Introduction to Game Theory

8. Stochastic Games

Dana Nau

University of Maryland

Stochastic Games

- A stochastic game is a collection of normal-form games that the agents play repeatedly
- The particular game played at any time depends probabilistically on
  - the previous game played
  - the actions of the agents in that game
- Like a probabilistic FSA in which
  - the states are the games
  - the transition labels are joint action-payoff pairs

Markov Games

- A **stochastic** (or **Markov**) game includes the following:
  - a finite set Q of states (games),
  - a set $N = \{1, \ldots, n\}$ of agents,
  - For each agent i, a finite set A_i of possible actions
  - A **transition probability function** $P : Q \times A_1 \times \cdots \times A_n \times Q \to [0, 1]$

    $P(q, a_1, \ldots, a_n, q') =$ probability of transitioning to state q' if the action profile $(a_1, \ldots, a_n)$ is used in state q
  - For each agent i, a real-valued **payoff function** $r_i : Q \times A_1 \times \cdots \times A_n \to \Re$
- This definition makes the inessential but simplifying assumption that each agent's strategy space is the same in all games
  - So the games differ only in their payoff functions

Histories and Rewards

- Before, a history was just a sequence of actions
  - But now we have action profiles rather than individual actions, and each profile has several possible outcomes
- Thus a history is a sequence $h_t = (q^0, a^0, q^1, a^1, \ldots, a^{t-1}, q^t)$, where t is the number of stages
- As before, the two most common methods to aggregate payoffs into an overall payoff are **average reward** and **future discounted reward**

- Stochastic games generalize both Markov decision processes (MDPs) and repeated games
  - An MDP is a stochastic game with only 1 player
  - A repeated game is a stochastic game with only 1 state
    - Iterated Prisoner's Dilemma, Roshambo, Iterated Battle of the Sexes, …

Strategies

- For agent i, a **deterministic** strategy specifies a choice of action for i at every stage of every possible history
- A mixed strategy is a probability distribution over deterministic strategies
- Several restricted classes of strategies:
  - As in extensive-form games, a **behavioral strategy** is a mixed strategy in which the mixing take place at each history independently
  - A **Markov strategy** is a behavioral strategy such that for each time t, the distribution over actions depends only on the current state
    - But the distribution may be different at time t than at time $t' \neq t$
  - A **stationary strategy** is a Markov strategy in which the distribution over actions depends only on the current state (not on the time t)

Equilibria

- First consider the (easier) discounted-reward case
- A strategy profile is a **Markov-perfect equilibrium** (MPE) if
  - it consists of only Markov strategies
  - it is a Nash equilibrium regardless of the starting state
- **Theorem**. Every n-player, general-sum, discounted-reward stochastic game has a MPE
- The role of Markov-perfect equilibria is similar to role of subgame-perfect equilibria in perfect-information games

Equilibria

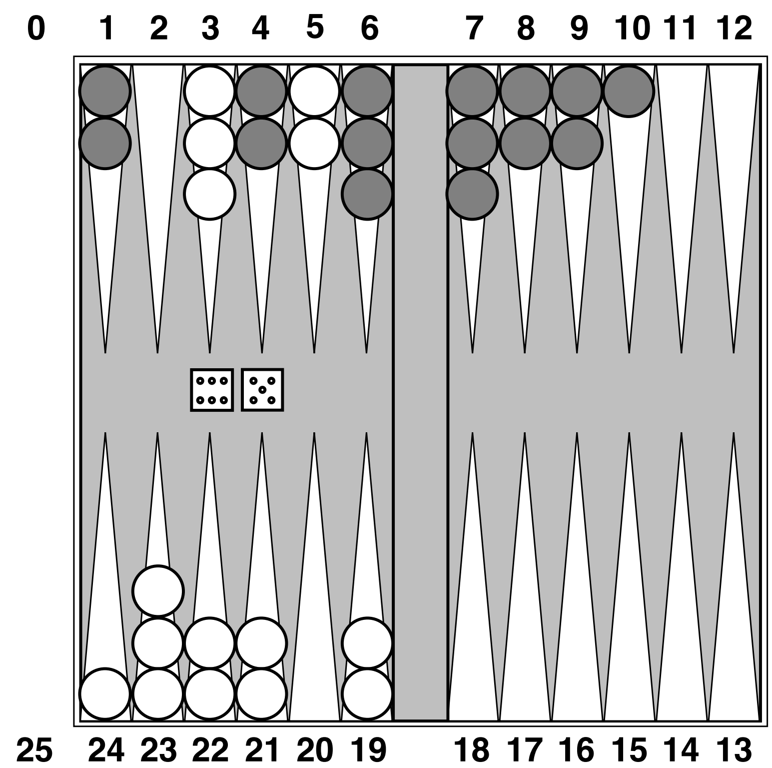
- Now consider the average-reward case

- A stochastic game is **irreducible** if every game can be reached with positive probability regardless of the strategy adopted
- **Theorem**. Every 2-player, general-sum, average reward, irreducible stochastic game has a Nash equilibrium

- A payoff profile is **feasible** if it is a convex combination of the outcomes in a game, where the coefficients are rational numbers
- There's a folk theorem similar to the one for repeated games:
  - If (p_1,p_2) is a feasible pair of payoffs such that each p_i is at least as big as agent i's minimax value, then (p_1,p_2) can be achieved in equilibrium through the use of enforcement

Two-Player Zero-Sum Stochastic Games

- For two-player zero-sum stochastic games
  - The folk theorem still applies, but it becomes vacuous
  - The situation is similar to what happened in repeated games
    - The only feasible pair of payoffs is the minimax payoffs

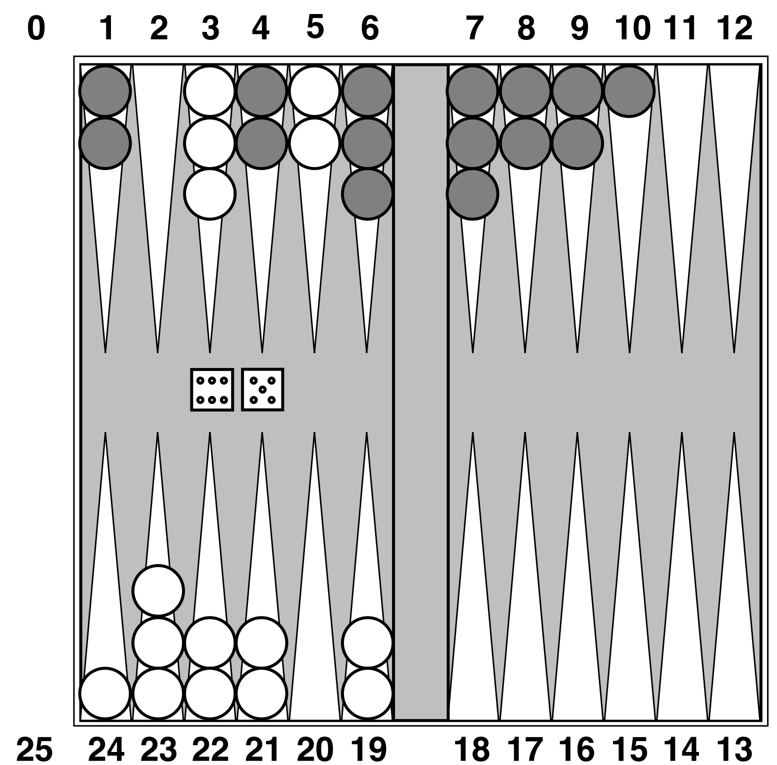


- One example of a two-player zero-sum stochastic game is Backgammon
- Two agents who take turns
  - Before his/her move, an agent must roll the dice
  - The set of available moves depends on the results of the dice roll

Backgammon

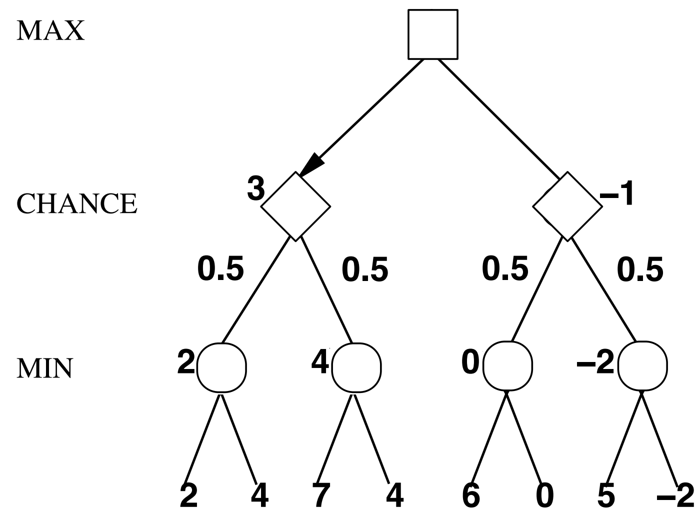
- Mapping Backgammon into a Markov game is straightforward, but slightly awkward
- Basic idea is to give each move a stochastic outcome, by combining it with the dice roll that comes *after* it
- Every state is a pair:
(current board, current dice configuration)
  - Initial set of states = {initial board} × {all possible results of agent 1's first dice roll}
  - Set of possible states after agent 1's move =

    {the board produced by agent 1's move}
    × {all possible results of agent 2's dice roll}
  - Vice versa for agent 2's move
- We can extend the minimax algorithm to deal with this
  - But it's easier if we don't try to combine the moves and the dice rolls
  - Just keep them separate



The Expectiminimax Algorithm

- Two-player zero-sum game in which
  - Each agent's move has a deterministic outcome
  - In addition to the two agents' moves, there are chance moves
- The algorithm gives optimal play (highest expected utility)



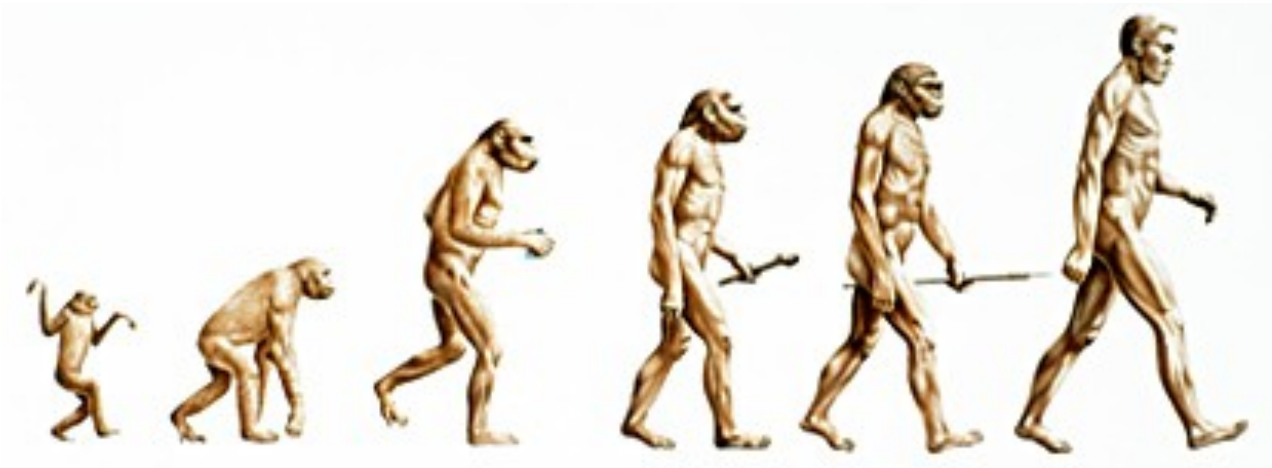
function EXPECTIMINIMAX(s) **returns** an expected utility
  if s is a terminal state **then return** Max's payoff at s
  if s is a "chance" node **then**
    return $\sum_{s'} P(s'|s)$EXPECTIMINIMAX(s')
  else if it is Max's move at s **then**
    return max{EXPECTIMINIMAX(result(a, s)) : a is applicable to s}
  else return min{EXPECTIMINIMAX(result(a, s)) : a is applicable to s}

In practice

- Dice rolls increase branching factor
  - 21 possible rolls with 2 dice
  - Given the dice roll, ≈ 20 legal moves on average
    - For some dice roles, can be much higher
    - depth 4 = $20\times(21\times20)^3 \approx 1.2\times10^9$
  - As depth increases, probability of reaching a given node shrinks
    - ⇒ value of lookahead is diminished
- α-β pruning is much less effective
- TDGammon uses depth-2 search + very good evaluation function
  - ≈ world-champion level
  - The evaluation function was created automatically using a machine-learning technique called *Temporal Difference* learning
    - hence the *TD* in TDGammon

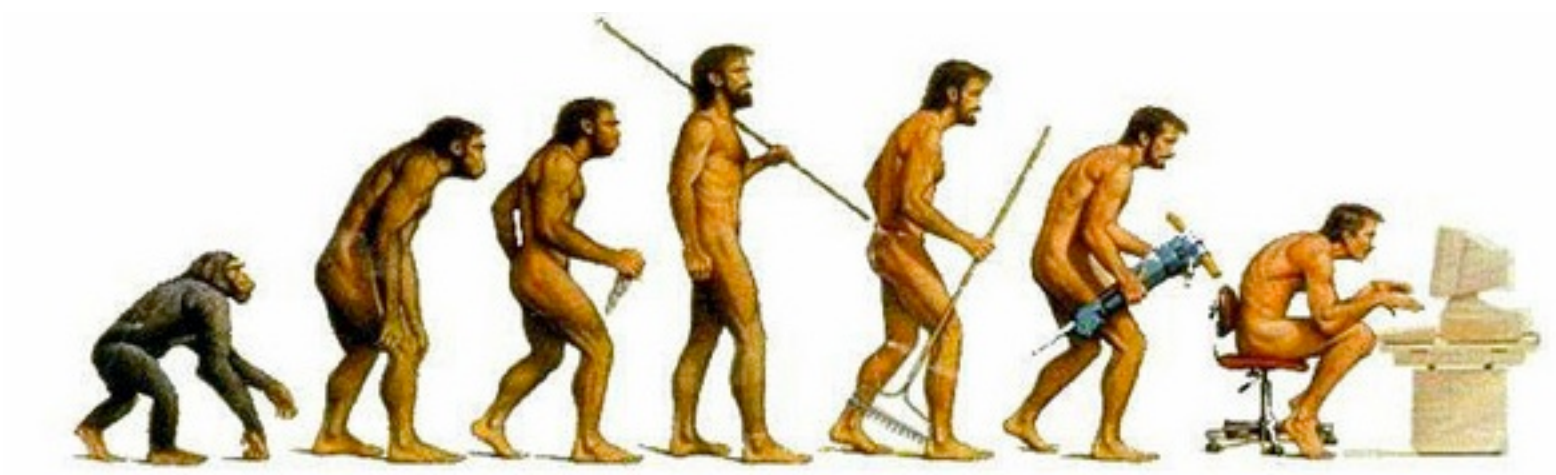
Evolutionary Simulations

- An evolutionary simulation is a stochastic game whose structure is intended to model certain aspects of evolutionary environments
  - At each **stage** (or **generation**) there is a large set (e.g., hundreds) of agents
- Different agents may use different strategies
  - A strategy *s* is represented by the set of all agents that use strategy *s*
  - Over time, the number of agents using *s* may grow or shrink depending on how well *s* performs
- *s*'s **reproductive success** is the fraction of agents using *s* at the end of the simulation,
  - i.e., (number of agents using *s*)/(total number of agents)



Reproduction Dynamics

- At each stage, some set of agents (maybe all of them, maybe just a few) is selected to perform actions at that stage
  - Each agent receives a *fitness* value: a stochastic function of the action profile
- Depending on the agents' fitness values, some of them may be removed and replaced with agents that use other strategies
  - Typically an agent with higher fitness is likely to see its numbers grow
  - The details depend on the **reproduction dynamics**
    - The mechanism for selecting which agents will be removed, which agents will reproduce, and how many progeny they'll have



Replicator Dynamics

- **Replicator dynamics** works as follows:
  - $p_i^{new} = p_i^{curr}\, r_i / R$,

  where

  - p_i^{new} is the proportion of agents of type i in the next stage
  - p_i^{curr} is the proportion of agents of type i in the current stage
  - r_i = average payoff received by agents of type i in the current stage
  - R_i = average payoff received by all agents in the current stage

- Under the replicator dynamics, an agent's numbers grow (or shrink) proportionately to how much better it does than the average
- Probably the most popular reproduction dynamics
  - e.g., does well at reflecting growth of animal populations

Replicator Dynamics

- **Imitation dynamics** (or **tournament selection**) works as follows:
  - Randomly choose 2 agents from the population, and compare their payoffs
    - The one with the higher payoff reproduces into the next generation
  - Do this *n* times, where *n* is the total population size

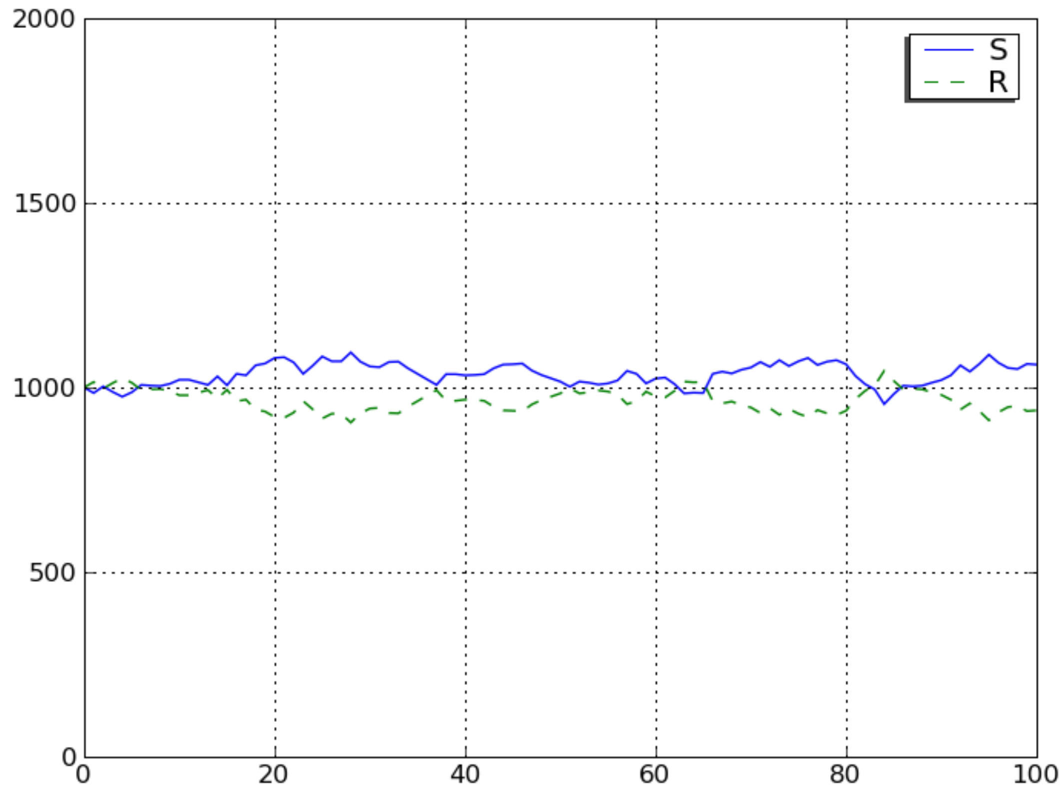
- Under the imitation dynamics, an agent's numbers grow if it does better than the average
  - But unlike replicator dynamics, the amount of growth doesn't depend on **how much** better than the average
- Thought to be a good model of the spread of behaviors in a culture

Example: A Simple Lottery Game

- A repeated lottery game
- At each stage, agents make choices between two lotteries
  - "Safe" lottery: guaranteed reward of 4
  - "Risky" lottery: [0, 0.5; 8, 0.5],
    - i.e., probability ½ of 0, and probability ½ of 8
- Let's just look at stationary strategies
- Two pure strategies:
  - *S*: always choose the "safe" lottery
  - *R*: always choose "risky" lottery
- Many mixed strategies, one for every p in [0,1]
  - R_p: probability p of choosing the "risky" lottery, and probability $1–p$ of choosing the "safe" lottery

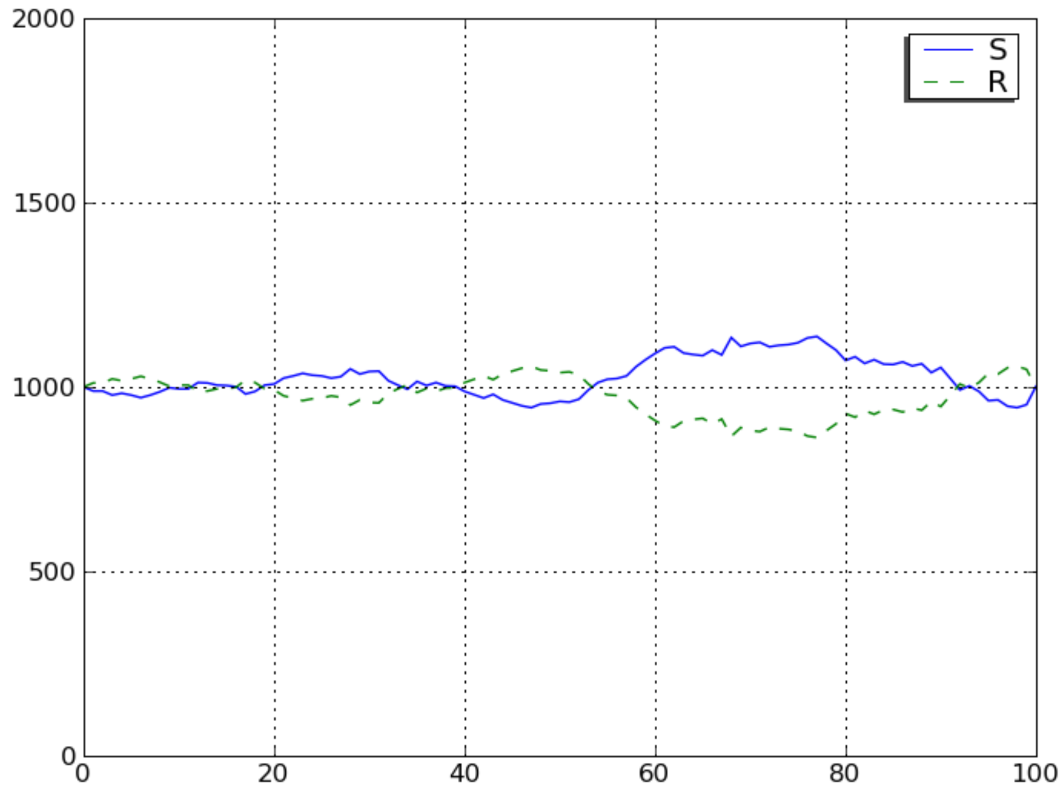
Lottery Game with Replicator Dynamics

- At each stage, each strategy's average payoff is 4
  - Thus on average, each strategy's population size should stay roughly constant
- Verified by simulation for *S* and *R*
- Would get similar behavior with any of the R_p strategies



Lottery Game with Imitation Dynamics

- Pick any two agents, and let *s* and *t* be their strategies
- Regardless of what *s* and *t* are, each agent has equal probability of getting a higher payoff than the other
  - Again, each strategy's population size should stay roughly constant
- Verified by simulation for *S* and *R*
- Would get similar behavior with any of the S_p strategies

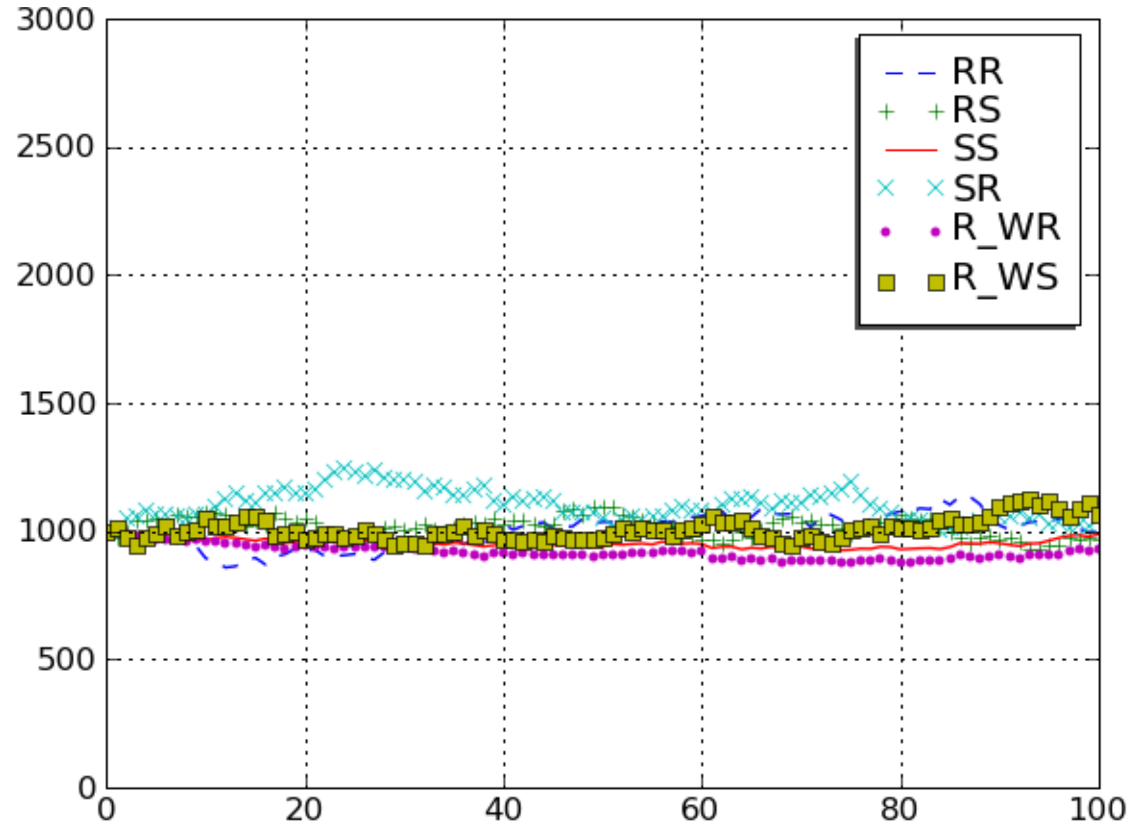


Double Lottery Game

- Now, suppose that at each stage, agents make two rounds of lottery choices
  1. Choose between the safe or risky lottery, get a reward
  2. Choose between the safe or risky lottery again, get another payoff
- This time, there are 6 stationary pure strategies
  - *SS*: choose “safe” both times
  - *RR*: choose “safe” both times
  - *SR*: choose “safe” in first round, “risky” in second round
  - *RS*: choose “risky” in first round, “safe” in second round
  - *R-WR*: choose “risky” in first round
    - If it wins (i.e., reward is 8), then choose “risky” again in second round
    - Otherwise choose “safe” in second round
  - *R-WS*: choose “risky” in first round
    - If it wins (i.e., reward is 8), then choose “safe” in second round
    - Otherwise choose “risky” in second round

Double Lottery Game, Replicator Dynamics

- At each stage, each strategy's average payoff is 8
  - Thus on average, each strategy's population size should stay roughly constant
- Verified by simulation for all 6 strategies



Double Lottery Game, Imitation Dynamics

- Pick any two agents *a* and *b*, and let choose actions
  - Reproduce the agent (hence its strategy) that wins (i.e., higher reward)
  - If they get the same reward, choose one of them at random

- We need to look at each strategy's distribution of payoffs:

R-WS			*R-WR*			*SR*		*RS*		*SS*	*RR*		
12	**8**	**0**	**16**	**8**	**4**	**12**	**4**	**12**	**4**	**8**	**16**	**8**	**0**
.5	.25	.25	.25	.25	.5	.5	.5	.5	.5	1	.25	.5	.25

- Suppose *a* uses *SS* and *b* uses *SR*
  - *P*(*SR* gets 12 and *SS* gets 8) = (0.5)(1.0) = 0.5 => *SR* wins
  - *P*(*SR* gets 4 and *SS* gets 8) = (0.5)(1.0) = 0.5 => *SS* wins
  - Thus *a* and *b* are equally likely to reproduce
- Same is true for any two of {*SS, SR, RS, RR*}

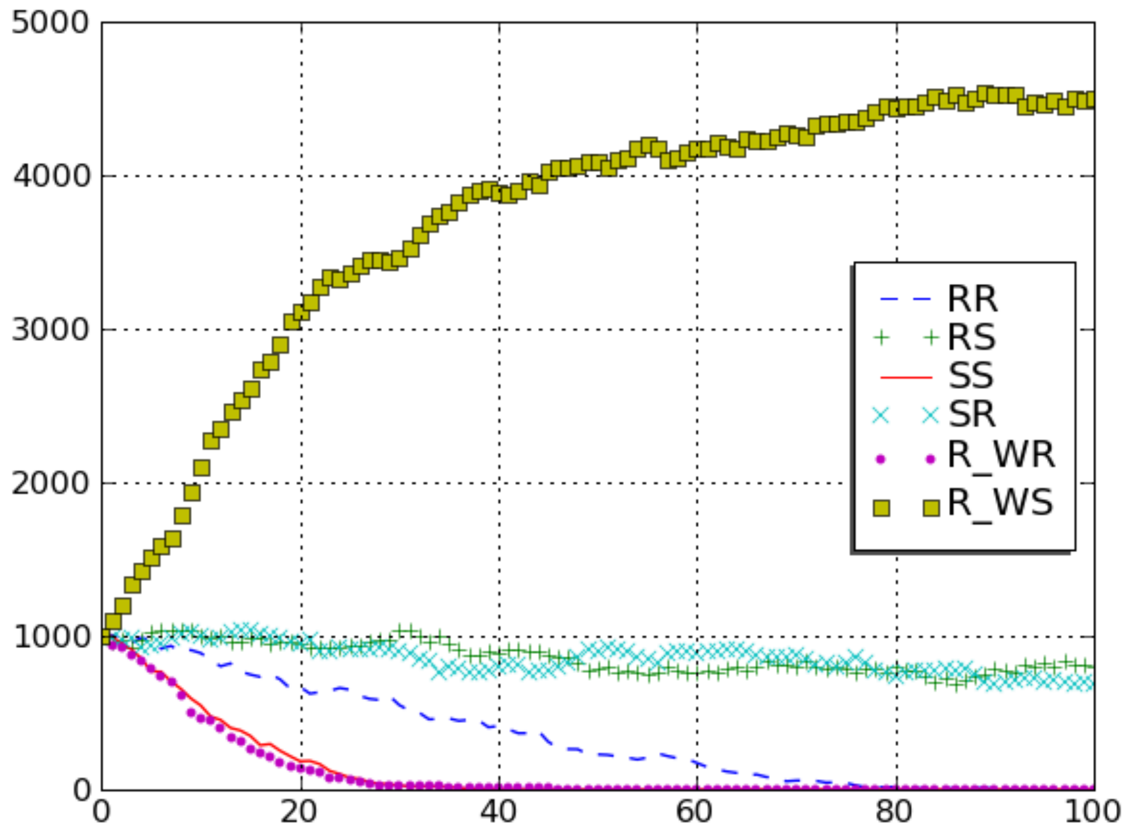
Double Lottery Game, Imitation Dynamics

R-WS			*R-WR*			*SR*		*RS*		*SS*	*RR*		
12	**8**	**0**	**16**	**8**	**4**	**12**	**4**	**12**	**4**	**8**	**16**	**8**	**0**
.5	.25	.25	.25	.25	.5	.5	.5	.5	.5	1	.25	.5	.25

- Suppose *a* uses *R-WS* and *b* uses *SS*
  - Even though they have the same *expected* reward, *R-WS* is likely to get a slightly higher reward than *SS*:
    - *P*(*R-WS* gets 12 and *SS* gets 8) = (0.5)(1.0) = 0.5 => *R-WS* wins
    - *P*(*R-WS* gets 8 and *SS* gets 8) = (0.25)(1.0) = 0.25 => tie
    - *P*(*R-WS* gets 0 and *SS* gets 8) = (0.25)(1.0) = 0.25 => *SS* wins
  - Thus *a* reproduces with probability 0.625,
    and *b* reproduces with probability 0.375
- Similarly, *a* is more likely to reproduce than *b*
  if *a* uses *R-WS* and *b* uses any of {*SS, RR, R-WR*}

Double Lottery Game, Imitation Dynamics

- If we start with equal numbers of all 6 strategies, *S-WR* will increase until *SS, RR*, and *R-WR* become extinct
  - The population should stabilize with a high proportion of *S-WR*, and low proportions of *SR* and *RS*
  - Verified by simulation:



Significance

- Recall from Session 1 that people are risk-averse
- Furthermore, there's evidence that people's risk preferences are *state-dependent*
  - Someone who's sufficiently unhappy his/her their current situation is likely to be risk-prone rather than risk-averse

- Question: why does such behavior occur?
- The evolutionary game results suggest an interesting possibility:
  - Maybe it has an evolutionary advantage over other behaviors

P. Roos and D. S. Nau. Conditionally risky behavior vs. expected value maximization in evolutionary games. In *Sixth Conference of the European Social Simulation Association (ESSA 2009), Sept. 2009.*

Summary

- Stochastic (Markov) games
  - Reward functions, equilibria
  - Expectiminimax
  - Example: Backgammon
- Evolutionary simulations
  - Replicator dynamics versus imitation dynamics
  - Example: lottery games, risk preferences