Finding Nash Equilibria (NE)

	$\mathbf{L}$	$\mathbf{R}$
$\mathbf{U}$	$(2, 1)$	$(0, 0)$
$\mathbf{D}$	$(0, 0)$	$(1, 2)$

We can also find *mixed* Nash equilibria.
Example: Column player (player 2) plays $\mathbf{L}$ with probability p and $\mathbf{R}$ with probability $(1 - p)$. Then

$$\mathbb{E}u_1(\mathbf{U}) = \mathbb{E}u_1(\mathbf{D})$$
$$2p + 0(1 - p) = 0p + 1(1 - p)$$
$$p = \frac{1}{3}$$

Similarly, we can compute the strategy for player 1 arriving at $(\frac{2}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{2}{3})$ as Nash equilibrium.

Computing NE

Consider the following two-player game:

	L	**R**
U	$3, 2$	$1, 3$
D	$1, 0$	$2, -2$

Task 1: Find all NE.

Computing NE in Zero-Sum Games

Computing NE in Zero-Sum Games

We can also compute Nash equilibria for two-player, zero-sum games using linear programming:

Computing NE in Zero-Sum Games

We can also compute Nash equilibria for two-player, zero-sum games using linear programming:

$$\max_{s,U} \quad U \tag{1}$$

$$\text{s.t.} \quad \sum_{a_1 \in \mathcal{A}_1} s(a_1) u_1(a_1, a_2) \geq U \qquad \forall a_2 \in \mathcal{A}_2 \tag{2}$$

$$\sum_{a_1 \in \mathcal{A}_1} s(a_1) = 1 \tag{3}$$

$$s(a_1) \geq 0 \qquad \forall a_1 \in \mathcal{A}_1 \tag{4}$$

Computing NE

Consider the following zero-sum normal-form game (the row player is maximizing the utility, the column player is minimizing):

	L	**M**	**R**
U	3	4	-1
C	1	2	0
D	0	-1	1

Task 2: Compute a Nash equilibrium.

Computing NE

Consider the following zero-sum normal-form game (the row player is maximizing the utility, the column player is minimizing):

	L	M	R
U	3	4	-1
C	1	2	0
D	0	-1	1

Task 2: Compute a Nash equilibrium.

Task 3: Alice and Bob are playing a zero-sum game and both of them compute a Nash equilibrium strategy. Before playing, Alice announces which strategy she is going to play. Is Bob going to change strategy?

Computing Correlated Equilibria

Computing a Correlated equilibrium is easier compared to Nash and can be found by linear programming:

$$\sum_{s_{-i} \in S_{-i}} p(s_i, s_{-i}) u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} p(s_i, s_{-i}) u_i(s_i', s_{-i})$$

$$\forall i \in \mathcal{N},$$

$$\forall s_i' \in S_i$$

$$\sum_{s \in S} p(s) = 1 \qquad p(s) \geq 0 \quad \forall s \in S$$

Computing CE

Task 4: Alice and Bob are driving towards a crossroads. They can either decide to **G**o or **S**top. If they both decided to go, they will crash. Find a correlated equilibrium that is not a Nash equilibrium.

	G	**S**
G	$-10, -10$	$5, 0$
S	$0, 5$	$-1, -1$

Computing CE

Task 4: Alice and Bob are driving towards a crossroads. They can either decide to **G**o or **S**top. If they both decided to go, they will crash. Find a correlated equilibrium that is not a Nash equilibrium.

	G	**S**
G	$-10, -10$	$5, 0$
S	$0, 5$	$-1, -1$

Task 5: Modify the algorithm so that the correlated equilibrium maximizes the minimum expected utility value of the players.