CS 6840: Algorithmic Game Theory **Spring 2017**

Lecture 9: February 13

Lecturer: Éva Tardos *Scribe: Seung Won (Wilson) Yoo*

9.1 Computability of CCE

We begin with the computational tractability of CCE. We claim that computing CCE is polynomial in the number of strategy vectors. Let p_s be the probability of strategy vector $\vec{s} = (s_1 \cdots s_k)$. Since $p_s \geq 0$, $\sum_s p_s = 1$, p_s is indeed a probability. We can then encode the definition of a CCE as a linear program by the expressing the expectations

$$\mathbb{E}_{\vec{s} \sim \sigma}[c_i(\vec{s})] \leq \mathbb{E}_{\vec{s} \sim \sigma}[c_i(s'_i, \vec{s}_{-i})]$$

required for all players i and all strategies s'_i in the CCE definition in terms of p_s as:

$$\forall i, \forall s'_i \sum_{\vec{s}} p_{\vec{s}} c_i(\vec{s}) \leq \sum_{\vec{s}} p_{\vec{s}} c_i(s'_i, \vec{s}_{-i})$$

These constraints are now linear, so we can then enter this into any solver to solve for a CCE. Note that if we have k players and n strategies, we have nk constraints (other than the nonnegativity of the variables), but unfortunately, we have n^k variables.

As a quick aside, we wonder if we can do something like this with mixed Nash equilibria. If we define the variable $p^i_{s_i}$ as the probaility player i plays strategy s_i, we get the constraints

$$p_{\vec{s}} = \prod_i p^i_{s_i}, \vec{s} = (s_1 \cdots s_k)$$

The issue with this constraint, of using the product instead of $p_{\vec{s}}$ in the above inequalities is that the resulting system is no longer linear, and is not even convex in the variables.

In our linear program for CCE, two problems arise. First, since there was a variable for each strategy vector, when there are k players and n strategies, there ares n^k variables (which is exponential in the input size). Second, if this is a linear program, why are we not optimizing it to get the best CCE possible? We will get back to these questions.

9.2 Implementing CCE

We now turn to the question of how we can ask players to follow our CCE with a couple of simple games.

9.2.1 A Game of Chicken

The game of chicken is a two agent game with the following cost matrix:

	D	C
D	10,10	0,1
C	1,0	1,1

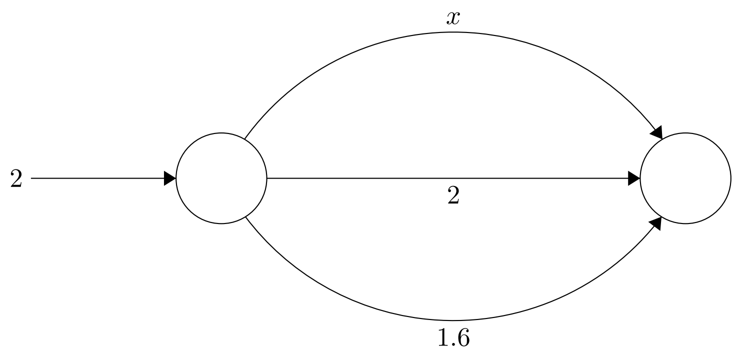
You can imagine the game as two teenagers racing cars towards each other, daring each other to swerve out of the way. The player who swerves faces humiliation (cost 1) but when both players don't swerve, there is a big cost (a crash, cost 10). The two strategies are Chicken (swerve) and Dare (don't swerve). The pure Nash equilibria are clearly $(D, C), (C, D)$. The mixed Nash equilibrium is when each player dares with probability 1/10 and chickens with probability 9/10. This gets the expected cost for the opponent paying dare to be $\frac{1}{10} \cdot 10 + \frac{9}{10} \cdot 0 = 1$, the same as the payoff for Chicken, so doing the same mixed strategy for both players is a Nash equilibrium.

However, there are other CCE equilibria. For example a CCE in this game is to randomize uniformly between the two PNE, implementable with a traffic light. Both players in this scenario will follow what the traffic light tells them to do because the alternative is worse:

- Told to C $\implies$ other player is told D $\implies$ switching will increase cost to 10
- Told to D $\implies$ other player is told C $\implies$ switching will increase cost to 1

9.2.2 A Problematic Routing Game

However, the situation is not always so nice. Imagine the following atomic routing game:



The pure nash equilibrium is when one person goes on the top route and the other person goes on the bottom route. The total cost is 2.6. The CCE when we choose with probability 1/2-1/2 between the two pure Nash equilibria has an expected cost of 1.3 for each player. Now, imagine a bad CCE where we assign one person to top and the other person to middle edge uniformly at random choosing which is which. We then get

$$\mathbb{E}[c_i(s)] = \frac{1}{2} * 2 + \frac{1}{2} * 1 = 1.5$$

Furthermore, for each of the possible alternate strategies x, 1.6, and 2 , we have:

$$\mathbb{E}[c_i(s_i^x, s_{-i})] = \frac{1}{2} * 2 + \frac{1}{2} * 1 = 1.5$$
$$\mathbb{E}[c_i(s_i^{1.6}, s_{-i})] = 1.6$$
$$\mathbb{E}[c_i(s_i^2, s_{-i})] = 2$$

Hence this is a equilibrium under the CCE definition. However would a real person do this? What if you were to change behavior conditional on what CCE tells you to do? If you were suggested to take 2 it would always be beneficial to change to 1.6. Therefore, this is an example of a CEE that is not implementable.

9.3 Correlated Equilibria

This motivates the definition of the correlated equilibrium, which is a stronger version of the coarse correlated equilibrium. Again, let σ be a probability distribution on $\vec{s} = (s_1 \cdots s_k)$. σ is a correlated equilibria if:

$$\forall i, s_i', s_i'' \, \mathbb{E}_{s\sim\sigma}[c_i(s)|s_i = s_i'] \leq \mathbb{E}_{s\sim\sigma}[c_i(s_i'', s_{-i})|s_i = s_i']$$

This is then an implementable equilibria via advice, since we rule out switching conditional any advice the player is given. The CEE in our previous example is not a CE since it fails the inequality when $s' = 2, s'' = 1.6$. The CCE in the chicken game also happens to be a CE. Also, note that with a similar formulation as before, we can solve for correlated equilibria as a LP.

9.4 Hardness of Finding Best CCE

As a final example, we return to our point about finding the best CEE and show that it can be computationally difficult. Consider a very artificial game where we are given a 3CNF (conjuncitve normal form) formula ϕ and the players are the variables whose strategies are either $true$ or $false$. Let us set

$$c_i(s) = \begin{cases} 1 & \text{if } s \text{ satisfies } \phi \\ 0 & \text{otherwise} \end{cases}$$

Finding the best CCE is now equivalent to finding a satisfying assignment to ϕ. In fact this is also tbe best CE, the best mixed Nash, and the best PNE. Finding any of these involves deciding if the formula ϕ is satisfiable. We know that the 3-SAT problem is NP-Hard, so this reduction suggests that the problem of finding best CE is difficult.