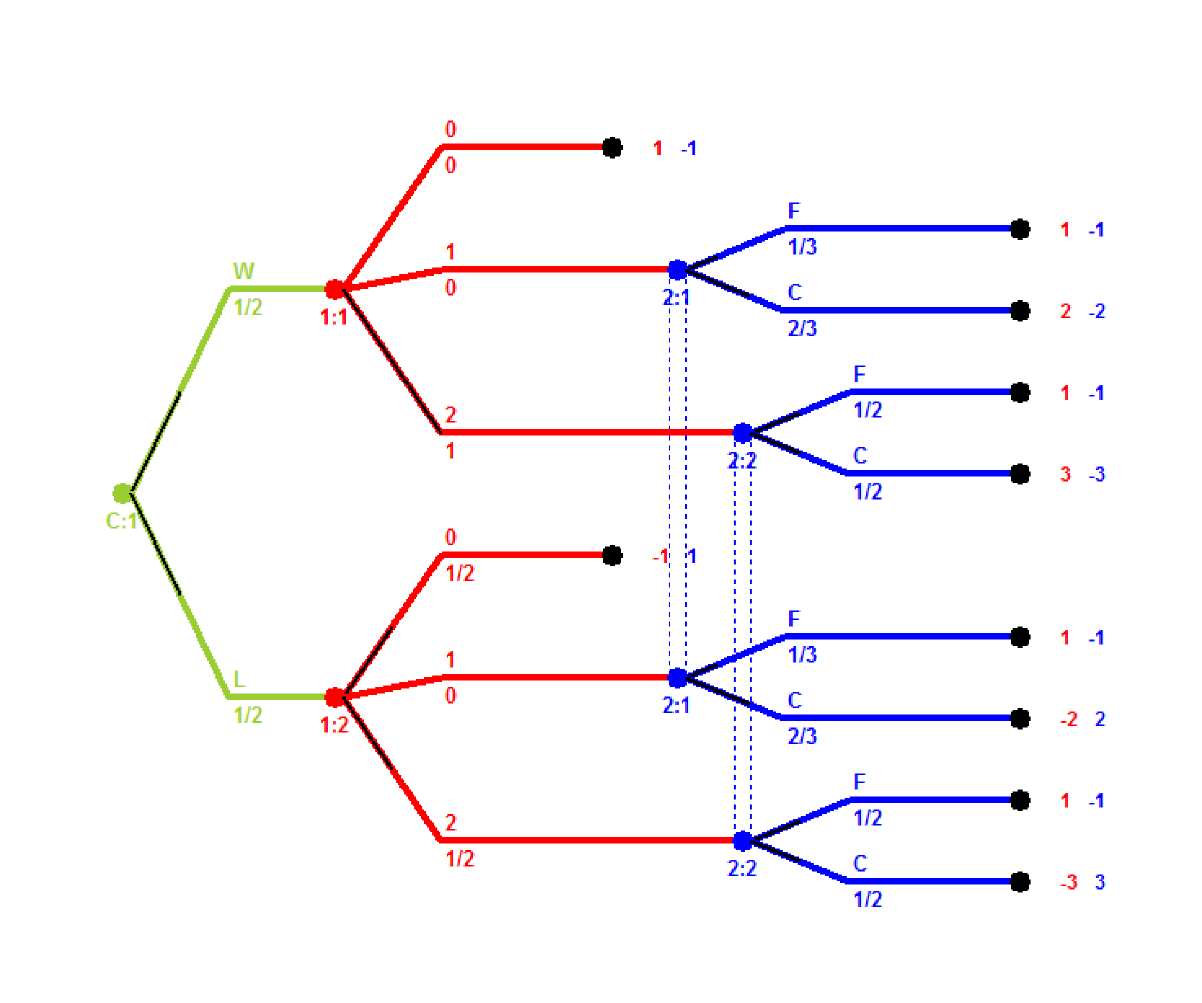
CAP 5993/CAP 4993
Game Theory

Instructor: Sam Ganzfried

sganzfri@cis.fiu.edu

Announcements

- HW2 out today – due 2/21


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

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				

Gambit functionality and approaches

- http://gambit.sourceforge.net/gambit13/contents.html

Chicken

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

Fig. 2: Chicken with numerical payoffs

Maxmin security

	L	R
T	2, 1	2, -20
M	3, 0	-10, 1
B	-100, 2	3, 3

Maxmin and minmax strategies for two-player general-sum games

- Let G be an arbitrary two-player game

  $G = (\{1,2\}, A_1 \times A_2, (u_1, u_2))$.

- Define the zero-sum game in which P1's utility is unchanged and P2's utility is the negative of P1's.

  $G' = (\{1,2\}, A_1 \times A_2, (u_1, -u_1))$.

- By Minmax Theorem every strategy for player 1 which is part of a Nash equilibrium strategy profile for G' is a maxmin strategy for player 1 in G'.
  - P1's maxmin strategy is independent of P2's utility function.
  - So P1's maxmin strategy is the same in G and G'.
- Same idea to compute minmax strategy for P2.

Identifying dominated strategies

forall *pure strategies* $a_i \in A_i$ *for player* i *where* $a_i \neq s_i$ **do**
- $dom \leftarrow true$
- **forall** *pure-strategy profiles* $a_{-i} \in A_{-i}$ *for the players other than* i **do**
  - **if** $u_i(s_i, a_{-i}) \geq u_i(a_i, a_{-i})$ **then**
    - $dom \leftarrow false$
    - break
- **if** $dom = true$ **then**
  - **return** $true$

return $false$

Figure 4.7: Algorithm for determining whether s_i is strictly dominated by any pure strategy

- Works because we do not need to check every *mixed*-strategy profile of the other players.
- If $u_i(s_i, a_{-i}) < u_i(a_i, a_{-i})$, for all a_{-i} then there cannot exist any mixed-strategy profile s_{-i} for which $u_i(s_i, s_{-i}) >= u_i(a_i, s_{-i})$, because of linearity of expectation.
- Can the algorithm be modified for weak dominance?

Domination by a mixed strategy

	L	C	R
U	$3, 1$	$0, 1$	$0, 0$
M	$1, 1$	$1, 1$	$5, 0$
D	$0, 1$	$4, 1$	$0, 0$

Figure 3.15: A game with dominated strategies.

	L	C
U	$3, 1$	$0, 1$
M	$1, 1$	$1, 1$
D	$0, 1$	$4, 1$

Figure 3.16: The game from Figure 3.15 after removing the dominated strategy R.

- Strict domination

  Maximize ….

  Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) > u_i(s_i, a_{-i})$ for all a_{-i} in A_{-i}

  $\Sigma_{j \text{ in } Ai}\, p_j = 1$

  $p_j >= 0$ for all j in A_i

- Valid?

Domination by a mixed strategy

- Strict domination

Minimize $\Sigma_{j \text{ in } Ai}\, p_j$

Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) >= u_i(s_i, a_{-i})$ for all a_{-i} in A_{-i}

$p_j >= 0$ for all j in A_i

- Weak domination

Maximize $\Sigma_{a\text{-}i \text{ in } A\text{-}i} [(\Sigma_{j \text{ in } Ai}\, p_j * u_i(a_j, a_{-i})) - u_i(s_i, a_{-i})$

Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) >= u_i(s_i, a_{-i})$ for all a_{-i} in A_{-i}

$p_j >= 0$ for all j in A_i

$\Sigma_{j \text{ in } Ai}\, p_j = 1$

- It requires only polynomial time to iteratively remove dominated strategies until the game has been maximally reduced (i.e., no strategy is dominated for any player). A single step of this process consists of checking whether every pure strategy of every player is dominated by any other mixed strategy which requires us to solve at worst $\Sigma_{i \text{ in } N} |A_i|$ linear programs. Each step removes one pure strategy for each player, so there can be at most $\Sigma_{i \text{ in } N} (|A_i| - 1)$ steps.

1. **(Strategy elimination)** Does there exist some elimination path under which the strategy s_i is eliminated?
2. **(Reduction identity)** Given action subsets A'_i subset of A_i, for each player i, does there exist a maximally reduced game where each player i has the actions A'_i?
3. **(Reduction Size)** Given constants k_i for each player i, does there exist a maximally reduced game where each player i has exactly k_i actions?

- Theorem: For iterated strict dominance, the strategy elimination, reduction identity, uniqueness and reduction size problems are in P. For iterated weak dominance, these problems are NP-complete.

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

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

$s^k_2 >= 0$ 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

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

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

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

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

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

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

Maximize U^*_1

Subject to $a * s^1_2 + b * s^2_2 + c * s^3_2 <= U^*_1$

$d * s^1_2 + e * s^2_2 + f * s^3_2 <= U^*_1$

$g * s^1_2 + h * s^2_2 + j * s^3_2 <= U^*_1$

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

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

- 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,…,1], e = 1, F = [1,…,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
  - $X(i) = \{x \text{ in } X \mid x_i = 0\}$,
  - $X(j) = \{x \text{ in } X \mid b_j x >= b_k x \text{ for all } k \text{ in } N\}$
  - $Y(i) = \{y \text{ in } Y \mid a_i y >= a_k y \text{ for all } k \text{ in } M\}$
  - $Y(j) = \{y \text{ in } Y \mid y_j = 0\}$
- 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.

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

$$Ex = e$$

$$Fy = f$$

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

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

$$x, y >= 0$$

- *Complementarity condition:* requires that whenever an action is played by a given player with positive probability (i.e., whenever an action is in the support of a given player’s mixed strategy), then the corresponding slack variable must be zero. Under this requirement, each slack variable can be viewed as the player’s incentive to deviate from the corresponding action. Thus, the complementarity condition captures the fact that, in equilibrium, all strategies that are played with positive probability must yield the same expected payoff, while all strategies that lead to lower expected payoffs are not played.

- 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

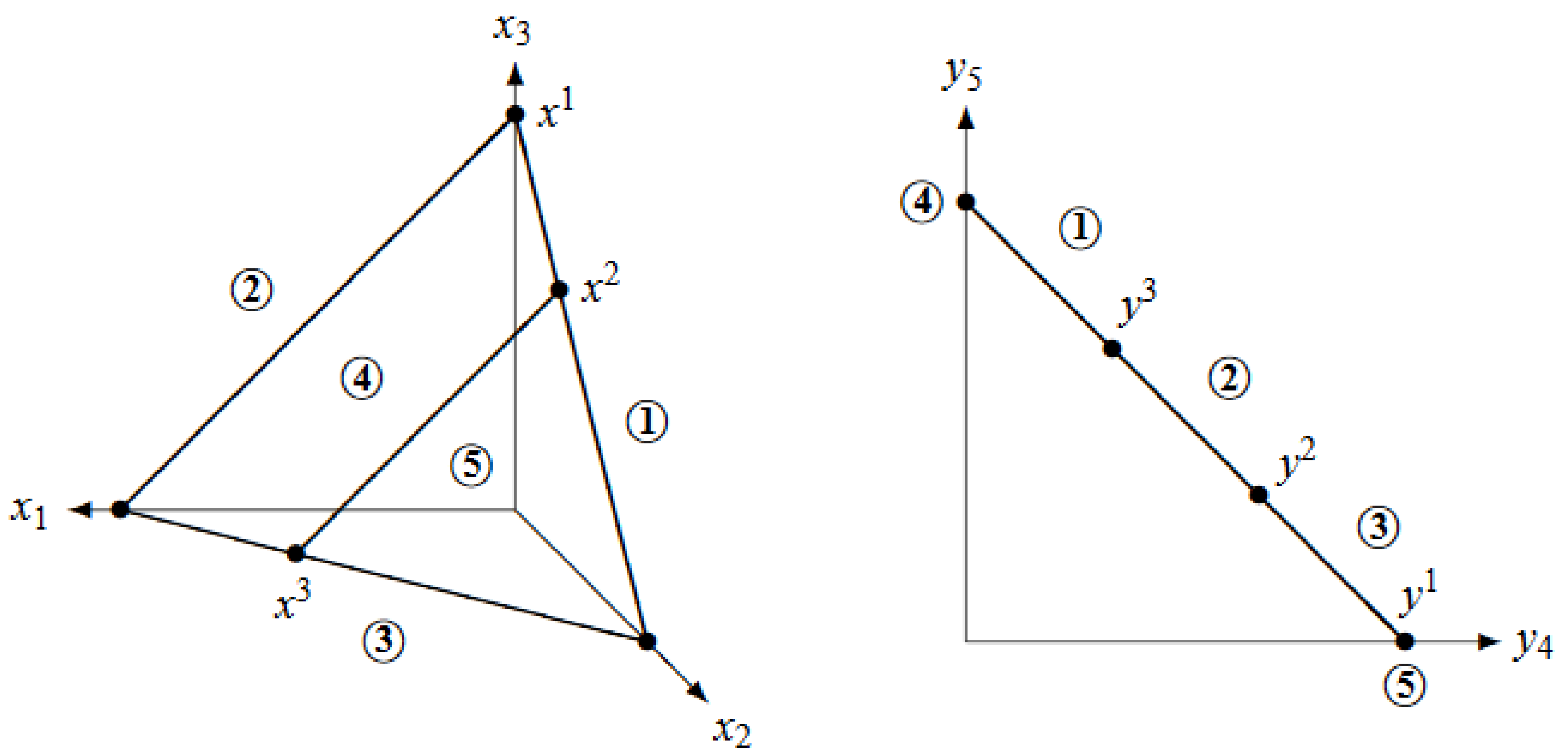


Figure 2.2. Mixed strategy sets X and Y of the players for the bimatrix game (A,B) in (2.15). The labels $1,2,3$, drawn as circled numbers, are the pure strategies of player 1 and marked in X where they have probability zero, in Y where they are best responses. The pure strategies of player 2 are similar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

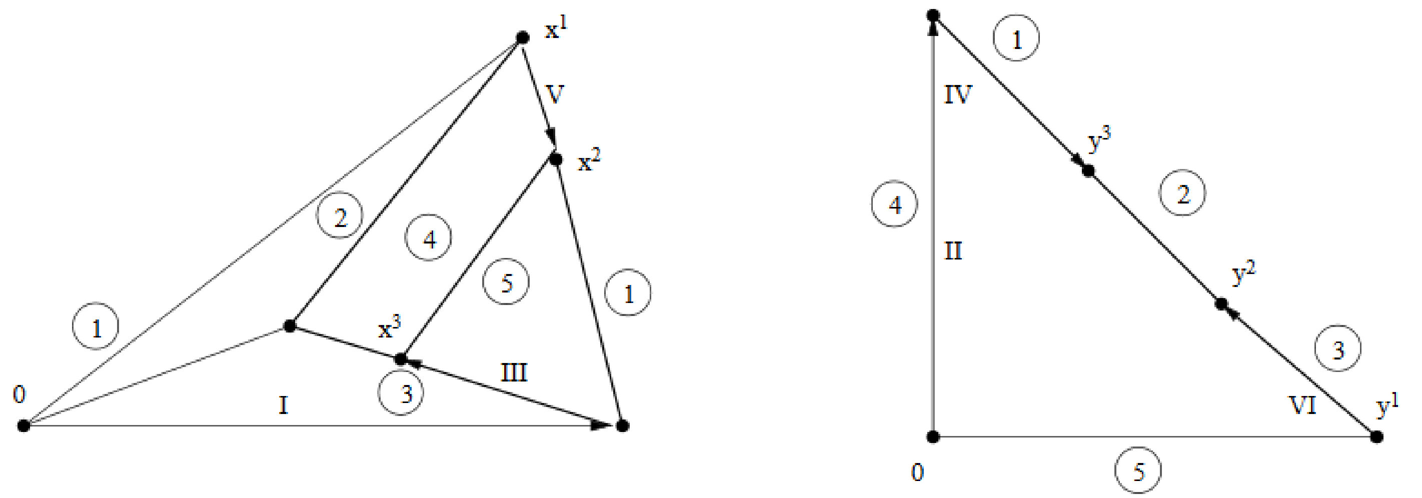


FIGURE 2. The graphs G_1 and G_2 for the game (5.1).

Figure 2 demonstrates the algorithm on the game (5.1) defined above with $k = 2$. The algorithm starts with $x = (0,0,0)$ and $y = (0,0)$. Step I: since x contains label 2, y will remain the same and we must switch x in G_1. It is clear that we must change x to $(0,1,0)$, which causes label 5 to be duplicated. Step II: dropping label 5 in G_2 changes y to $(0,1)$, which picks up label 1. Step III: dropping label 1 in G_1 changes x to $(\frac{2}{3}, \frac{1}{3}, 0)$, which duplicates label 4. Step IV: dropping label 4 in G_2 changes y to $(\frac{1}{3}, \frac{2}{3})$, which has the missing label 2. So the algorithm terminates at the Nash equilibrium $((\frac{2}{3}, \frac{1}{3}, 0), (\frac{1}{3}, \frac{2}{3}))$. Similarly, steps V and VI in the figure join the equilibria (x^1, y^1) and (x^2, y^2) on a 2-almost completely labeled path.

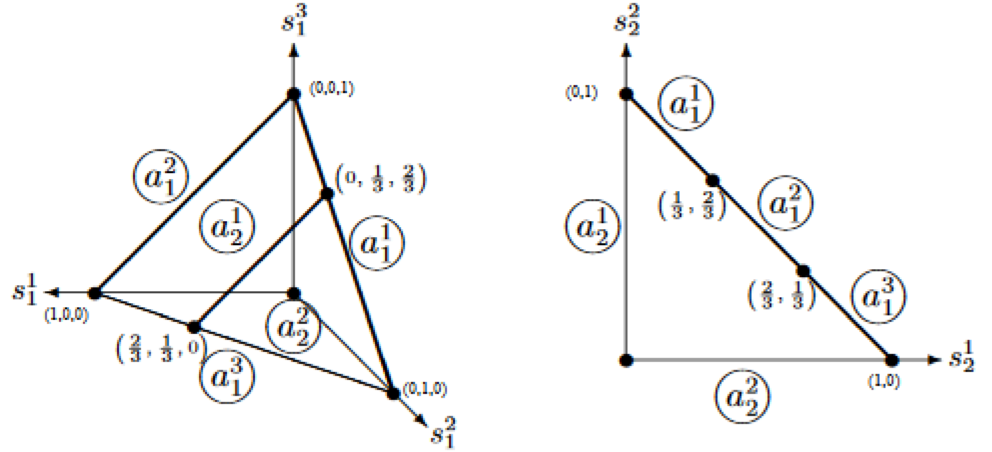


Figure 4.3: Labeled strategy spaces for player 1 (left) and player 2 (right) in the game from Figure 4.1.

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)
  - n-player extension of Lemke-Howson Algorithm

Next class

- Go over HW1.
- Lemke Howson algorithm details.
- Algorithms for extensive-form games.
- Game Theory Explorer software package.

Assignment

- HW2 out today, due 2/21.
- Reading for next class: Gambit software http://gambit.sourceforge.net/gambit13/contents.html