CS639: Algorithmic Game Theory & Learning

Chapter 3: Solving Games via Linear Programs

Kirthevasan Kandasamy

UW-Madison

Outline

1. Review of linear programming
2. Computing safe strategies via LPs
3. Computing correlated equilibria via LPs
4. Computing coarse correlated equilibria via LPs

Slides are intended as teaching aids only and do not include all material discussed in class. Students are strongly encouraged to attend lectures and take their own notes.

Ch 3.1: A brief review of linear programming

Example (E.g. A.2.1 in KP). A small factory makes two products using two resources A and B. The first product uses 4 units of resource A and 1 unit of resource B. The second uses 1 unit of resource A and 2 units of resource B.

The first product brings in \$2 of revenue per pound and the second \$1 per pound. The factory has 6 units of resource A and 5 units of resource B. How many pounds of each product should the factory make to maximize revenue?

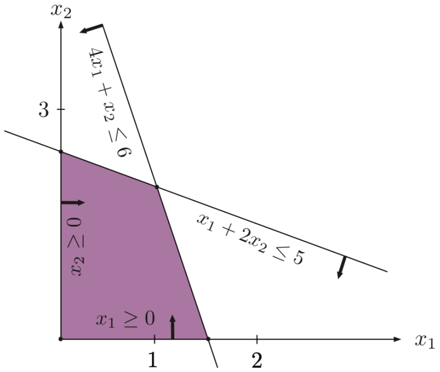
We can write this as the following optimization problem: Let x_1, x_2 denote the number of pounds produced for each product.

$$\begin{aligned}
\text{maximize} \quad & 2x_1 + x_2 && \texttt{Revenue} \\
\text{subject to} \quad & 4x_1 + x_2 \le 6, && \texttt{Resource A} \\
& x_1 + 2x_2 \le 5, && \texttt{Resource B} \\
& x_1 \ge 0, \\
& x_2 \ge 0.
\end{aligned}$$

Resource allocation example (cont'd)

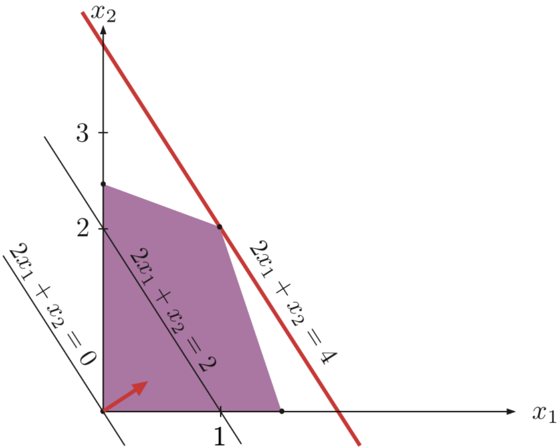
This optimization problem is an example of a *linear program (LP)*: It has a linear objective, and linear constraints.

Feasible region. The set of (x_1, x_2) satisfying the constraints is called the feasible region. It is a polygon defined by the intersection of halfspaces.



Resource allocation example (cont'd)

An optimum of the LP always lies at a vertex of the feasible region.



Linear Program

Generally, an LP can be written as follows[1]:

$$
\begin{aligned}
\min_{x\in\mathbb{R}^n} \quad & c^\top x \\
\text{subject to,} \quad & A_{\text{ub}}x \le b_{\text{ub}}, \\
& A_{\text{eq}}x = b_{\text{eq}}, \\
& \ell \le x \le u.
\end{aligned}
$$

Here $x \in \mathbb{R}^n$ is the optimization variable, $c \in \mathbb{R}^n$ are the objective coefficients, $A_{\text{ub}}x \le b_{\text{ub}}$, where $A_{\text{ub}} \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ are inequality constraints, $A_{\text{eq}}x = b_{\text{eq}}$, where $A_{\text{ub}} \in \mathbb{R}^{p\times n}, b \in \mathbb{R}^m$ are equality constraints, and ℓ, u, where $\ell, u \in \mathbb{R}^n$ bounds on variables

[1] This is the form expected by many solvers. The equality constraints and bounds can also be represented as inequality constraints. Hence, an LP can be succinctly written as $\min_x c^\top x$ s.t, $Ax \le b$.

Linear Program (cont'd)

Recall the LP from our example. Let us write this LP in the form shown in the right.

$$\begin{aligned} \max_{x_1, x_2 \in \mathbb{R}} \quad & 2x_1 + x_2 \\ \text{subject to} \quad & 4x_1 + x_2 \leq 6, \\ & x_1 + 2x_2 \leq 5, \\ & x_1, x_2 \geq 0. \end{aligned} \qquad \begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{subject to,} \quad & A_{\text{ub}} x \leq b_{\text{ub}}, \\ & A_{\text{eq}} x = b_{\text{eq}}, \\ & \ell \leq x \leq u. \end{aligned}$$

Solution: The optimization variable is $x = [x_1, x_2]$. We can then write,

$$c = \begin{bmatrix} -2 \\ -1 \end{bmatrix}, \quad A_{\text{ub}} = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix}, \quad b_{\text{ub}} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \quad \ell = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad u = \begin{bmatrix} \infty \\ \infty \end{bmatrix}.$$

Solving LPs with SciPy *(read at home)*

You can use your favorite library to solve an LP (*e.g.*, SciPy, CVXPY, PuLP, PuLP). The following example shows how to do so using SciPy's 'linprog' function.

```
from scipy.optimize import linprog
c = [-2, -1]
A_ub = [[4, 1],
        [1, 2]]
b_ub = [6, 5]
bounds = [(0, None), (0, None)]
res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=bounds)
print("Optimal x1, x2:", res.x)
print("Max objective:", -res.fun)
```

SciPy returns 'res.x' (optimal decision variables) and 'res.fun' (optimal value for minimization). If you have equality constraints, they can be passed via the `A_eq, b_eq` arguments.

Solving maximin objectives via LPs

Often, we can frame optimization problems with nonlinear objectives as LPs, with some work. One such use case (relevant to games), occurs when we have maximin objectives of the following form:

$$\max_{x \in \mathbb{R}^m} \quad \min_{j \in [n]} \sum_{i=1}^{m} c_{i,j}\, x_i,$$
$$\text{subject to,} \quad x \in \mathcal{X},$$

where, $\mathcal{X}$ is a polygon (can be expressed by linear constraints).

Here, we have linear constraints, but the objective $f(x) = \min_{j \in [n]} \sum_{i=1}^{m} c_{i,j}\, x_i$ is nonlinear due to the minimum over j.

Solving maximin objectives via LPs (cont'd)

Question: How do we convert this to an LP?
Key idea: The $f(x) = \min_j \sum_i c_{i,j} x_i$ is the smallest of several linear functions of x. If we introduce an auxiliary variable $t \in \mathbb{R}$ that is a lower bound on all those linear functions, then maximizing t will make it equal to that minimum.

Formally, we will introduce n additional constraints

$$t \le \sum_{i=1}^{m} c_{i,j}\, x_i \quad \text{for all } j \in [n].$$

If we maximize t subject to these constraints, it will match the true minimum.

Solving maximin objectives via LPs (cont'd)

Original minimization problem:

$$\max_{x \in \mathbb{R}^m} \quad \min_{j \in [n]} \sum_{i=1}^{m} c_{i,j}\, x_i\,,$$

$$\text{subject to,} \quad x \in \mathcal{X},$$

Our LP has the following form:

$$\begin{aligned} \max_{x \in \mathbb{R}^m, t \in \mathbb{R}} \quad & t \\ \text{subject to} \quad & t \le \sum_{i=1}^{m} c_{i,j}\, x_i \quad \text{for } j \in [n], \\ & x \in \mathcal{X}. \end{aligned}$$

Quiz

Which of the following other optimization problems can be formulated as LPs? In all cases, $\mathcal{X}$ is a polygon.

1. $\displaystyle \min_{x \in \mathbb{R}^m} \max_{j \in [n]} \sum_{i=1}^{m} c_{i,j} \, x_i, \quad \text{subject to } x \in \mathcal{X}.$
2. $\displaystyle \max_{x \in \mathbb{R}^m} \max_{j \in [n]} \sum_{i=1}^{m} c_{i,j} \, x_i, \quad \text{subject to } x \in \mathcal{X}.$
3. $\displaystyle \min_{x \in \mathbb{R}^m} \min_{j \in [n]} \sum_{i=1}^{m} c_{i,j} \, x_i, \quad \text{subject to } x \in \mathcal{X}.$

Ch 3.2: Computing safe strategies via LPs

Consider a finite two player game with action sets $\mathcal{A}_1 = [m]$ and $\mathcal{A}_2 = [n]$. Let $\mathcal{S}_1 = \Delta_m = \{x \in \mathbb{R}_+^m; \mathbf{1}^\top x = 1\}$ and $\mathcal{S}_2 = \Delta_n$ be the strategy spaces. Let the utilities be given by $u_i : [m] \times [n] \to \mathbb{R}$, for $i \in \{1, 2\}$.

Question: How do we compute the safe strategies for both players?

Let us first recall the definition of safe strategies in a two player game: Let $g_1 : \mathcal{S}_1 \to \mathbb{R}$, where $g_1(x) = \min_{y \in \Delta_n} u_1(x, y)$ denote the lowest possible utility P1 could achieve by choosing x, over the strategies y of P2. A strategy $\widetilde{x}$ is safe for P1 if $g_1(\widetilde{x}) \geq g_1(x)$ for all $x \in \Delta_m$.
We can likewise define the safe strategy $\widetilde{y}$ for P2.

Computing safe strategies via LPs (cont'd)

Let $U^{(1)} \in \mathbb{R}^{m\times n}$ such that $U^{(1)}_{i,j} = u_1(i,j)$. Hence, if $\widetilde{x}$ is safe for P1, we have,

$$
\begin{aligned}
& g_1(\widetilde{x}) \geq g_1(x) \qquad \text{for all } x \in \Delta_m, \quad \texttt{by definition} \\
\iff \quad & \min_{y\in\Delta_n} u_1(\widetilde{x}, y) \geq \min_{y\in\Delta_n} u_1(x,y) \qquad \text{for all } x \in \Delta_m. \\
\iff \quad & \min_{y\in\Delta_n} \widetilde{x}^\top U^{(1)} y \geq \min_{y\in\Delta_n} x^\top U^{(1)} y \qquad \text{for all } x \in \Delta_m. \qquad (1)
\end{aligned}
$$

Fact. Let $c \in \mathbb{R}^n$. The minimum of the linear function $c^\top y$ over the simplex $y \in \Delta_n$ satisfies, $\min_{y\in\Delta_n} c^\top y = \min_{j\in[n]} c_j$. (Why?)

Therefore, we can write (1) as,

$$
\begin{aligned}
(1) \quad \iff \quad & \min_{j\in[n]} \left({U^{(1)}}^\top \widetilde{x} \right)_j \geq \min_{j\in[n]} \left({U^{(1)}}^\top x \right)_j \qquad \text{for all } x \in \Delta_m. \\
\iff \quad & \min_{j\in[n]} \sum_{i=1}^m u_1(i,j)\, \widetilde{x}_i \;\geq\; \min_{j\in[n]} \sum_{i=1}^m u_1(i,j)\, x_i \quad \text{for all } x \in \Delta_m.
\end{aligned}
$$

Computing safe strategies via LPs (cont'd)

Therefore, we can write computing the safe strategies as the following problem:

$$\widetilde{x} = \operatorname*{argmax}_{x \in \Delta_m} \min_{j \in [n]} \sum_{i=1}^{m} u_1(i,j)\, x_i$$

The domain $x \in \Delta_m$ can be expressed as linear constraints $(x \geq 0, \mathbf{1}^\top x = 1)$, but the inner term of the objective is not linear in x.

Fortunately, we saw that we can formulate such problems as the following LP:

$$\begin{aligned} \max_{x \in \mathbb{R}^m, v \in \mathbb{R}} \quad & v \\ \text{subject to} \quad & v \leq \sum_{i=1}^{m} u_1(i,j)\, x_i \quad \text{for all } j \in [n] \\ & \mathbf{1}^\top x = 1, \\ & x \geq 0. \end{aligned}$$

Example

Write LPs to compute both players' safe strategies in the following example.

P1 \ P2	**1**	**2**
1	$(4,0)$	$(1,3)$
2	$(2,1)$	$(3,2)$
3	$(0,4)$	$(5,1)$

For player 1:

$$\begin{aligned} \max_{x\in\mathbb{R}^3, v\in\mathbb{R}} \quad & v \\ \text{subject to} \quad & v \le 4x_1 + 2x_2 + 0x_3, \\ & v \le 1x_1 + 3x_2 + 5x_3, \\ & \mathbf{1}^\top x = 1, \quad x \ge 0. \end{aligned}$$

For player 2:

$$\begin{aligned} \max_{y\in\mathbb{R}^2, v\in\mathbb{R}} \quad & v \\ \text{subject to} \quad & v \le 0y_1 + 3y_2, \\ & v \le 1y_1 + 2y_2, \\ & v \le 4y_1 + 1y_2, \\ & \mathbf{1}^\top y = 1, \quad y \ge 0. \end{aligned}$$

Quiz

1. Consider any feasible x, *i.e.*, $x \in \Delta_m$. What is an intuitive interpretation of the maximum value of v which satisfies all constraints? That is,

$$v \leq \sum_{i=1}^{m} u_1(i,j)\, x_i \qquad \text{for all } j \in [n]$$

2. Recall the safe strategy for P2, $\widetilde{y} = \max_{y \in \Delta_n} \min_{x \in \Delta_m} u_2(x, y)$. Here, $u_2(x, y) = \sum_{i,j} x_i y_j \cdot u_2(i, j)$. Write an LP to compute $\widetilde{y}$.

Quiz (cont'd): state if the following statements are true or false

3. You can use LPs to compute Nash equilibria in a two player zero sum game.

4. You can use LPs to compute safe strategies in an n player game.

Ch 3.3: Computing CEs via LPs

We will now use LPs to compute correlated equilibria (CE). Let us first recall the definition of a CE.

Review (CE). Recall the following definitions: There are n players in a normal form game. The action space of player i is $\mathcal{A}_i$. Let $\mathcal{A} = \bigtimes_{i=1}^{n} \mathcal{A}_i$. We can write the utility of player i as $u_i(a) = u_i(a_i, a_{-i})$.

Definition (CE). A joint distribution $s \in \Delta(\mathcal{A})$ is a correlated equilibirum if, for all $i \in [n]$, and any $a_i \in \mathcal{A}_i$ and $a_i' \neq a_i$, we have $\mathbb{E}_{a \sim s}\left[u_i(a) \middle| a_i\right] \geq \mathbb{E}_{a \sim s}\left[u_i(a_i', a_{-i}) \middle| a_i\right]$.

Interpretation of a CE:

- The joint distribution $s \in \Delta(\mathcal{A})$ is revealed ahead of time to all players.
- A trusted third party draws $a \sim s$ and reveals a_i and only a_i to player i.
- Player i can follow a_i or choose any other action a_i'.
- Then, following a_i maximizes player i's utility, provided that others are following a_{-i}.

We will begin with a simple example.

Example: Computing CE in a two player game

Let us characterize all possible correlated equilibria in this two player game (left).

P1 \ P2	**1**	**2**
1	$(1, 8)$	$(3, 4)$
2	$(7, 5)$	$(2, 6)$

Joint distribution:

P1 \ P2	**1**	**2**
1	$s_{1,1}$	$s_{1,2}$
2	$s_{2,1}$	$s_{2,2}$

Consider a joint distribution (right), satisfying $\sum_{i,j} s_{i,j} = 1$ and $s_{i,j} \geq 0$.

Let us write the conditions for a CE. When P1 is told to follow action 1, deviating to action 2 should not be beneficial:

$$
\begin{aligned}
& \mathbb{E}_{a \sim s}\left[u_1(1, a_2) | a_1 = 1\right] \geq \mathbb{E}_{a \sim s}\left[u_1(2, a_2) | a_1 = 1\right] \\
\iff \quad & u_1(1, 1)\mathbb{P}_{a \sim s}(a_2 = 1 | a_1 = 1) + u_1(1, 2)\mathbb{P}_{a \sim s}(a_2 = 2 | a_1 = 1) \\
& \qquad \geq u_1(2, 1)\mathbb{P}_{a \sim s}(a_2 = 1 | a_1 = 1) + u_1(2, 2)\mathbb{P}_{a \sim s}(a_2 = 2 | a_1 = 1) \\
\iff \quad & 1 \cdot \frac{s_{1,1}}{s_{1,1} + s_{1,2}} + 3 \cdot \frac{s_{1,2}}{s_{1,1} + s_{1,2}} \geq 7 \cdot \frac{s_{1,1}}{s_{1,1} + s_{1,2}} + 2 \cdot \frac{s_{1,2}}{s_{1,1} + s_{1,2}}
\end{aligned}
$$

Example: Computing CE in a two player game (cont'd)

$$1 \cdot \frac{s_{1,1}}{s_{1,1}+s_{1,2}} + 3 \cdot \frac{s_{1,2}}{s_{1,1}+s_{1,2}} \geq 7 \cdot \frac{s_{1,1}}{s_{1,1}+s_{1,2}} + 2 \cdot \frac{s_{1,2}}{s_{1,1}+s_{1,2}}$$

$$\iff \quad 6 \cdot s_{1,1} - 1 \cdot s_{1,2} \leq 0.$$

When P1 is told to follow action 2, deviating to 1 should not be beneficial:

$$7 \cdot \frac{s_{2,1}}{s_{2,1}+s_{2,2}} + 2 \cdot \frac{s_{2,2}}{s_{2,1}+s_{2,2}} \geq 1 \cdot \frac{s_{2,1}}{s_{2,1}+s_{2,2}} + 3 \cdot \frac{s_{2,2}}{s_{2,1}+s_{2,2}} \iff -6s_{2,1} + 1s_{2,2} \leq 0.$$

When P2 is told to follow action 1, deviating to 2 should not be beneficial:

$$8 \cdot \frac{s_{1,1}}{s_{1,1}+s_{2,1}} + 5 \cdot \frac{s_{2,1}}{s_{1,1}+s_{2,1}} \geq 4 \cdot \frac{s_{1,1}}{s_{1,1}+s_{2,1}} + 6 \cdot \frac{s_{2,1}}{s_{1,1}+s_{2,1}} \iff -4s_{1,1} + 1s_{2,1} \leq 0.$$

When P2 is told to follow action 2, deviating to 1 should not be beneficial:

$$4 \cdot \frac{s_{1,2}}{s_{1,2}+s_{2,2}} + 6 \cdot \frac{s_{2,2}}{s_{1,2}+s_{2,2}} \geq 8 \cdot \frac{s_{1,2}}{s_{1,2}+s_{2,2}} + 5 \cdot \frac{s_{2,2}}{s_{1,2}+s_{2,2}} \iff 4s_{1,2} - 1s_{2,2} \leq 0.$$

Example: Computing CE in a two player game (cont'd)

P1 \ P2	**1**	**2**
1	$(1, 8)$	$(3, 4)$
2	$(7, 5)$	$(2, 6)$

Joint distribution:

P1 \ P2	**1**	**2**
1	$s_{1,1}$	$s_{1,2}$
2	$s_{2,1}$	$s_{2,2}$

Summary. The following *linear* constraints characterize all CE in this game:

$$s_{1,1} + s_{1,2} + s_{2,1} + s_{2,2} = 1, \qquad s_{i,j} \geq 0,$$

$$6 \cdot s_{1,1} - 1 \cdot s_{1,2} \leq 0. \qquad -6s_{2,1} + 1s_{2,2} \leq 0.$$

$$-4s_{1,1} + 1s_{2,1} \leq 0. \qquad 4s_{1,2} - 1s_{2,2} \leq 0.$$

This is an example of a linear satisfiability problem. One approach to find a feasible point is to use a dummy objective in an LP, *e.g.*, $c = \mathbf{0}$ or c chosen randomly. (There are other methods too.)

Example: Computing CE in a two player game (cont'd)

P1 \ P2	**1**	**2**
1	$(1, 8)$	$(3, 4)$
2	$(7, 5)$	$(2, 6)$

Joint distribution:

P1 \ P2	**1**	**2**
1	$s_{1,1}$	$s_{1,2}$
2	$s_{2,1}$	$s_{2,2}$

We may want to find a "good" CE. For example, consider the welfare:

$$\begin{aligned} W(s) = u_1(s) + u_2(s) &= \mathbb{E}_{a\sim s}[u_1(a_1, a_2)] + \mathbb{E}_{a\sim s}[u_2(a_1, a_2)] \\ &= 9s_{1,1} + 7s_{1,2} + 12s_{2,1} + 8s_{2,2}. \end{aligned}$$

To find a CE which maximizes the welfare, we can maximize $W(s)$, which is linear in s, subject to the previous constraints in an LP.

Computing CEs in an n player game

Consider an n player game. The action space of player i is $\mathcal{A}_i$. Let $\mathcal{A}, \mathcal{A}_{-i}$ have their usual meanings. We can write the utility of player i as $u_i(a) = u_i(a_i, a_{-i})$.

Let us write an LP to find the welfare maximizing CE. In a CE, when recommending action $a_i = a_i'$, player i should not deviate to any other action $a_i = a_i''$. That is,

$$\begin{aligned}
& \mathbb{E}_{a\sim s}\left[u_i(a_i, a_{-i}) | a_i = a_i'\right] \geq \mathbb{E}_{a\sim s}\left[u_i(a_i'', a_{-i}) | a_i = a_i'\right] \\
\iff & \sum_{a_{-i}\in\mathcal{A}_{-i}} u_i(a_i', a_{-i}) \underbrace{\frac{s(a_i', a_{-i})}{\sum_{\widetilde{a}_{-i}\in\mathcal{A}_{-i}} s(a_i', \widetilde{a}_{-i})}}_{=\mathbb{P}_{a\sim s}(a_{-i}|a_i=a_i')} \geq \sum_{a_{-i}\in\mathcal{A}_{-i}} u_i(a_i'', a_{-i}) \underbrace{\frac{s(a_i', a_{-i})}{\sum_{\widetilde{a}_{-i}\in\mathcal{A}_{-i}} s(a_i', \widetilde{a}_{-i})}}_{=\mathbb{P}_{a\sim s}(a_{-i}|a_i=a_i')} \\
\iff & \sum_{a_{-i}\in\mathcal{A}_{-i}} \left(u_i(a_i'', a_{-i}) - u_i(a_i', a_{-i})\right) s(a_i', a_{-i}) \leq 0.
\end{aligned}$$

Computing CEs in an n player game (cont'd)

This leads to the following LP to find a CE which maximizes welfare:

$$\begin{aligned}
\max_{s} \quad & W(s) \triangleq \sum_{a \in \mathcal{A}} s(a) \left(\sum_{i=1}^{n} u_i(a) \right) \\
\text{subject to} \quad & \sum_{a \in \mathcal{A}} s(a) = 1, \\
& s(a) \geq 0 \quad \text{for all } a \in \mathcal{A}, \\
& \sum_{a_{-i} \in \mathcal{A}_{-i}} \big(u_i(a_i'', a_{-i}) - u_i(a_i', a_{-i}) \big)\, s(a_i', a_{-i}) \leq 0, \quad \forall\, i, a_i' \in \mathcal{A}_i, a_i'' \neq a_i'.
\end{aligned}$$

Question: Suppose we have n players, and each player has m possible actions, *i.e.*, $|\mathcal{A}_i| = m$. How many variables and constraints do we have in this LP?

Exercise *(try at home)*

The usual definition for the welfare $W(s) = \sum_{i=1}^{n} u_i(s)$, which sums all agent utilities, is also called the utilitarian welfare. It is just one way to quantify how well all agents are collectively doing. Another option is to consider the egalitarian welfare, which is defined as the minimum utility among all agents:

$$W_{\mathrm{egal}}(s) = \min_{i \in [n]} u_i(s).$$

Write an LP to find a W_{egal}-maximizing correlated equilibrium.

Ch 3.4: Computing CCEs via LPs

We will now use LPs to compute coarse correlated equilibria (CCE). Let us recall the definition of a CCE.

Review (CCE). Recall the following definitions: There are n players in a normal form game. The action space of player i is $\mathcal{A}_i$. Let $\mathcal{A} = \bigtimes_{i=1}^{n} \mathcal{A}_i$. We can write the utility of player i as $u_i(a) = u_i(a_i, a_{-i})$.

Definition. A joint distribution $s \in \Delta(\mathcal{A})$ is a CCE if, for all $i \in [n]$, and any $a_i' \in \mathcal{A}_i$, we have

$$\mathbb{E}_{a \sim s}\left[u_i(a)\right] \geq \mathbb{E}_{a \sim s}\left[u_i(a_i', a_{-i})\right].$$

Interpretation:

- The distrbution s is known ahead of time to the players.

- Each player i can choose their own alternative action a_i', or agree to a contract where a trusted third party will draw $a \sim s$ and play a_i on behalf of player i.

Computing CCEs via LPs

In a CCE, following s should be no worse for agent i than choosing some action a_i':

$$\mathbb{E}_{a\sim s}\left[u_i(a)\right] \geq \mathbb{E}_{a\sim s}\left[u_i(a_i', a_{-i})\right] \iff \sum_{a\in\mathcal{A}} u_i(a_i, a_{-i})s(a) \geq \sum_{a\in\mathcal{A}} u_i(a_i', a_{-i})s(a)$$

$$\iff \sum_{a\in\mathcal{A}} \left(u_i(a_i', a_{-i}) - u_i(a_i, a_{-i})\right) s(a) \leq 0.$$

This yields the following LP to find a CCE which maximizes welfare:

$$\begin{aligned} \max_{s} \quad & W(s) \triangleq \sum_{a\in\mathcal{A}} s(a) \left(\sum_{i=1}^{n} u_i(a)\right) \\ \text{subject to} \quad & \sum_{a\in\mathcal{A}} s(a) = 1, \quad s(a) \geq 0 \ \ \forall\, a \in \mathcal{A}, \\ & \sum_{a\in\mathcal{A}} \left(u_i(a_i', a_{-i}) - u_i(a_i, a_{-i})\right) s(a) \leq 0 \quad \forall\, i, a_i' \in \mathcal{A}_i. \end{aligned}$$

Example: Computing CCE in a two player game

Write the linear constraints which characterize all CCE in this two player game (left).

P1 \ P2	1	2
1	$(1,8)$	$(3,4)$
2	$(7,5)$	$(2,6)$

Joint distribution:

P1 \ P2	1	2
1	$s_{1,1}$	$s_{1,2}$
2	$s_{2,1}$	$s_{2,2}$

Consider a joint distribution (right), satisfying $\sum_{i,j} s_{i,j} = 1$ and $s_{i,j} \geq 0$. The expected utilities of P1 and P2 are:

$$u_1(s) = \mathbb{E}_{a\sim s}[u_1(s)] = 1 \cdot s_{1,1} + 3 \cdot s_{1,2} + 7 \cdot s_{2,1} + 2 \cdot s_{2,2},$$
$$u_2(s) = \mathbb{E}_{a\sim s}[u_2(s)] = 8 \cdot s_{1,1} + 4 \cdot s_{1,2} + 5 \cdot s_{2,1} + 6 \cdot s_{2,2}.$$

P1 following s should be no worse than choosing action 1 ahead of time:

$$\begin{aligned} u_1(s) &\geq \mathbb{E}_{a\sim s}[u_1(1, a_{-i})] = u_1(1,1)\mathbb{P}(a_2 = 1) + u_1(1,2)\mathbb{P}(a_2 = 2) \\ &= 1 \cdot (s_{1,1} + s_{2,1}) + 3(s_{1,2} + s_{2,2}), \\ \iff \quad & -6 \cdot s_{2,1} + 1 \cdot s_{2,2} \geq 0. \end{aligned}$$

Example: Computing CCE in a two player game

P1 following s should be no worse than choosing action 2 ahead of time:

$$
\begin{aligned}
& u_1(s) \geq \mathbb{E}_{a\sim s}[u_1(2, a_{-i})] = 7(s_{1,1} + s_{2,1}) + 2(s_{1,2} + s_{2,2}), \\
\iff \quad & 6 \cdot s_{1,1} - 1 \cdot s_{1,2} \leq 0.
\end{aligned}
$$

P2 following s should be no worse than choosing action 1 ahead of time:

$$
\begin{aligned}
& u_2(s) \geq \mathbb{E}_{a\sim s}[u_2(1, a_{-i})] = 8(s_{1,1} + s_{1,2}) + 5(s_{2,1} + s_{2,2}), \\
\iff \quad & 4s_{1,2} - 1 \cdot s_{2,2} \leq 0.
\end{aligned}
$$

P2 following s should be no worse than choosing action 2 ahead of time:

$$
\begin{aligned}
& u_2(s) \geq \mathbb{E}_{a\sim s}[u_2(2, a_{-i})] = 4(s_{1,1} + s_{1,2}) + 6(s_{2,1} + s_{2,2}), \\
\iff \quad & -4 \cdot s_{1,1} + 1 \cdot s_{2,1} \leq 0.
\end{aligned}
$$

Final remarks

- Linear programs can be solved in polynomial time. With p variables and q constraints, interior point methods have worst-case complexity $\mathcal{O}(p^2 q^{3/2})$. Some recent methods are more efficient, both in theory and practice.
- But p, q can be very large: with n players, and m actions each, we have $p = m^n$ variables, $q = nm(m-1) + m^n + 1$ constraints for computing a CE.
- Roadmap for the next two chapters:
  - Ch4: Introduction to online learning
  - Ch5: Using online learning to approximate CE and CCE in general sum games and NE in zero sum games.