# CAP 5993/CAP 4993
# Game Theory

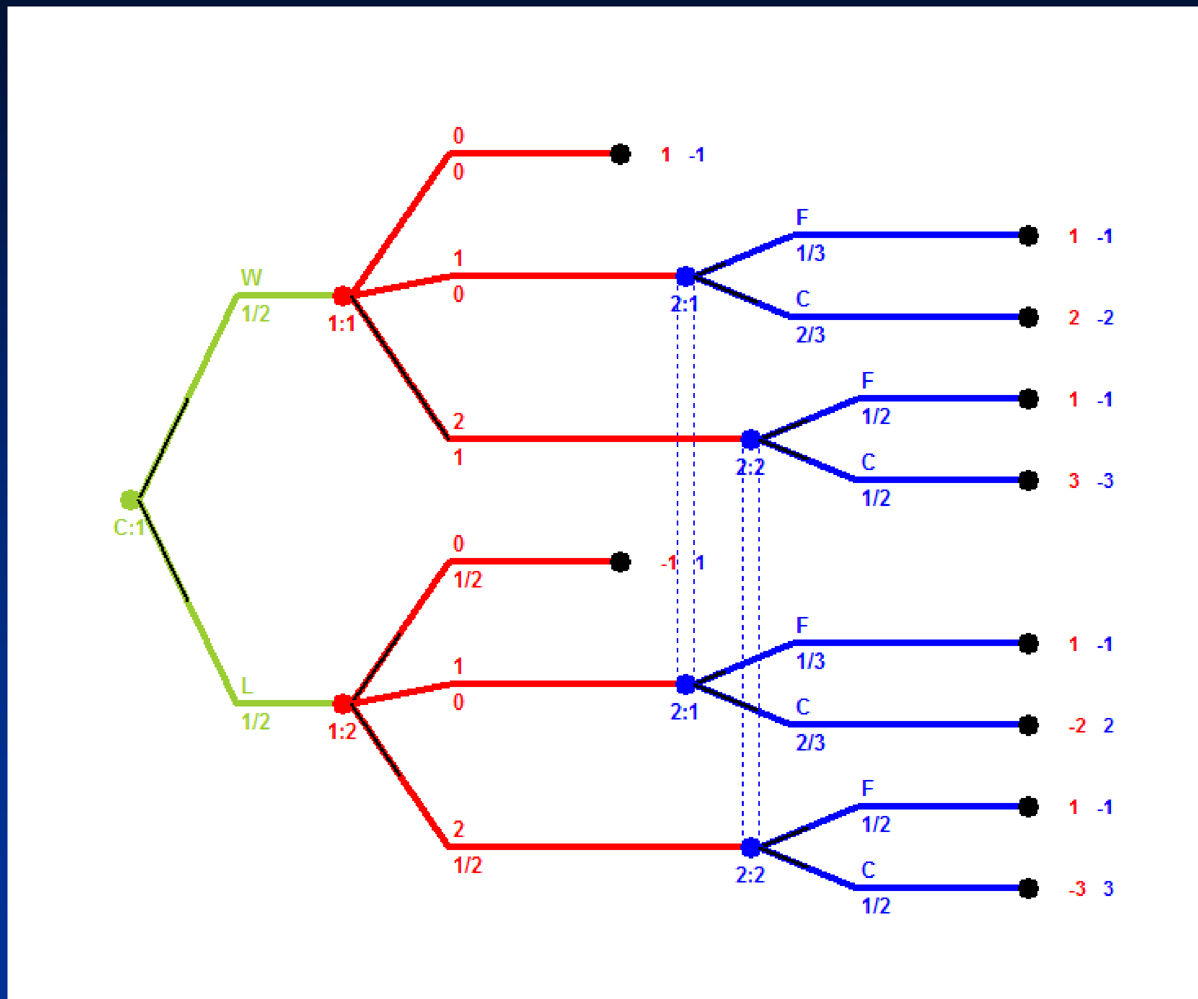
**Instructor: Sam Ganzfried**
**sganzfri@cis.fiu.edu**

# Announcements

- HW1
- HW2
- midterm

# Extensive-form game

- A game in **extensive form** is given by a *game tree*, which consists of a directed graph in which the set of vertices represents positions in the game, and a distinguished vertex, called the *root*, represents the starting position of the game. A vertex with no outgoing edges represents a terminal position in which play ends. To each terminal vertex corresponds an outcome that is realized when the play terminates at that vertex. Any nonterminal vertex represents either a chance move (e.g., a toss of a die or a shuffle of a deck of cards) or a move of one of the players. To any chance-move vertex corresponds a probability distribution over edges emanating from that vertex, which correspond to the possible outcomes of the chance move.


C:1
W
1/2
L
1/2
1:1
1:2
0
0
1 -1
1
0
2:1
F
1/3
1 -1
C
2/3
2 -2
2
1
2:2
F
1/2
1 -1
C
1/2
3 -3
0
1/2
-1 1
1
0
2:1
F
1/3
1 -1
C
2/3
-2 2
2
1/2
2:2
F
1/2
1 -1
C
1/2
-3 3

# Perfect vs. imperfect information

- To describe games with imperfect information, in which players do not necessarily know the full board position (like poker), we introduce the notion of *information sets*. An information set of a player is a set of decision vertices of the player that are indistinguishable by him given his information at that stage of the game. A game of *perfect information* is a game in which all information sets consist of a single vertex. In such a game whenever a player is called to take an action, he knows the exact history of actions and chance moves that led to that position.

- A *strategy* of a player is a function that assigns to each of his information sets an action available to him at that information set. A path from the root to a terminal vertex is called a *play* of the game. When the game has no chance moves, any vector of strategies (one for each player) determines the play of the game, and hence the outcome. In a game with chance moves, any vector of strategies determines a probability distribution over the possible outcomes of the game.

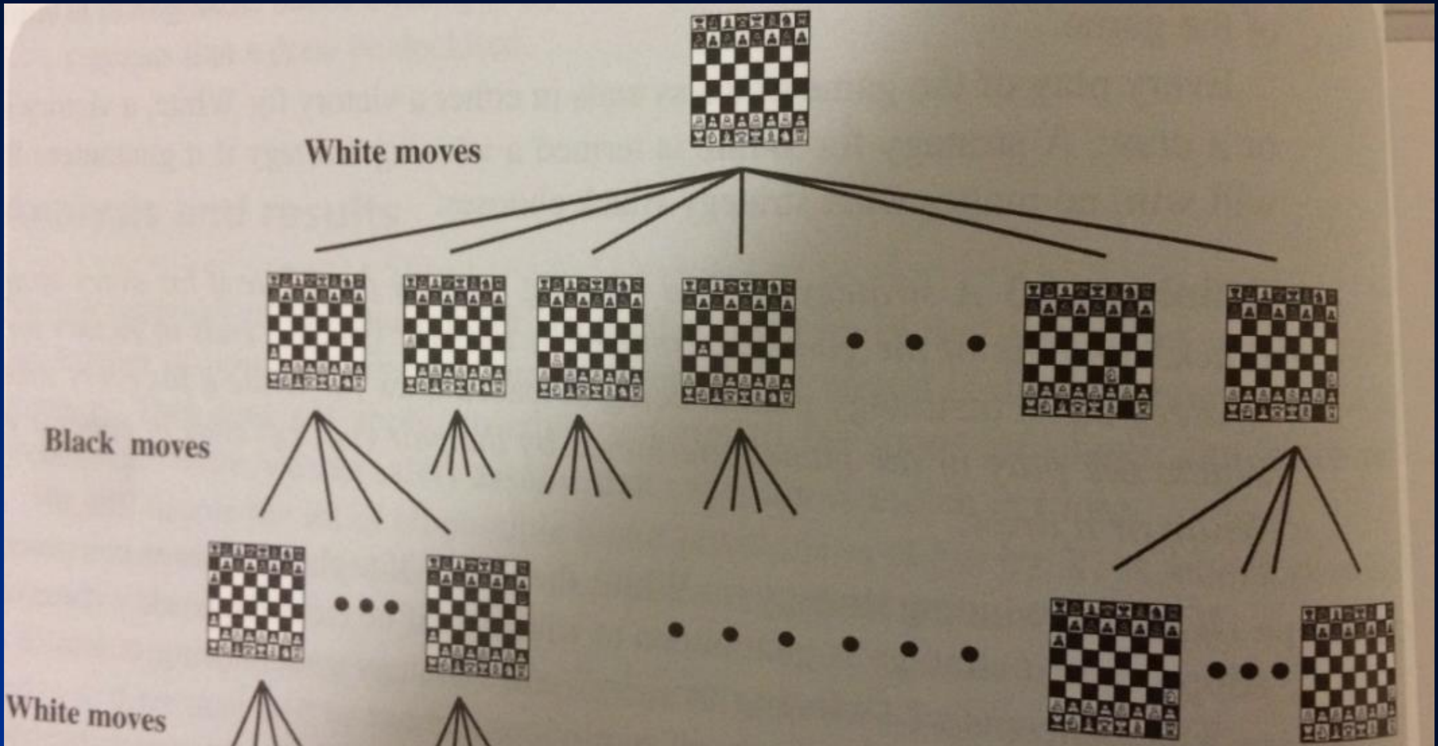


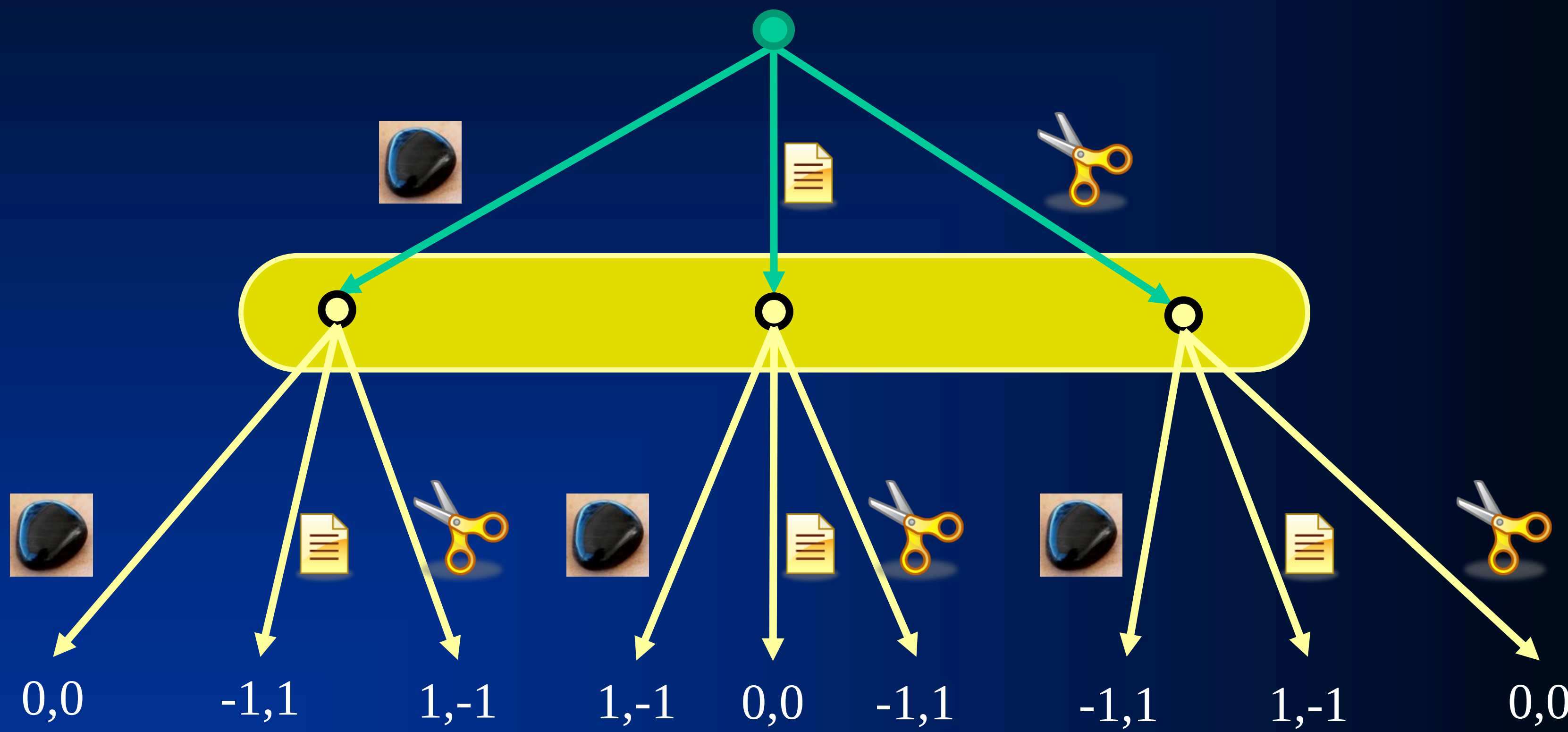
Figure 1.1 The game of chess presented in extensive form

- Theorem (von Neumann [1928]) In every two-player game (with perfect information) in which the set of outcomes is O = {Player 1 wins, Player 2 wins, Draw}, one and only one of the following three alternatives holds:
  1. Player 1 has a winning strategy.
  2. Player 2 has a winning strategy.
  3. Each of the two players has a strategy guaranteeing at least a draw.

# Chance moves

- In the games we have seen so far, the transition from one state to another is always accomplished by actions undertaken by the players. Such a model is appropriate for games such as chess and checkers, but not for card games or dice games (such as poker or backgammon) in which the transition from one state to another may depend on a chance process: in card games, the shuffle of the deck, and in backgammon, the toss of the dice. It is possible to come up with situations in which transitions from state to state depend on other chance factors, such as the weather, earthquakes, or the stock market. These sorts of state transitions are called *chance moves*. To accommodate this feature, our model is expanded by labeling some of the vertices in the game tree $(V, E, x_0)$ as chance moves. The edges emanating from vertices corresponding to chance moves represent the possible outcomes of a lottery, and next to each such edge is listed the probability that the outcome it represents will be the result of the lottery.

# Rock-paper-scissors



# Strategies in imperfect-information games

- A strategy of player i is a function from each of his information sets to the set of actions available at that information set.
- Just as in games with chance moves and perfect information, a strategy vector determines a distribution over the outcomes of a game.

- Theorem (Kuhn) Every finite game with perfect information has at least one pure strategy Nash equilibrium.

- Corollary of Nash's Theorem: Every extensive-form game (of perfect or imperfect information) has an equilibrium in mixed strategies.

- Let’s solve some games!!
- Gambit -- http://www.gambit-project.org/
  - Strategic-form and extensive-form games
  - Dominance, equilibrium, quantal response equilibrium
  - Remove iteratively weakly/strictly dominated strategies
  - Remove iteratively weakly/strictly “dominated actions” in extensive-form games

# Quantal response equilibrium

- **Quantal response equilibrium** (**QRE**) provides an equilibrium notion with bounded rationality. QRE is not an equilibrium refinement, and it can give significantly different results from Nash equilibrium.
- In a quantal response equilibrium, players are assumed to make errors in choosing which pure strategy to play. The probability of any particular strategy being chosen is positively related to the payoff from that strategy. In other words, very costly errors are unlikely.
- The equilibrium arises from the realization of beliefs. A player's payoffs are computed based on beliefs about other players' probability distribution over strategies. In equilibrium, a player's beliefs are correct.

- By far the most common specification for QRE is **logit equilibrium (LQRE)**.

$$P_{ij} = \frac{\exp(\lambda EU_{ij}(P_{-i}))}{\sum_k \exp(\lambda EU_{ik}(P_{-i}))}$$

- $P_{ij}$ is probability of player i choosing strategy j.
- The parameter $\lambda$ is non-negative (sometimes written as $1/\mu$).
- $\lambda$ can be thought of as the rationality parameter. As $\lambda \to 0$, players become "completely non-rational", and play each strategy with equal probability. As $\lambda \to \infty$, players become "perfectly rational," and play approaches a Nash equilibrium.

# Dominance

- Remove iteratively weakly/strictly dominated strategies
- Remove iteratively weakly/strictly "dominated actions" in extensive-form games

- Compute one Nash equilibrium
  - Solving Linear program
  - Simplicial subdivision
  - Tracing logit equilibria

- Compute as many Nash equilibria as possible
  - By solving a linear complementarity program
  - By looking for pure strategy equilibria
  - By minimizing the Lyapunov function
  - By global Newton tracing
  - By iterated polymatrix approximation
  - By solving a system of polynomial equations

- Compute all Nash equilibria
  - By enumerating extreme points

# Extensive-form games

- Compute one Nash equilibrium
  - Using the extensive-form game
  - Using the strategic-form game
- Some of the algorithms appropriate for just extensive vs. strategic form

# Prisoner's dilemma

| | **C** | **D** |
|---|---|---|
| C | 4, 4 | 0, 5 |
| D | 5, 0 | 1, 1 |

# Battle of the sexes

| | Opera | Football |
|---|---|---|
| Opera | 3,2 | 0,0 |
| Football | 0,0 | 2,3 |

*Battle of the Sexes 1*

| | Opera | Football |
|---|---|---|
| Opera | 3,2 | 1,1 |
| Football | 0,0 | 2,3 |

*Battle of the Sexes 2*

# Rock-paper-scissors

| | **rock** | **paper** | **scissors** |
|---|---|---|---|
| Rock | 0,0 | -1, 1 | 1, -1 |
| Paper | 1,-1 | 0, 0 | -1,1 |
| Scissors | -1,1 | 1,-1 | 0,0 |

# Security game

- Random strategy:
  - *Increase cost/uncertainty to attackers*

**Adversary**

**Defender**

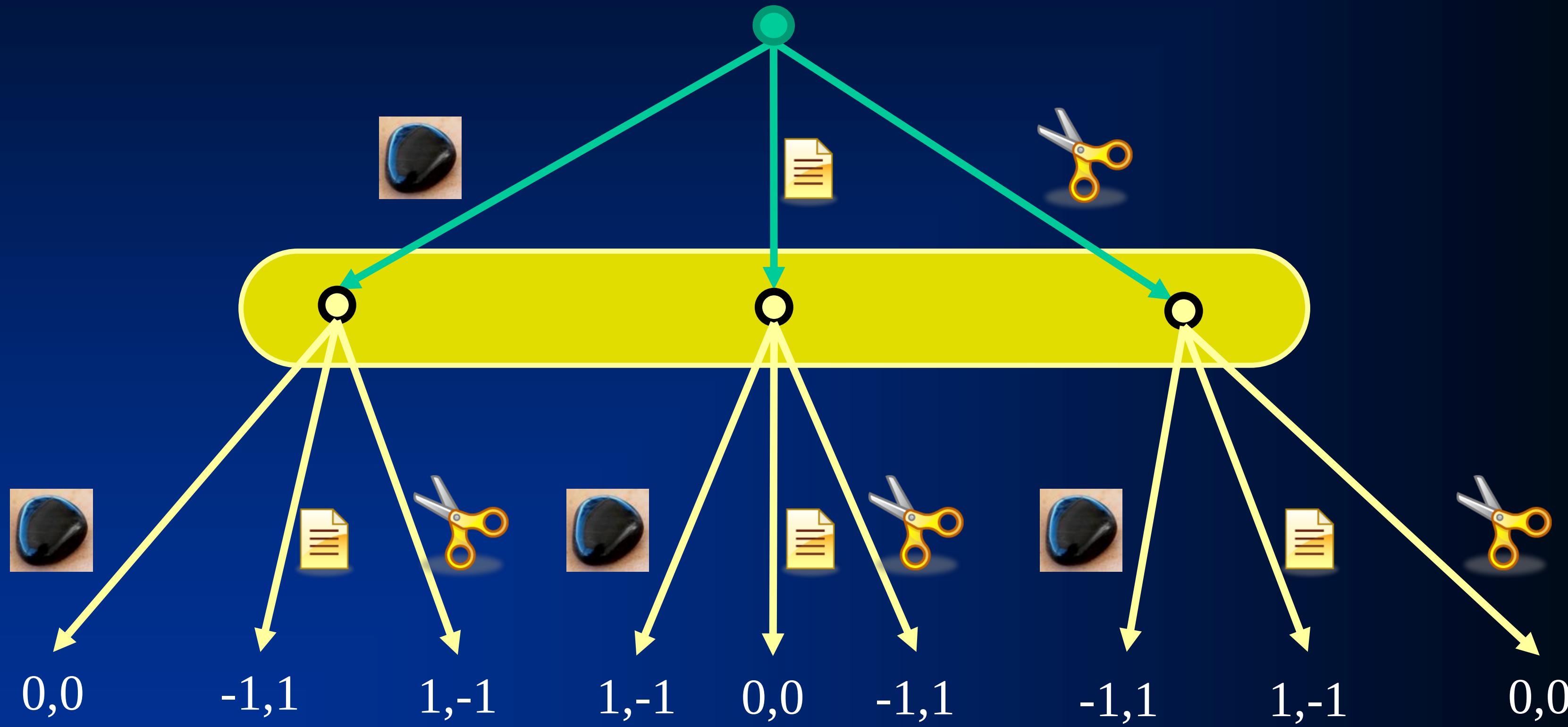
| | Target #1 | Target #2 |
|---|---|---|
| Target #1 | 4, -3 | -1, 1 |
| Target #2 | -5, 5 | 2, -1 |

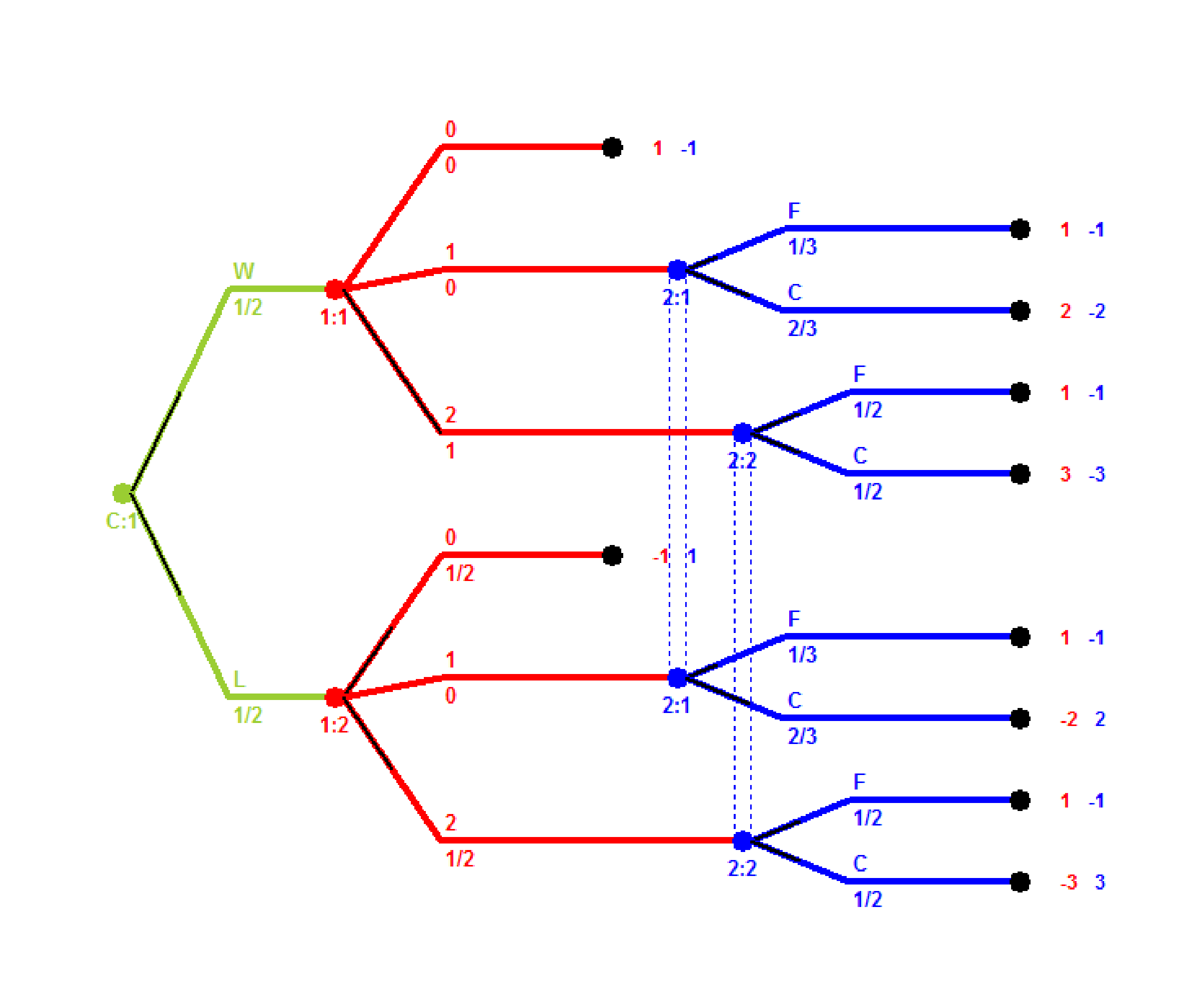
# Chicken

| | Swerve | Straight |
|---|---|---|
| Swerve | 0, 0 | -1, +1 |
| Straight | +1, -1 | -10, -10 |

*Fig. 2: Chicken with numerical payoffs*

# Rock-paper-scissors





C:1
W
1/2
L
1/2
1:1
1:2
0
0
1
0
2
1
0
1/2
1
0
2
1/2
1 -1
-1 1
2:1
2:2
2:1
2:2
F
1/3
C
2/3
F
1/2
C
1/2
F
1/3
C
2/3
F
1/2
C
1/2
1 -1
2 -2
1 -1
3 -3
1 -1
-2 2
1 -1
-3 3

| WL/12 | CC | CF | FC | FF |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | -0.5 | -0.5 | 1 | 1 |
| 02 | -1 | 1 | -1 | 1 |
| 10 | … | | | |
| 11 | | | | |
| 12 | | | | |
| 20 | | | | |
| 21 | | | | |
| 22 | | | | |

- “Because of the field’s widespread applicability and the variety of mathematical and computational issues it encompasses, it is hard to place game theory within any single discipline (although it has traditionally been viewed as a branch of economics). While the field is clearly benefitting from being analyzed from many different perspectives, it is also important to make sure that it doesn’t become disorganized as a result. When I started doing research for my thesis, I was surprised at how difficult it was to find a basic introduction to the fundamental mathematical and computational results. I had to turn to game theory textbooks for proofs of classical results, operations research and optimization books for results in linear programming and linear complementarity, and more recent computer science and economics papers for algorithms and complexity results. Thus, the major contribution of this paper is to present the basic mathematical and computational results related to computing Nash equilibria in a coherent form that can benefit people from all fields.”
- http://www.cs.cmu.edu/~sganzfri/SeniorThesis.pdf

# Computing Nash equilibria of two-player zero-sum games

- Consider the game $G = (\{1,2\}, A_1 \times A_2, (u_1, u_2))$.
- Let $U^*_i$ be the expected utility for player i in equilibrium (the value of the game); since the game is zero-sum, $U^*_1 = - U^*_2$.
- Recall that the Minmax Theorem tells us that $U^*_1$ holds constant in all equilibria and that it is the same as the value that player 1 achieves under a minmax strategy by player 2.
- Using this result, we can formulate the problem of computing a Nash equilibrium as the following optimization:

| | | |
|---|---|---|
| Minimize | $U^*_1$ | |
| Subject to | $\Sigma_{k \text{ in } A2}\, u_1,(a^j_1, a^k_2) * s^k_2 <= U^*_1$ | for all j in $A_1$ |
| | $\Sigma_{k \text{ in } A2}\, s^k_2 = 1$ | |
| | $s^k_2 >= 0$ | for all k in $A_2$ |

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

- Note that all of the utility terms $u_1(*)$ are constants while the mixed strategy terms $s^k_2$ and $U^*_1$ are variables.

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

- First constraint states that for every pure strategy j of player 1, his expected utility for playing any action j in $A_1$ given player 2's mixed strategy $s_1$ is at most $U^*_1$. Those pure strategies for which the expected utility is exactly $U^*_1$ will be in player 1's best response set, while those pure strategies leading to lower expected utility will not.
- As mentioned earlier, $U^*_1$ is a variable; we are selecting player 2's mixed strategy in order to minimize $U^*_1$ subject to the first constraint. Thus, player 2 plays the mixed strategy that minimizes the utility player 1 can gain by playing his best response.

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

- The final two constraints ensure that the variables $s^k_2$ are consistent with their interpretation as probabilities. Thus, we ensure that they sum to 1 and are nonnegative.

| | L | C | R |
|---|---|---|---|
| T | 3, -3 | -5, 5 | -2, 2 |
| M | 1, -1 | 4, -4 | 1, -1 |
| B | 6, -6 | -3, 3 | -5, 5 |

- v_ = 1 and v^ = 1. Player 1 can guarantee that he will get a payoff of a least 1 (using the maxmin strategy M), while player 2 can guarantee that he will pay at most 1 (by way of minmax strategy R).
- So the value v=1.

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all $j$ in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all $k$ in $A_2$

Minimize $U^*_1$

Subject to $3 * s^1_2 + (-5) * s^2_2 + (-2) * s^3_2 <= U^*_1$

$1 * s^1_2 + 4 * s^2_2 + 1 * s^3_2 <= U^*_1$

$6 * s^1_2 + (-3) * s^2_2 + (-5) * s^3_2 <= U^*_1$

$s^1_2 + s^2_2 + s^3_2 = 1$

$s^1_2 >= 0,\ s^2_2 >= 0,\ s^3_2 >= 0$

# Linear programs

- A *linear program* is defined by:
  - a set of real-valued variables
  - a linear objective function (i.e., a weighted sum of the variables)
  - a set of linear constraints (i.e., the requirement that a weighted sum of the variables must be less than or equal to some constant).
- Let the set of variables be $\{x_1, x_2, \ldots, x_n\}$, which each $x_i$ in R. The objective function of a linear program, given a set of constraints $w_1, w_2, \ldots, w_n$, is

  Maximize $\Sigma^n_{i=1} w_i x_i$

- Linear programs can also express minimization problems: these are just maximization problems with all weights in the objective function negated.
- Constraints express the requirement that a weighted sum of the variables must be greater or equal to some constant. Specifically, given a set of constants $a_{1j}, \ldots, a_{nj}$, and a constant $b_j$, a constraint is an expression
  $\Sigma^n_{i=1} a_{ij} x_i <= b_j$

$$\Sigma^{n}_{i=1} a_{ij}x_i <= b_j$$

- By negating all constraints we can express greater-than-or-equal constraints.
- By providing both less-than-or-equal and greater-than-or-equal constraints with the same constants, we can express equality constraints.
- By setting some constants to zero, we can express constraints that do not involve all of the variables.
- We *cannot* always write strict inequality constraints, though sometimes such constraints can be enforced through changes to the objective function.

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

Minimize $U^*_1$

Subject to $3 * s^1_2 + (-5) * s^2_2 + (-2) * s^3_2 <= U^*_1$

$1 * s^1_2 + 4 * s^2_2 + 1 * s^3_2 <= U^*_1$

$6 * s^1_2 + (-3) * s^2_2 + (-5) * s^3_2 <= U^*_1$

$s^1_2 + s^2_2 + s^3_2 = 1$

$s^1_2 >= 0,\ s^2_2 >= 0,\ s^3_2 >= 0$

- We can solve the dual linear program to obtain a Nash equilibrium strategy for player 1.

Maximize $U^*_1$

Subject to $\Sigma_{j \text{ in } A1}\, u_1(a^j_1, a^k_2) * s^j_1 >= U^*_1$ for all k in $A_2$

$\Sigma_{j \text{ in } A1}\, s^j_1 = 1$

$s^j_1 >= 0$ for all j in $A_1$

- Duality theorem: If both a LP and its dual are feasible, then both have optimal vectors and the values of the two programs are the same.

# Why does this matter?

- Linear programs can be solved “efficiently.”
  - Ellipsoid method runs in polynomial time.
  - Simplex algorithm runs in worst-case exponential time, but runs efficiently in practice.

- Note the following equivalent formulation of the original LP:

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 + r^j_1 = U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

$r^j_1 >= 0$ for all j in $A_1$

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

# Two-player general sum games

- Random strategy:
  - *Increase cost/uncertainty to attackers*

**Adversary**

**Defender**

| | Target #1 | Target #2 |
|---|---|---|
| Target #1 | 4, -3 | -1, 1 |
| Target #2 | -5, 5 | 2, -1 |

- Minmax Theorem does not apply, so we cannot formulate as a linear program. We can instead formulate as a *Linear Complemetarity Problem (LCP)*.

Minimize ….. (No objective!)

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 + r^j_1 = U^*_1$ for all j in $A_1$

$\Sigma_{j \text{ in } A1}\, u_2(a^j_1, a^k_2) * s^k_2 + r^k_2 = U^*_2$ for all k in $A_2$

$\Sigma_{j \text{ in } A1}\, s^j_1 = 1,\ \Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^j_1, s^k_2 >= 0$ for all j in $A_1$, k in $A_2$

$r^j_1, r^k_2 >= 0$ for all j in $A_1$, k in $A_2$

$r^j_1 * s^j_1 = 0, r^j_2 * s^j_2 = 0$ for all j in $A_1$, k in $A_2$

- B. von Stengel (2002), Computing equilibria for two-person games. Chapter 45, *Handbook of Game Theory*, *Vol. 3*, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, 1723-1759.
  - http://www.maths.lse.ac.uk/personal/stengel/TEXTE/nashsurvey.pdf
- Longer earlier version (with more details on equivalent definitions of degeneracy, among other aspects): B. von Stengel (1996), Computing Equilibria for Two-Person Games. Technical Report 253, Department of Computer Science, ETH Zürich.

- Define $E = [1,\ldots,1]$, $e = 1$, $F = [1,\ldots,1]$, $f = 1$
- Given a fixed y in Y, a best response of player 1 to y is a vector x in X that maximizes the expression $x^T(Ay)$. That is, x is a solution to the LP:

  Maximize $x^T(Ay)$

  Subject to $Ex = e$, $x >= 0$

- The dual of this LP with variables u:

  Minimize $e^Tu$

  Subject to $E^Tu >= Ay$

- So a minmax strategy y of player 2 (minimizing the maximum amount she has to pay) is a solution to the LP

  Minimize $e^T u$

  Subject to $Fy = f$

  $E^T u - Ay >= 0$

  $y >= 0$

- Dual LP:

  Maximize $f^T v$

  Subject to $Ex = e$

  $F^T v - B^T x <= 0$

  $x >= 0$

- Theorem: The game (A,B) has the Nash equilibrium (x,y) if and only if for suitable u,v

$$Ex = e$$

$$Fy = f$$

$$E^T u - Ay >= 0$$

$$F^T v - B^T x >= 0$$

$$x, y >= 0$$

- This is called a *linear complementarity program*.
- Best algorithm is Lemke Howson Algorithm.
  - Does NOT run in polynomial time. Worst-case exponential.
- Computing a Nash equilibrium in these games is PPAD-complete, unlike for two-player zero-sum games where it can be done in polynomial time.

- Assume disjoint strategy sets M and N for both players. Any mixed strategy x in X and y in Y is labeled with certain elements if M union N. These labels denote the unplayed pure strategies of the player and the pure best responses of his or her opponent. For i in M and j in N, let
  - $X(i) = \{x \text{ in } X |\ x_i = 0\}$,
  - $X(j) = \{x \text{ in } X |\ b_j x >= b_k x \text{ for all } k \text{ in } N\}$
  - $Y(i) = \{y \text{ in } Y \mid a_i y >= a_k y \text{ for all } k \text{ in } M\}$
  - $Y(j) = \{y \text{ in } Y \mid y_j = 0\}$
- Then x has label k if x in X(k) (i.e., x is a best response to strategy k for player 2), and y has label k if y in Y(k), for k in M Union N.

- Clearly, the best-response regions X(j) for j in N are polytopes whose union is X. Similarly, Y is the union of the sets Y(i) for i in M. Then a Nash equilibrium is a *completely labeled* pair (x,y) since then by Theorem 2.1, any pure strategy k of a player is either a best response or played with probability zero, so it appears as a label of x or y.

- Theorem: A mixed strategy pair (x,y) in X x Y is a Nash equilibrium of (A,B) if and only if for all k in M Union N either x in X(k) or y in Y(k) (or both).

- For the following game, the labels of X and Y are:

| | **L** | **R** |
|---|---|---|
| T | 0, 1 | 6, 0 |
| M | 2, 0 | 5, 2 |
| B | 3, 4 | 3, 3 |

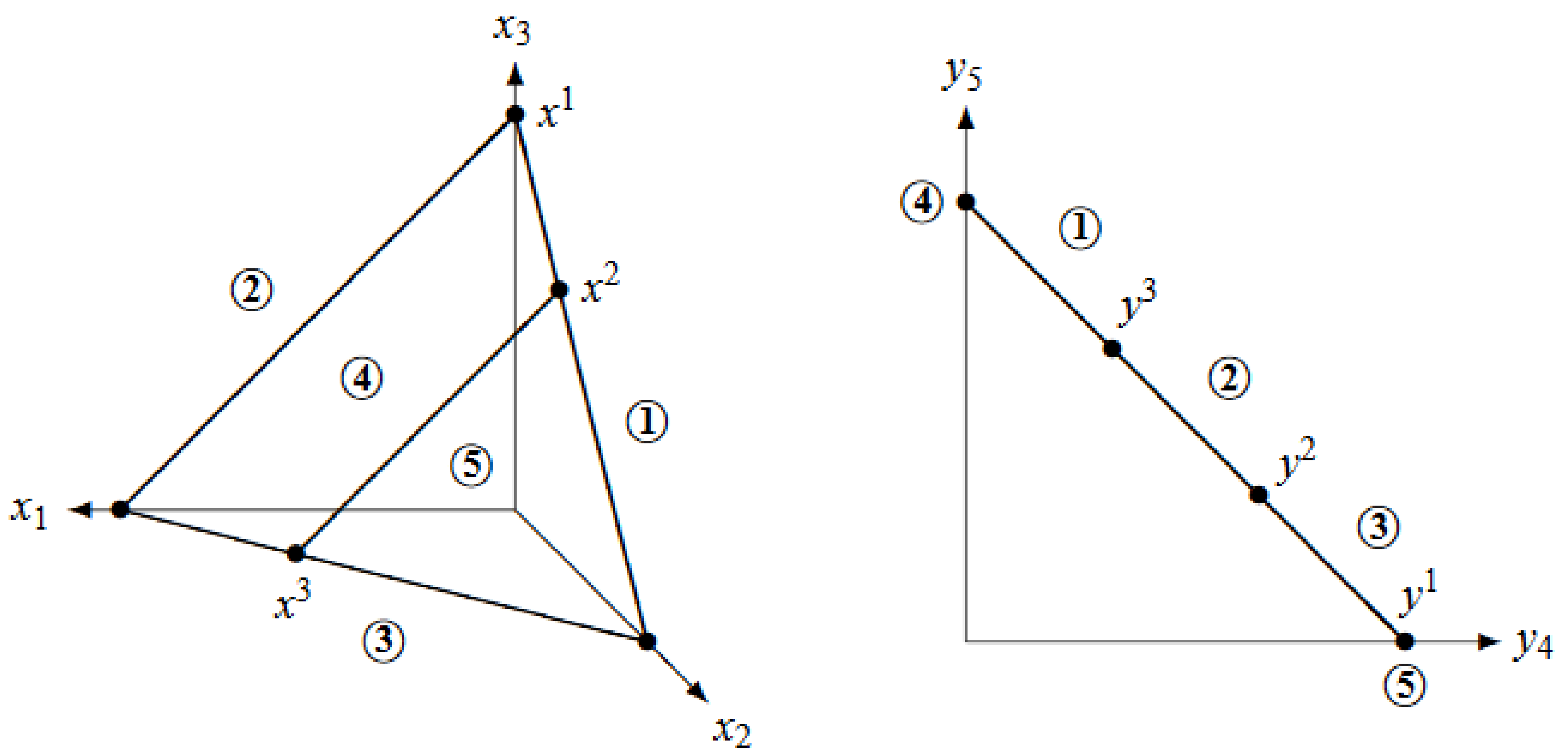



Figure 2.2. Mixed strategy sets $X$ and $Y$ of the players for the bimatrix game $(A,B)$ in (2.15). The labels $1,2,3$, drawn as circled numbers, are the pure strategies of player 1 and marked in $X$ where they have probability zero, in $Y$ where they are best responses. The pure strategies of player 2 are similar labels $4,5$. The dots mark points $x$ and $y$ with a maximum number of labels.

- The equilibria are:
  - $(x_1,y_1) = ((0,0,1),(1,0))$, where $x_1$ has the labels 1, 2, 4 (and $y_1$ has the remaining labels 3 and 5),
  - $(x_2,y_2) = ((0,1/3,2/3),(2/3,1/3))$, with labels 1, 4, 5 for $x_2$
  - $(x_3,y_3) = ((2/3,1/3,0),(1/3,2/3))$, with labels 3, 4, 5 for $x_3$

- This “inspection” is effective at finding equilibria of games of size up to 3x3. It works by inspecting any point x for P1 with m labels and checking if there is a point y having the remaining n labels. A game is “nondegenerate” if any x has at most m labels and every y has at most n labels.
- “Most” games are nondegenerate, since having an additional label imposes an additional equation that will usually reduce the dimension of the set of points having these labels by one. Since the complete set X has dimension m-1, we expect no points to have more than m labels. This will fail only in exceptional cirtcumstances if there is a special relationship between the elements of A and B.

# n-player general-sum games

- For n-player games with n >= 3, the problem of computing an NE can no longer be expressed as an LCP. While it can be expressed as a *nonlinear complementarity problem*, such problems are often hopelessly impractical to solve exactly.
- Can solve sequence of LCPs (generalization of Newton's method).
  - Not globally convergent
- Formulate as constrained optimization (minimization of a function), but also not globally convergent (e.g., hill climbing, simulated annealing can get stuck in local optimum)
- Simplicial subdivision algorithm (Scarf)
  - Divide space into small regions and search separately over the regions.
- Homotopy method (Govindan and Wilson)

# Coming up

- Algorithms for computing solution concepts in strategic-form and extensive-form games, Gambit and Game Theory Explorer software packages.
- Go through more specific examples of solving linear programs and linear complementarity programs.
- Solving games with >= 3 players.
- Computing maxmin and minmax for two-player general-sum games.
- Identifying dominated strategies
  - Domination by pure strategies, domination by mixed strategies, iterated dominance

# Assignment

- Reading for next class: Game Theory Explorer document and slides, http://www.cs.cmu.edu/~sganzfri/SeniorThesis.pdf