Harvard CS 1360, Spring 2025
Instructor: Ariel Procaccia
Scribe: Taig Singh

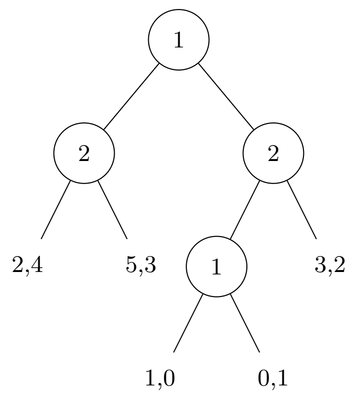
Extensive-Form Games

—

Lecture 3

Definition 1 (Extensive-Form Games)**.** In an extensive-form game, moves are done sequentially, not simultaneously. The game forms a tree where nodes are labeled by players, edges represent the strategies of players, and leaves show payoffs. Games start at the root of the tree and at each node, the node's corresponding player chooses a strategy out of the edges that come out of the node.

Example 1 (Basic example of an extensive-form game)**.** Consider the following graph representation of an extensive-form game.

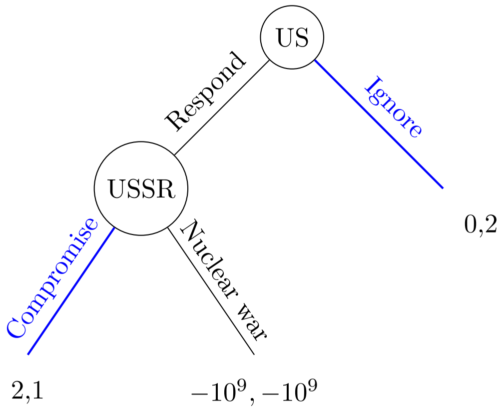


Every path from root to leaf in the tree represents a combination of strategies of players 1 and 2 that results in a payoff. This tree describes the following payoff structure:

- If player 1 chooses *left*, then player 2 chooses *left*, player 1 gets 2 and player 2 gets 4.
- If player 1 chooses *left*, then player 2 chooses *right*, player 1 gets 5 and player 2 gets 3.
- If player 1 chooses *right*, then player 2 chooses *left*, then player 1 chooses *left*, player 1 gets 1 and player 2 gets 0.
- If player 1 chooses *right*, then player 2 chooses *left*, then player 1 chooses *right*, player 1 gets 0 and player 2 gets 1.
- If player 1 chooses *right*, then player 2 chooses *right*, player 1 gets 3 and player 2 gets 2.

So how do extensive-form games compare to normal-form games? Consider the following (very abstract) model of the Cuban Missile Crisis which we will model as a game between the US and the USSR:

Example 2 (The Cuban Missile Crisis)**.** In this scenario, the USSR has positioned its missiles in Cuba and the US has two options: it can either ignore this move or it can respond. Further, if the US responds, the USSR has two options: it can either compromise or start a nuclear war. Consider the following extensive-form representation of this game, provided on the left below.



	Compromise	Nuclear war
Respond	2,1	$-10^9, -10^9$
Ignore	0,2	0,2

Note that if the US chooses to ignore, the payoff is $(0, 2)$. If the US chooses to respond and the USSR chooses to compromise, the payoff is $(1, 2)$ and if in this case the USSR chooses to engage in a nuclear war, the payoff is $(-10^9, -10^9)$.

Now, consider the equivalent normal-form representation of the game, provided on the right above. Note that the payoff is $(0, 2)$ if the US chooses to ignore regardless of the USSR's strategy, as the game always ends in this leaf if the US ignores. Now, note that the strategy profile (Ignore, Nuclear War), highlighted in blue in both representations, is a Nash equilibrium of this game. However, there is a problem: the USSR's threat of a nuclear war is not credible! That is because if the game ever ends up at the node in which the USSR must make a decision, the USSR will always choose to compromise because that will result in a payoff of 1 as compared to the -10^9 that they will receive if they choose to engage in a nuclear war.

The lack of credibility of the USSR's threat is due to the sequential nature of the game, which is only captured in the extensive-form representation of the game, not the normal-form representation. Although (Ignore, Nuclear War) is a Nash equilibrium of this game, the USSR would never choose to engage in a nuclear war after the US has chosen to respond, which leads us to believe that our notion of a Nash equilibrium has various issues in this setting. This leads us to the concept of a subgame-perfect equilibrium.

Definition 2 (Subgame-Perfect Equilibrium)**.** In an extensive-form game, each subtree forms a subgame. A set of strategies is a subgame-perfect equilibrium if it is a Nash equilibrium in each subgame.

Let's apply this to our previous example. In the subgame where the USSR chooses either Compromise or Nuclear war, Compromise is a Nash equilibrium as this is a one-player game where Compromise is a dominant strategy. Now, considering the larger subgame that starts with the US choosing Respond or Ignore, we see that the strategy profile (Respond, Compromise) is the only subgame-perfect equilibrium. We can check this by noting that if the US deviates to Ignore, their payoff will decrease from 2 to 0, and if the USSR deviates to Nuclear War, their payoff will decrease from 1 to -10^9.

An interesting phenomenon of extensive-form games is that agents may be able to improve their equilibrium payoff by eliminating their own strategies from the game. In our example, the USSR can improve their subgame-perfect equilibrium outcome by eliminating their Compromise strategy from the game. Note that in this case, the US will always choose Ignore, resulting in a payoff of 0 for the US and 2 for the USSR rather than choosing Respond which will result in a payoff of -10^9 for both players. Thus, by removing their Compromise strategy, the USSR increases their equilibrium payoff from 1 to 2.

So how do we solve for subgame-perfect equilibria? It's actually quite simple - we can proceed with backward induction by solving for the equilibria at the leaves of the tree and working up from there. At each step, we can simplify a subgame into the payoffs received by all players if the game ever reaches the root node of that subgame. Consider the following example:

Example 3 (Backward Induction)**.** Consider the extensive-form game depicted in Figure 1a below. In order to solve for the subgame-perfect equilibrium, start with the subgame represented by the subtree rooted at the highlighted node, where player 1 needs to choose between *left* and *right* receiving a payoff of 1 or 0 respectively. In the subgame-perfect equilibrium, if the game ever reaches this subgame, player 1's strategy will trivially be to choose $left$ and receive a payoff of 1. Thus, we can replace the subtree rooted at the highlighted node with the payoff that the game will result in if the game ever reaches that subgame, shown in Figure 1b as we replace the subtree with a $1, 0$.

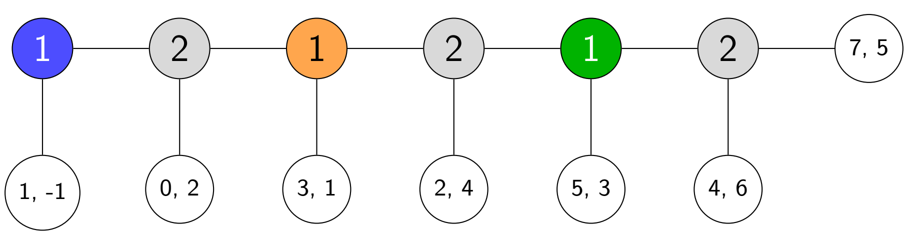
Now, considering the subgame represented by the subtree rooted at the new highlighted node in 1b, we see that player 2 must choose between $left$ and $right$ receiving a payoff of 4 or 3 respectively. In the subgame-perfect equilibrium, if the game ever reaches this subgame, player 2's strategy will trivially be to choose $left$ and receive a payoff of 4. Thus, we can replace the subtree rooted at this highlighted node with the payoff that the game will result in if the game ever reaches that subgame, shown in Figure 1c as we replace the subtree with a $2, 4$.

Continuing this process, we consider the subgame represented by the subtree rooted at the highlighted node in Figure 1c, we see that player 2 will always choose *right* in this subgame, so we replace this subtree with a $3, 2$ in Figure 1d. Finally, considering the final subgame represented by the subtree rooted at the highlighted node in Figure 1d, player 1 will always choose *right* in this subgame so we can replace this tree with a $3, 2$ as depicted in the first cell of Figure 1e, which is the payoff of the players in the subgame-perfect equilibrium. Note that to get the strategy profile of this subgame-perfect equilibrium, we can just keep track of the strategies we found for the two players in each subgame, as shown in Figure 1f.

Note that this isn't a proof that backward induction successfully identifies a subgame-perfect equilibrium, however, this process is reasonably intuitive as at every point in the game, a player is doing what is best for them given that the moves after that point will also be optimal for themselves. In fact, backward induction up to tie-breaking will give you all of the possible subgame-perfect equilibrium in a game. This process is easy enough if we have the representation of the game and we are able to eliminate nodes one by one. However, if you are faced with a very large game, this process is not computationally feasible at all.

Note that while subgame-perfect equilibria capture more nuances in player strategies than Nash equilibria do in extensive-form games, there are still some issues with them when looking at the empirical strategies of players in certain situations.

Example 4. Consider the following extensive-form game played between two players:

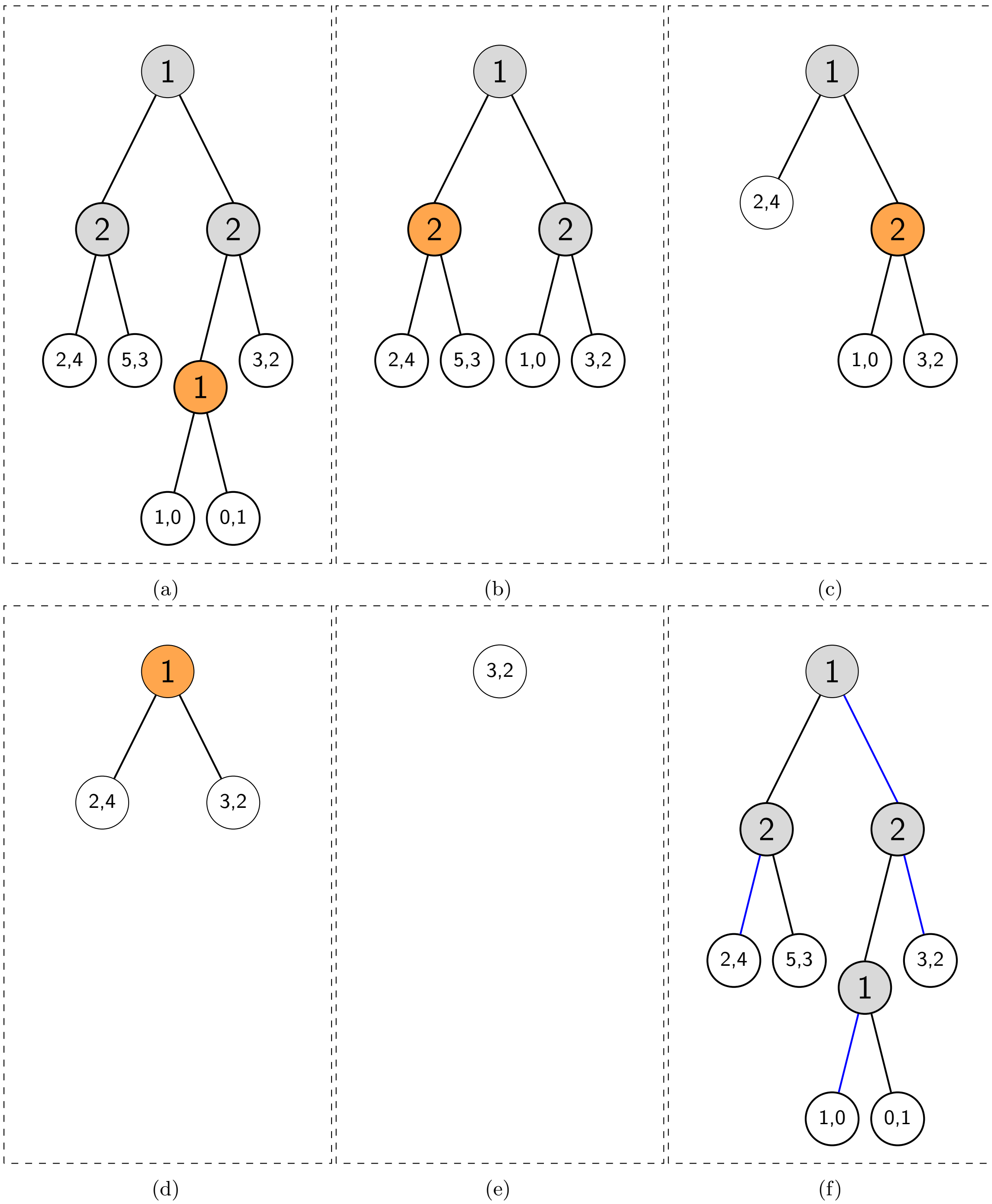


We can solve for the subgame-perfect equilibrium via backward induction. Starting at the smallest subgame, we see that player 2 will always choose *down* (payoff of 6) over *right* (payoff of 5) if the game ever reaches this subgame. Continuing up the tree to the subgame rooted at the green node, we see that player 1 will always choose *down* (payoff of 5) over *right* (payoff of 4 because player 2 will choose *down*). This process continues up the tree, and we get that at any subtree, the player whose turn it is will choose *down*. Thus, when the game starts, player 1 will choose *down* in the subgame-perfect equilibrium, resulting in a payoff of $(1, -1)$. However, because of the structure of the game, payoffs generally increase as the game continues down the tree and so both players are generally better off by letting the game continue rather than "defecting" and choosing *down* and some early point in the game.

If you were playing the game as player 1 against a classmate, where would you choose *down*? At the blue node, the orange node, or the green node? Although the subgame-perfect equilibrium suggests the answer to this question is going *down* on the blue node, most people would say otherwise.

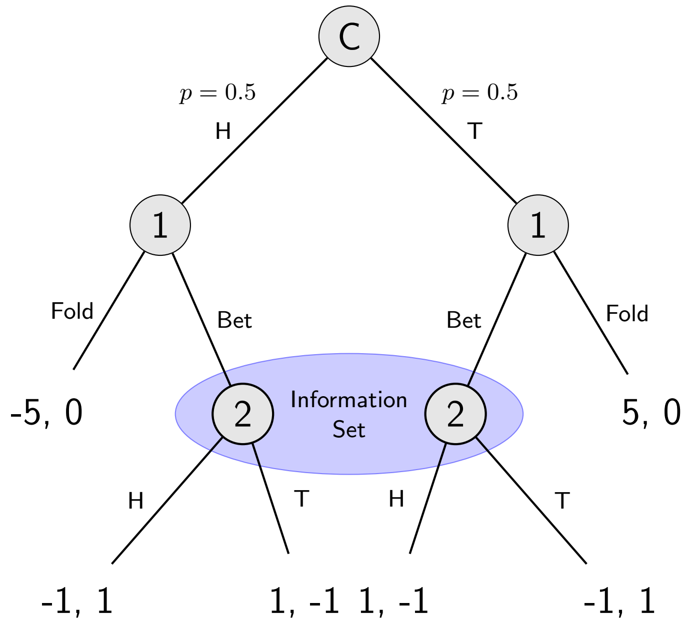
Definition 3 (Imperfect Information Games)**.** A chance node chooses between several actions according to a known probability distribution. An information set is a set of nodes that a player may be in, given the available information. A strategy must be identical for all nodes in an information set.

Example 5 (An Imperfect Information Game)**.** Consider the following imperfect information game:



1
2
2
2,4
5,3
1
3,2
1,0
0,1
(a)
1
2
2
2,4
5,3
1,0
3,2
(b)
1
2,4
2
1,0
3,2
(c)
1
2,4
3,2
(d)
3,2
(e)
1
2
2
2,4
5,3
1
3,2
1,0
0,1
(f)

Figure 1



The root node represents a chance node where a coin is flipped, with a 0.5 probability of H and a 0.5 probability of T. Player 1 observes the outcome of the flip before choosing their action. Player 1 can either "Bet" or "Fold." If Player 1 folds, the game ends with payoffs of -5 for Player 1 and 0 for Player 2 if the coin shows heads, and 5 for Player 1 and 0 for Player 2 if the coin shows tails. If Player 1 bets, the decision moves to Player 2, who does not know the outcome of the coin flip. Player 2 then chooses between "H" and "T." The payoffs vary depending on Player 2's choice and the actual coin flip, with payoffs of -1,1 or 1,-1.

Note that trivially, if the coin lands H then player 1 will Bet, and if the coin lands T then player 1 will Fold. Therefore, player 2 will always play H, because they will only ever make a decision if the coin lands on H. If the -5 and 5 payoffs were flipped, however, then player 2 will always play T. Thus, we see that it's impossible to compute the optimal strategy of a subgame in isolation in imperfect information games.

We will now move on to a 2-step special case.

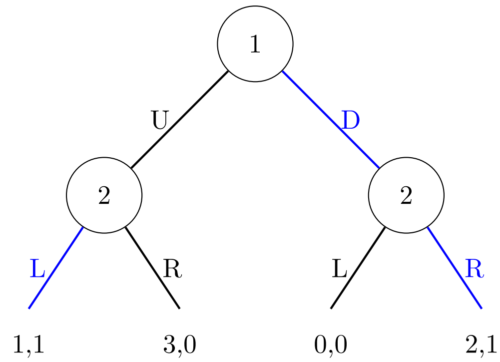
Definition 4 (Stackelberg Games)**.** In a Stackelberg game, there is a leader and a follower. The leader commits to a certain strategy and subsequently the follower observes the commitment and chooses a strategy.

Example 6 (A Basic Stackelberg Game)**.** Consider the following Stackelberg game, where the row player is the leader and the column player is the follower.

$$\begin{bmatrix} (1,1) & (3,0) \\ (0,0) & (2,1) \end{bmatrix}$$

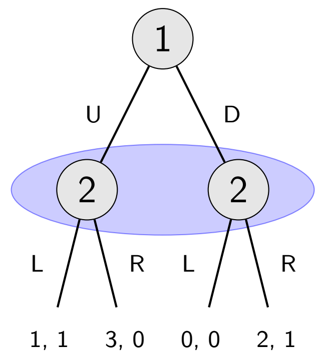
Here, playing up is a dominant strategy for the row player and so the column player would play left, and thus $(1,1)$ is the only Nash equilibrium outcome.

However, note that the row player can commit to playing down. Then, the follower will choose right, and the outcome will be $(2,1)$ and so the leader is strictly better off! Here, if the leader announces their commitment, the Stackelberg game can be rewritten as an extensive-form game of perfect information:



Here, the subgame-perfect equilibrium is highlighted in blue, which is achievable when the game is played as a Stackelberg game.

Now, what if the leader commits to a mixed strategy? What is the maximum payoff the leader can get with a mixed strategy, assuming that the follower breaks their ties in favor of the leader? This turns the game into the following imperfect information game:



If the leader plays *up* with probability p and *down* with probability $1 - p$, then the follower's expected payoff for choosing $left$ is

$$p \cdot 1 + (1 - p) \cdot 0 = p$$

and the follower's expected payoff for choosing $right$ is

$$p \cdot 0 + (1 - p) \cdot 1 = 1 - p$$

and so if $p > 0.5$ the follower will choose $left$ and if $p \leq 0.5$ the follower will choose $right$ (note that there is a tie at $p = 0.5$ and the leader is always better-off if the follower chooses $right$). Thus, if $p > 0.5$, the leader's expected payoff

$$p \cdot 1 + (1 - p) \cdot 0 = p$$

and if $p \leq 0.5$ the leader's expected payoff is just

$$p \cdot 3 + (1 - p) \cdot 2 = 2 + p$$

Thus, the leader's expected payoff is maximized when $p = 0.5$ at which point the leader's expected payoff is 2.5 which is greater than 2! Thus, the randomness helps the leader due to imperfect information.

Definition 5 (Stackelberg Equilibrium)**.** For a mixed strategy x_1 of the leader, define the best response set of the follower as

$$B_2(x_1) = \text{argmax}_{s_2 \in S} u_2(x_1, s_2)$$

In a strong Stackelberg equilibrium (SSE) the leader plays a mixed strategy in

$$\text{argmax}_{x_1 \in \Delta(S)} \max_{s_2 \in B_2(x_1)} u_1(x_1, s_2)$$

where $\Delta(S)$ is the set of all mixed strategies.

Now, we will see how an SSE can be computed via linear programming!

Example 7. Consider the following Stackelberg game where the row player is the leader and the column player is the follower:

$$\begin{bmatrix} (1,0) & (0,2) \\ (0,1) & (1,0) \\ (0,0) & (0,0) \end{bmatrix}$$

Note that playing *down* is weakly dominated for player 1, therefore we can essentially ignore this strategy, but we still include it to give player 1 more flexibility. The SSE will be a mixed strategy $s_1 = (p_1, p_2)$ for player 1, where p_1 is the probability of playing *up* and p_2 is the probability of playing *middle*. A best response for player 2 is either $left$ or $right$. We will set up two linear programs: one maximizing player 1's utility when player 2's best response is $left$ and one maximizing player 1's utility when player 2's utility is $right$. Then, whichever linear program achieves a higher objective function (player 1's utility), we will take player 1's corresponding mixed strategy and player 2's best response as the SSE.

We first consider the case where the follower's best response is $left$. This only happens when $p_2 \geq 2p_1$, because p_2 is the follower's utility for playing $left$ and $2p_1$ is the follower's utility for playing $right$. Further, the leader's utility when the follower plays $left$ is just p_1. Thus, we have the following linear program:

$$\begin{aligned} \max\ & p_1 \\ \text{s.t. } & p_2 \geq 2p_1, \\ & p_1 + p_2 \leq 1, \\ & p_1, p_2 \geq 0 \end{aligned}$$

The last two constraints come from the leader's mixed strategy being a probability distribution over the leader's strategies. The solution to this LP is $p_1 = \frac{1}{3}$ and $p_2 = \frac{2}{3}$, and the objective function is maximized at a value of $\frac{1}{3}$.

We now consider the case where the follower's best response is $right$. This happens when $p_2 \leq 2p_1$, for the analogous reasons as above. Here, the leader's utility when the follower plays $right$ is just p_2. Thus, we have the following linear program:

$$\begin{aligned} \max\ & p_2 \\ \text{s.t. } & p_2 \leq 2p_1, \\ & p_1 + p_2 \leq 1, \\ & p_1, p_2 \geq 0 \end{aligned}$$

The solution to this LP is also $p_1 = \frac{1}{3}$ and $p_2 = \frac{2}{3}$, and the objective function is maximized at a value of $\frac{2}{3}$. Note that in this case, both linear programs gave us the same optimal mixed strategy but the leader's utility is higher when the follower chooses $right$. Thus, the leader's optimal mixed strategy here makes the follower indifferent between $left$ and $right$, so assuming that the follower breaks ties in favor of the leader, the SSE here is $s_1 = (\frac{1}{3}, \frac{2}{3})$ and $s_2 = right$.

In Figure 2, you can find an illustration of the solution space that the linear programs induce, and where the optimal solution lies. The x-axis is p_1, the y-axis is p_2 and the solution space is limited by the constraint that $p_1 + p_2 \leq 1$ so all solutions must inside the large triangle. The orange region represents where the follower plays $right$, the blue region represents the region where the follower plays $left$, and these regions are divided by the line $p_2 = 2p_1$. Finally, note that the optimal solution exists on this indifference line at $(\frac{1}{3}, \frac{2}{3})$.

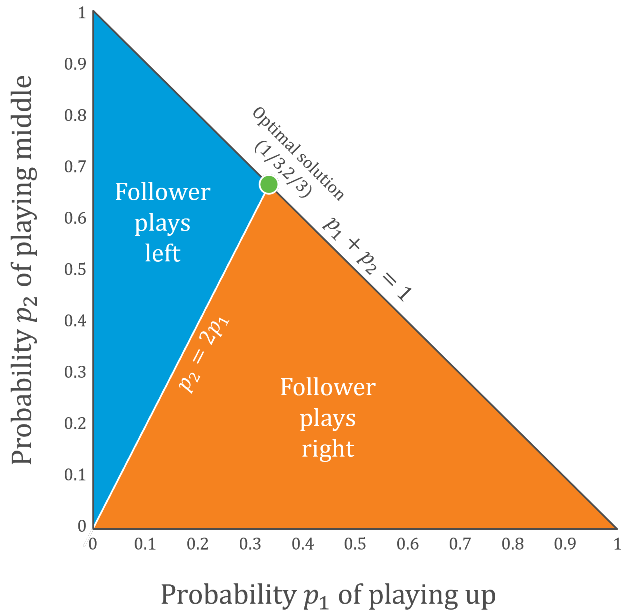


Figure 2

Now, consider slightly changing the payoff matrix to the following:

$$\begin{bmatrix} (1,0) & (\mathbf{2},2) \\ (0,1) & (1,0) \\ (0,0) & (0,0) \end{bmatrix}$$

This changes our second linear program (when the follower's best response is *right*) to the following:

$$\begin{aligned} \max\ & \mathbf{2p_1} + p_2 \\ \text{s.t. } & p_2 \leq 2p_1, \\ & p_1 + p_2 \leq 1, \\ & p_1, p_2 \geq 0 \end{aligned}$$

because now the leader receives a payoff of 2 when they play *up* and the follower plays *right*. This further changes the solution of this second LP to $s_1 = (1, 0)$ achieving an objective function of 2. Since this is higher than the objective function (leader's payoff) in the first LP, representing when the follower plays *left*, the SSE changes to $s_1 = (1, 0)$ and $s_2 = right$. A graphical depiction of this new game can be found in Figure 3. Note that the optimal solution has moved to $(1, 0)$ in this new game.

We will now provide a more general algorithm for computing SSEs. The leader's mixed strategy is defined by variables $x(s_1)$, which give the probability of playing each strategy $s_1 \in S$. For each follower strategy s_2^*, we compute a strategy x for the leader such that playing s_2^* is a best response for the follower, and under this constraint, x is optimal. This computation is done via the following LP:

$$\begin{array}{ll} \max & \sum_{s_1 \in S} x(s_1) u_1(s_1, s_2^*) \\ \text{s.t.} & \forall s_2 \in S, \sum_{s_1 \in S} x(s_1) u_2(s_1, s_2^*) \geq \sum_{s_1 \in S} x(s_1) u_2(s_1, s_2) \\ & \sum_{s_1 \in S} x(s_1) = 1 \end{array}$$

Finally, we take the x resulting from the "best" s_2^*.

We will end this lecture with a brief discussion about AI's role in game-playing. Over the last few years, we have seen many advances in AI game-playing. For example, poker is an extensive form game of

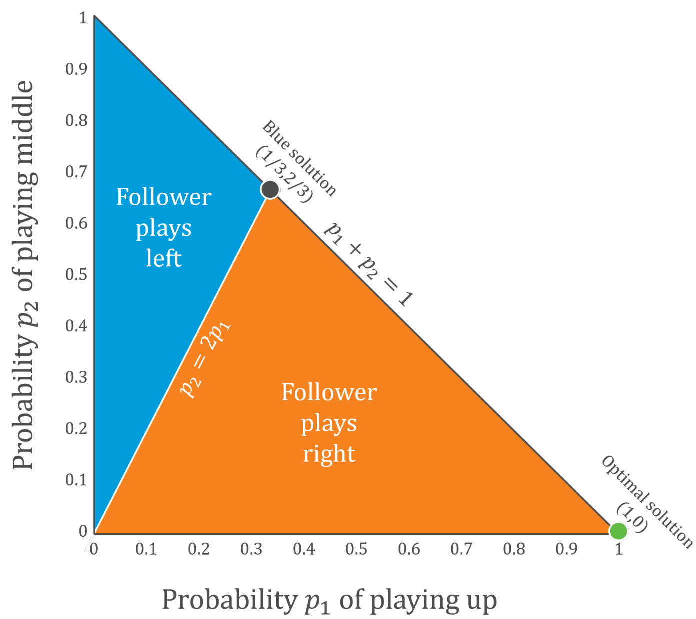


Figure 3

incomplete information because you do not know what cards your opponents are holding, and this introduces a lot of complexity in the game making it hard for AI. However, we have reached a point where we can solve extensive-form games with incomplete information and can do well in poker and beat humans. However, these algorithms can be used for more than just recreational games. We can use the same algorithms to solve problems that we care about in the real world. These algorithms are applicable to fields like negotiations and cybersecurity and can result in much better policies in these situations. At this point, determining what game we are playing in real-world situations and delineating the rules are far more difficult than actually playing the game well. Much of Professor Milind Tambe's work has been in this direction, as he has applied game theoretic algorithms in real-world situations like physical security and wildlife protection.