INSE6441	Jia Yuan Yu

4: Dynamic games

Concordia	April 9, 2021

Example 0.1 (Dollar auction)**.** Bidders must pay theirs bids win or lose, iterate between bidders. Winner is player with the highest total bid. Winner takes all. Miminum incremental bid of one cent. Does first player have an advantage?

How should you compete with another player in a winner takes all game (e.g., for a marriage prospect)?

Example 0.2 (Ultimatum game)**.** Divide class into two groups at random: Proposers, responders. Proposers write down anonymously their split of a prize of 10 CAD, and a random password. Randomly give proposals to responders. Responders write accept or reject. If split is accepted, prize is paid, otherwise, neither player gets anything.

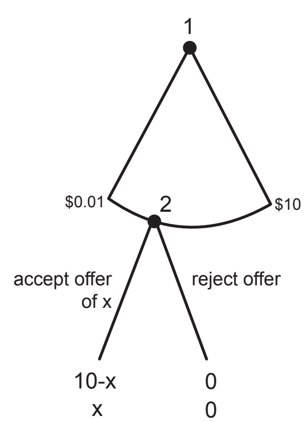

Figure 1: Ultimatum game tree.

We introduce dynamic game with non-simultaneous moves.

1 Computing equilibria (part 2)

In general, computing equilibria for games is hard. In the case of finite zero-sum two-player games, we can use linear programming to compute Nash equilibria.

1.1 Linear programming

Consider a two-player game where the player to one player is the negative of the payoff to the other player. Let A denote the $m \times n$ payoff matrix for the row player. The row player mixed strategy is $y = [y_1 y_2 \dots y_m]$. The column player mixed strategy is $x = [x_1 x_2 \dots x_n]$. The expected payoff to the column player is $y^T Ax$.

At a NE, the row player's strategy y must satisfy

$$\min_y y^T Ax,$$

hence, the column player must play x such that

$$\max_x \min_y y^T Ax. \tag{1}$$

Let e_i denote a unit vector with 1 in the i-th position. Observe that since Ax is a vector, we have $\min_y y^T Ax = \min_{i=1,\dots,m} e_i^T Ax$. Let $y^*(x)$ denote the best response for the row player (action that minimizes payoff to column player). If the row player plays $y^*(x)$, then

$$e_i^T Ax \geq y^*(x) Ax, \quad i = 1, \dots, m.$$

The problem (1) can be written as

$$\begin{aligned}
\max_{x,v} \quad & v \quad (= \min_{i=1,\dots,m} e_i^T Ax = y^*(x)Ax) \\
\text{s.t.} \quad & e_i^T Ax \geq v, \quad \text{for } i = 1, \dots, m, \\
& \sum_{j=1}^n x_j = 1, \\
& x_j \geq 0, \quad \text{for } i = 1, \dots, n.
\end{aligned}$$

Linear programs can be solved efficiently by many software packages[1].

2 Dynamic games

So far, we have considered static or simultaneous-action games. In this section, we consider games with dynamic actions. We introduce the notion of order of moves, and information available at the time of taking an action.

2.1 Stackelberg duopoly

Previously, we looked at competition among decision makers who make decisions simultaneously. We now look at sequential decisions. Heinrich von Stackelberg proposed the following model in 1934. Firm 1 is the leader and picks its production quantity q_1 first. The follower, Firm 2, then observes q_1 and picks its production quantity q_2 accordingly. Other that the distinction of the order of actions, we use the same assumptions as Cournot duopoly.

To find a Nash equilibrium in the Stackelberg duopoly, we find the best-response function for each firm by backward induction because the firms make decision sequentially, one after the other.

[1]Online solver: `https://www.math.ucla.edu/~tom/gamesolve.html`.

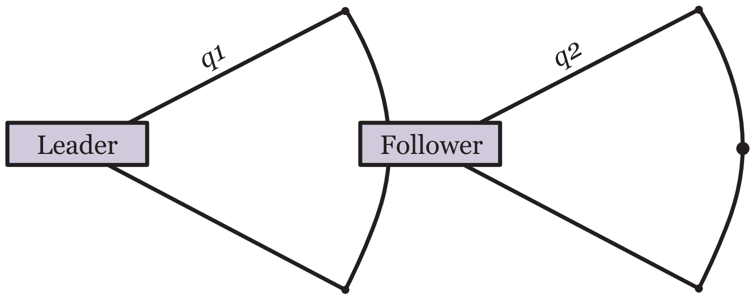

Figure 2: Stackelberg game tree

1. The follower's optimal decision (after observing q_1) is:

$$q_2^{\text{opt}}(q_1) = \arg\max_{q_2} \quad \pi(q_1 + q_2)q_2 - cq_2.$$

   This is a best-response function, as in Cournot duopoly. It can be found by setting the first derivative ($\frac{\mathrm{d}}{\mathrm{d}q_2}$) to zero. Try with $\pi(q) = 12 - q$.

2. The leader's optimal decision (knowing that the follower will respond to its decision q_1 according to the best-response function q_2^{opt}) is:

$$q_1^\star = \arg\max_{q_1} \quad \pi(q_1 + q_2^{\text{opt}}(q_1))q_1 - cq_1.$$

   This can also be found by setting the first derivative ($\frac{\mathrm{d}}{\mathrm{d}q_1}$) to zero.

How does $(q_1^\star, q_2^{\text{opt}}(q_1^\star))$ in Stackleberg duopoly compare with the Cournot case? The leader has an advantage and higher payoff.

2.2 Extensive form

The extensive form contains the following information:

1. set of players
2. order of moves: precedence relation (tree, player label at each non-terminal node)
3. payoffs: vectors at terminal nodes
4. available actions: labels on edges of the tree
5. available knowledge for each move: partition of nodes into information sets
6. exogeneous events: probabilities on edges (Nature, demand, private observations, etc.)

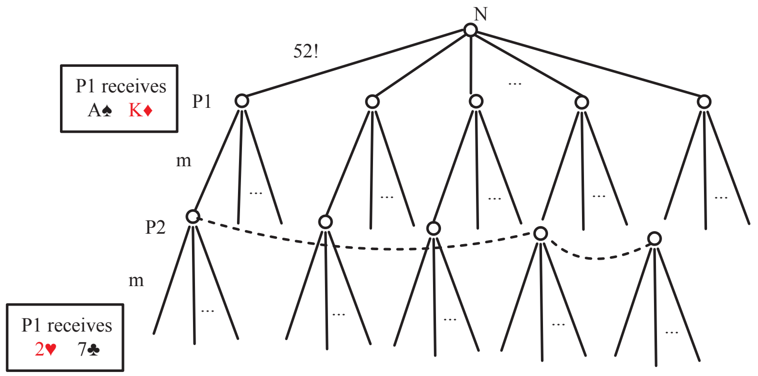

Figure 3: Poker as extensive form game

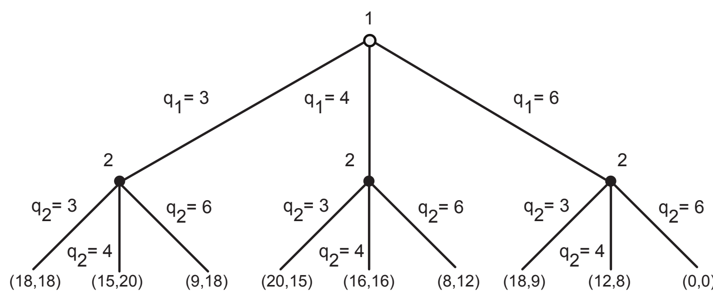

Figure 4: Stackelberg duopoly in extensive form: Firm 1 has a single information set, three actions, three pure strategies; player 2 has three information sets, three actions at each information set, 27 pure strategies.

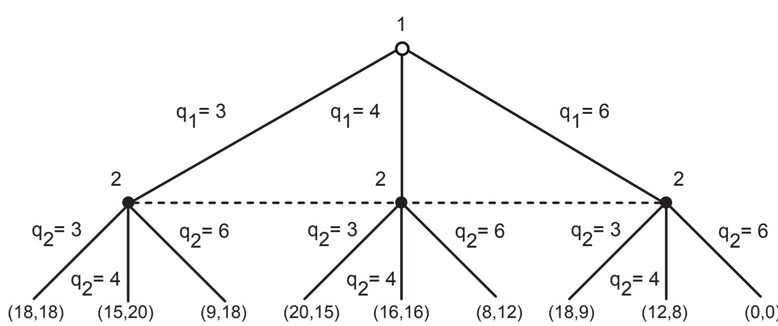

Figure 5: Cournot duopoly in extensive form.

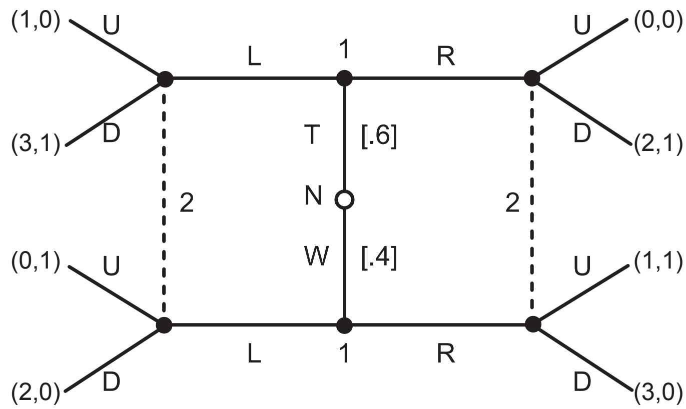

Figure 6: Nature chooses the type of player 1.

Strategies are defined slightly differently. Let H_i denote the set of information sets of player i; let $A(h_i)$ denote the possible actions at information set h_i and $A_i = \cup_{h_i \in H_i} A(h_i)$. A pure strategy is a mapping $s_i : H_i \to A_i$ from information sets to actions, such that

$$s_i(h_i) \in A(h_i) \quad \text{for all } h_i.$$

Each s_i takes a value in the Cartesian product $S_i = \times_{h_i \in H_i} A(h_i)$.

In an extensive-form game, a player can randomize its actions at each information set. If these random actions are independents (random variables), then we say that the player follows a behavior strategy. A behavior strategy b_i for player i in an extensive-form game is a mapping from information set h_i to a probability distribution $\Delta(A(h_i))$ over the actions $A(h_i)$. These take values in the Cartesian product $B_i = \times_{h_i \in H_i} \Delta(A(h_i))$.

Pure Nash equilibria are defined in terms of $(s_1, \ldots, s_I)$. Nash equilibria in behavior strategies are defined in terms of $(b_1, \ldots, b_I)$.

Remark 1. An extensive-form game has the same pure strategies as a strategic-form game with the above set of strategies $S_1, \ldots, S_I$. However, a behavior strategy takes values in B_i in contrast to a mixed strategy over $S_1 \times \ldots \times S_I$ (a much bigger space).

Theorem 2.1 (Kuhn, cf. section 3.4 of FT)**.** *In games of perfect recall (each information set is a single node in the game tree), for every mixed strategy over $S_1 \times \ldots \times S_I$, there exists an equivalent behavior strategy that generates outcomes with the same probabilities, and vice versa.*

Remark 2. It follows that a Nash equilibrium always exists in an extensive game with perfect recall.

3 Subgame-perfect equilibrium

Although Nash equilibria are well-defined to dynamic games, we can refine the notion of Nash equilibria to a special class of Nash equilibria that better describe reality. We introduce the notion of subgame-perfect equilibrium.

Definition 3.1 (Subgame)**.** A proper subgame of a game tree T is a subtree of T (with a single root node) such that two nodes of the subtree are in the same information set if and only if they are in the same information set in T.

Definition 3.2 (SPE)**.** A behavior strategy profile σ is a subgame-perfect equilibrium if for every proper subgame G, the restriction of σ to G is a Nash equilibrium of G.

Remark 3. A SPE is a Nash equilibrium. In finite games of perfect information, SPE can be found by backward induction.

Example 3.1 (Backward induction with multiple players)**.** Consider a finite game with perfect information. We can draw the game tree and specify a pure Nash equilibrium by backward induction.

4 Reading material

- Chapter 3 of Fudenberg and Tirole.