CS 91r Project: An extension to the standard Iterated Prisoner's Dilemma.

Rachel Gologorsky, under the supervision of Dr. Hilbe and Professor Parkes

April 27, 2018

1 Motivation

When two people engage in real-life repeated interactions, the payoffs they receive from each individual encounter is generally proportional to the importance of that particular encounter. For example, the daily life of a married couple is composed of many repeated interactions, where whether it is in one's long-term best interest to cooperate or defect on any individual issue is greatly impacted by the relative importance of that issue.

In addition, people often exert control over the importance of their encounters. Consider the case of colleagues who decide to start a research collaboration together. We can imagine a scenario where each time they choose to work together on a research project, they can choose between working on a straightforward, low-stakes project or on a high-stakes research project with a lot of potential. Conceivably, the possibility of higher payoffs if both researchers could agree to work on higher-stakes research might influence optimal strategies.

Thus, we created a model that expands the standard iterated Prisoner's Dilemma framework. This model allows the weight (benefit of cooperation) of individual P.D. encounters to vary between being low-stakes and high-stakes as a function of both player's preferences. In particular, we adopted the model that an interaction is only high-stakes if both players want it to be high-stakes. Other alternatives, such as unilaterally determined high-stakes encounters or stochastic weighting of encounters, are possible, and would be interesting to explore as well.

Given our model, we investigated how the presence of a higher-stakes game affects optimal strategies, as well as how it affects the critera for establishing mutual cooperation.

2 The Model

We model the choice between high-stakes and low-stakes encounters in an iterated Prisoner's Dilemma as a choice between two possible payoff matrices for each round: a "high-stakes" Game 1 matrix $\begin{bmatrix} b_1-c & -c \\ b_1 & 0 \end{bmatrix}$ and a "low-stakes" Game 2 matrix $\begin{bmatrix} b_2-c & c \\ b_2 & 0 \end{bmatrix}$. We have chosen to model a player's strategy as a vector

$$[p_{1cc}, p_{1cd}, p_{1dc}, p_{1dd}, \quad p_{2cc}, p_{2cd}, p_{2dc}, p_{2dd}, \quad x_{cc}, x_{cd}, x_{dc}, x_{dd}] \in [0,1]^{12},$$

where

- $p_{1cc}, p_{1cd}, p_{1dc}, p_{1dd}$ describe the probability of cooperating, given that we are in Game 1, for the previous round results of CC, CD, DC, and DD, respectively.
- $p_{2cc}, p_{2cd}, p_{2dc}, p_{2dd}$ describe the probability of cooperating, given we are in the Game 2, for the previous round results of CC, CD, DC, and DD, respectively,
- $x_{cc}, x_{cd}, x_{dc}, x_{dd}$ describe the probability of preferring Game 1 given the previous round results of CC, CD, DC, and DD, respectively.

Thus, strategies can react differently to a previous round outcome of $CC, .., DD$ if they are in a low-stakes situation or in a high-stakes situation.

It would be interesting to expand the strategy space to also include whether the prior round outcome was in Game 1 or Game 2. Time and space constraints, however, make this an avenue of future research.

Indeed, as our first foray into this new framework, we focused our initial examination on the simplest strategy space: (almost) pure memory-1 strategies, where the probabilities are all $\epsilon \approx 0$ and $\epsilon \approx 1$ (we introduce ϵ for the technical reason of making the transition matrix irreducible).

We investigated strategy dynamics by running a birth-death evolutionary simulation process with selection pressure for strategies with high payoff values, exploring which mutually cooperative strategies can be successful, and deriving mathematically the conditions under which those fully cooperative strategies can become subgame perfect Nash equilibria.

We found one subgame perfect Nash equilibria; determining whether there are other strategies that are subgame perfect Nash equilibria is future avenue of research.

3 Method Overview

3.1 Evolution Simulation

Start with a population of $N = 100$ individuals, who all initially play the $ALL - D$ strategy. For T timesteps, repeat:

- With probability μ, someone randomly adopts a new strategy in the strategy space.
- With probability $1 - \mu$, two individuals - a learner and a role model - are randomly sampled from the current population. With imitation probability proportional to the selection pressure β and the relative difference in average payoff, the learner adopts the role model's strategy. Specifically, the imitation probability is given by the sigmoid function $\frac{1}{1+e^x}$, where $x = -\beta * (\pi_{rolemodel} - \pi_{learner})$ and π_i refers to the average payoff of strategy i in the current population.

At the end of each simulation, we aggregate over the T timesteps and ouput:

- the most frequently used strategies over the T timesteps (and their frequency of use),
- the average individual payoff,
- the average probability of mutual cooperation (probability of being in a CC state) given the strategy dynamics in the population,
- the average probability of being in a Game 1 state given the strategy dynamics in the population,

On the level of the strategy descriptions (not interactions), we also keep track of:

- $x_{cc} + ..x_{dd}$, the average probability of the strategy preferring Game 1 (not considering strategy interactions),
- $p_{1cc} + ... + p_{1dd}$, $p_{2cc}, +... + p_{2dd}$, and $p_{1cc} + ... + p_{2dd}$, which are the strategues' average probability of cooperating in Game 1, Game 2, and overall, respectively (not considering strategy interactions).

3.2 Calculation Details

Given a pair of mixed memory-1 strategies, strategy i vs. j, of the form

$$[p_{1cc}, p_{1cd}, p_{1dc}, p_{1dd}, \quad p_{2cc}, p_{2cd}, p_{2dc}, p_{2dd}, \quad x_{cc}, x_{cd}, x_{dc}, x_{dd}],$$

we can construct the $8x8$ stochastic transition matrix Q describing the probability of transitioning among the states $1CC, 1CD, 1DC, 1DD, 2CC, 2CD, 2DC, 2DD$ (here $f(x, y) = xy$, so

Prob(Game 1) = Prob(Player 1 prefers Game 1 and Player 2 prefers Game 1) = Prob(Player 1 prefers Game 1) * Prob (Player 2 prefers Game 1)).

Q =

f[xcc, ycc] * p1cc * q1cc	f[xcc, ycc] * p1cc * (1 - q1cc)	f[xcc, ycc] * (1 - p1cc) * q1cc	f[xcc, ycc] * (1 - p1cc) * (1 - q1cc)	(1 - f[xcc, ycc]) * p2cc * q2cc	(1 - f[xcc, ycc]) * p2cc * (1 - q2cc)	(1 - f[xcc, ycc]) * (1 - p2cc) * q2cc	(1 - f[xcc, ycc]) * (1 - p2cc) * (1 - q2cc)
f[xcd, ydc] * p1cd * q1dc	f[xcd, ydc] * p1cd * (1 - q1dc)	f[xcd, ydc] * (1 - p1cd) * q1dc	f[xcd, ydc] * (1 - p1cd) * (1 - q1dc)	(1 - f[xcd, ydc]) * p2cd * q2dc	(1 - f[xcd, ydc]) * p2cd * (1 - q2dc)	(1 - f[xcd, ydc]) * (1 - p2cd) * q2dc	(1 - f[xcd, ydc]) * (1 - p2cd) * (1 - q2dc)
f[xdc, ycd] * p1dc * q1cd	f[xdc, ycd] * p1dc * (1 - q1cd)	f[xdc, ycd] * (1 - p1dc) * q1cd	f[xdc, ycd] * (1 - p1dc) * (1 - q1cd)	(1 - f[xdc, ycd]) * p2dc * q2cd	(1 - f[xdc, ycd]) * p2dc * (1 - q2cd)	(1 - f[xdc, ycd]) * (1 - p2dc) * q2cd	(1 - f[xdc, ycd]) * (1 - p2dc) * (1 - q2cd)
f[xdd, ydd] * p1dd * q1dd	f[xdd, ydd] * p1dd * (1 - q1dd)	f[xdd, ydd] * (1 - p1dd) * q1dd	f[xdd, ydd] * (1 - p1dd) * (1 - q1dd)	(1 - f[xdd, ydd]) * p2dd * q2dd	(1 - f[xdd, ydd]) * p2dd * (1 - q2dd)	(1 - f[xdd, ydd]) * (1 - p2dd) * q2dd	(1 - f[xdd, ydd]) * (1 - p2dd) * (1 - q2dd)
f[xcc, ycc] * p1cc * q1cc	f[xcc, ycc] * p1cc * (1 - q1cc)	f[xcc, ycc] * (1 - p1cc) * q1cc	f[xcc, ycc] * (1 - p1cc) * (1 - q1cc)	(1 - f[xcc, ycc]) * p2cc * q2cc	(1 - f[xcc, ycc]) * p2cc * (1 - q2cc)	(1 - f[xcc, ycc]) * (1 - p2cc) * q2cc	(1 - f[xcc, ycc]) * (1 - p2cc) * (1 - q2cc)
f[xcd, ydc] * p1cd * q1dc	f[xcd, ydc] * p1cd * (1 - q1dc)	f[xcd, ydc] * (1 - p1cd) * q1dc	f[xcd, ydc] * (1 - p1cd) * (1 - q1dc)	(1 - f[xcd, ydc]) * p2cd * q2dc	(1 - f[xcd, ydc]) * p2cd * (1 - q2dc)	(1 - f[xcd, ydc]) * (1 - p2cd) * q2dc	(1 - f[xcd, ydc]) * (1 - p2cd) * (1 - q2dc)
f[xdc, ycd] * p1dc * q1cd	f[xdc, ycd] * p1dc * (1 - q1cd)	f[xdc, ycd] * (1 - p1dc) * q1cd	f[xcd, ydc] * (1 - p1cd) * (1 - q1dc)	(1 - f[xdc, ycd]) * p2dc * q2cd	(1 - f[xdc, ycd]) * p2dc * (1 - q2cd)	(1 - f[xdc, ycd]) * (1 - p2dc) * q2cd	(1 - f[xdc, ycd]) * (1 - p2dc) * (1 - q2cd)
f[xdd, ydd] * p1dd * q1dd	f[xdd, ydd] * p1dd * (1 - q1dd)	f[xdd, ydd] * (1 - p1dd) * q1dd	f[xdd, ydd] * (1 - p1dd) * (1 - q1dd)	(1 - f[xdd, ydd]) * p2dd * q2dd	(1 - f[xdd, ydd]) * p2dd * (1 - q2dd)	(1 - f[xdd, ydd]) * (1 - p2dd) * q2dd	(1 - f[xdd, ydd]) * (1 - p2dd) * (1 - q2dd)

Figure 1: Transition Matrix

By the Perron−Frobenius Theorem, the stationary eigenvector exists and yields the stationary state distribution when the strategies play each other:

$$\vec{v} = [v_{1cc}, v_{1cd}, v_{1dc}, v_{1dd}, v_{2cc}, v_{2cd}, v_{2dc}, v_{2dd}].$$

In order to make the steady state distribution unique, we introduce noise into the system in order to make Q irreducible.

Given the stationary state distribution, we calculate the following statistics for a pair of strategies:

- long-term avg. payoff (for strategy i): dot product of the long-term probability of being in a state (given by the eigenvector) with the associated payoff for that state (given by the payoff matrix); i.e. $v_{1cc} * (b_1 - c) + ... + v_{2dd} * 0$.
- long-term avg. probability of being in a C state; for strategy i, this is $v_{1cc} + v_{1cd} + v_{2cc} + v_{2cd}$ (analagous for strategy j).
- long-term avg. probability of being in a Game 1 state: $v_{1cc} + v_{1cd} + v_{1dc} + v_{dd}$.

Given the current population, a strategy's average payoff, probability of cooperation, and probability of being in a Game 1 state is calculated as the average value over all the possible head-to-head matchups with the other strategies in the current population (weighted by strategy prevalence in the current population).

We can then calculate the population's collective average payoff, probability of being Game 1 state, etc. as the weighted average of each individual strategy's average payoff, etc. (weighted by the strategy's frequency in the currrent population).

4 Simulation Results

4.1 Strategy Domain

I simulated the results of evolution over 2 strategy spaces:

- Pure strategies that always prefer Game 1, with noise level $\epsilon = 0.01$. This is equivalent to the standard one-game iterated Prisoner's Dilemma with noise, which I used to check that the simulation works correctly.
- Pure strategies with noise level $\epsilon = 0.01$.

Note: pure strategies are those whose probabilities are either 0 or 1; noise changes the probabilities to be ϵ and $1 - \epsilon$, so that the transition matrix Q is irreducible and hence has a unique stationary distribution. Physically, noise can represent random errors, a "slippery hand," or uncertainties in perception (Wu & Axelrod, "How to Cope with Noise in the Iterated Prisoner's Dilemma"; Nowak & Sigmund, "The Evolution of Stochastic Strategies in the Prisoner's Dilemma").

4.2 Sanity Test

I first tested my code in the standard iterated Prisoner's Dilemma scenario, with only Game 1 as an option. As a function of b_1 (the benefit of cooperation in Game 1), the average individual payoff and average probability of mutual cooperation looked as follows:

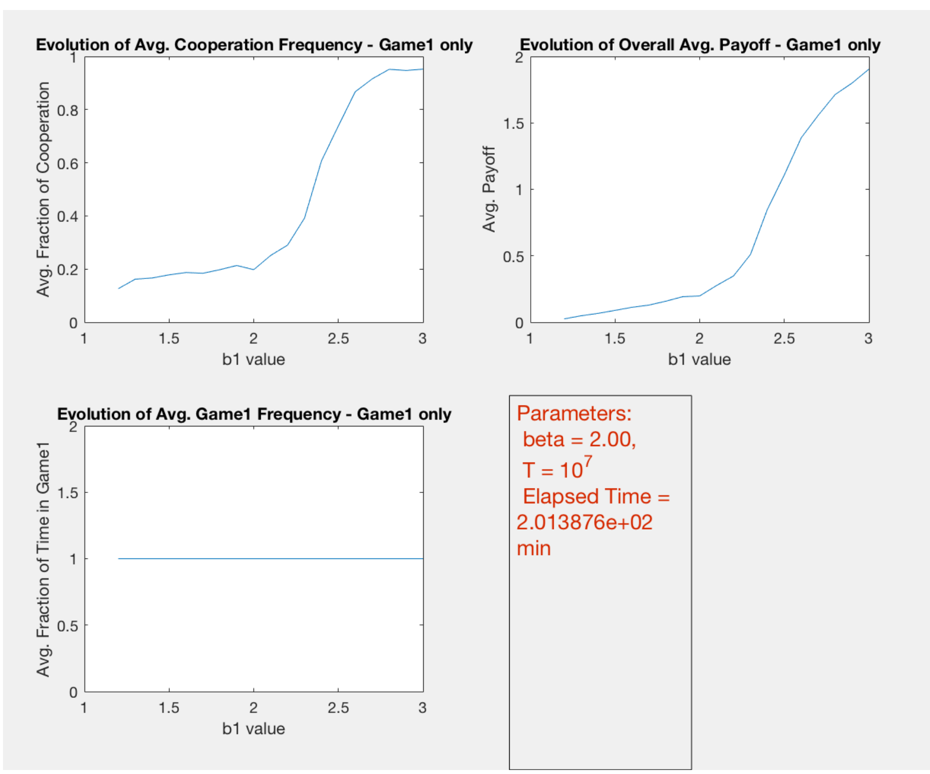

Figure 2: Dependence on b_1, with noise level $\epsilon = 0.01, T = 10^7, \beta = 2$

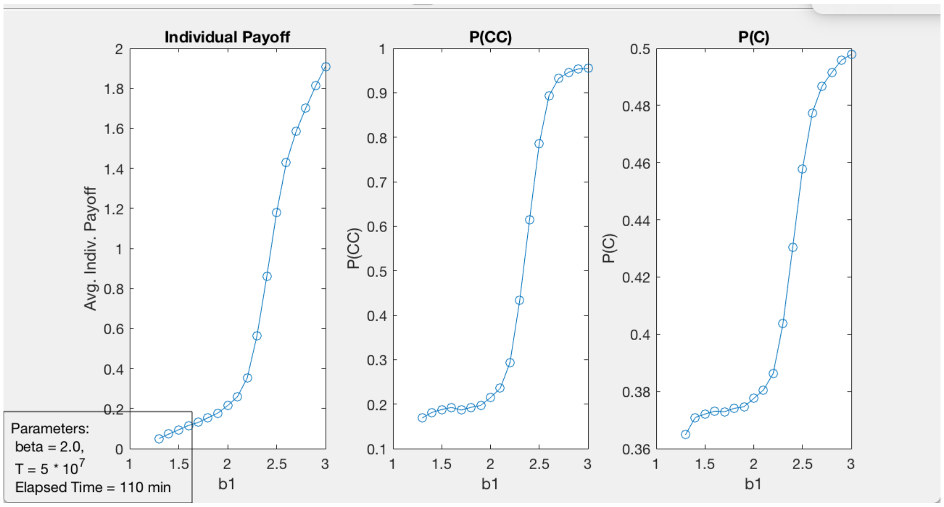

Figure 3: Dependence on b_1, with noise level $\epsilon = 0.01, T = 10^7, \beta = 2$

Here, avg. cooperation frequency refers to the average frequency of being in a mutually cooperative CC state (the evolution simulation parameters are: noise level $\epsilon = 0.01$, number of timesteps $T = 10^7$, and selection pressure $\beta = 2$).

To ensure consistency, I experimented with various T and β parameters until I found a combination that reduces variation between independent runs. With $T \geq 10^7, \beta = 2$, the average level of cooperation outputted by two independent evoultionary simulations typically does not differ by more than 0.05.

As predicted theoretically, mutual cooperation appears to begin to take root when $b_1 > 2c$, i.e. when $b_1 > 2$. The graph itself also looks similar to the evolution of cooperation under noiseless expectation.

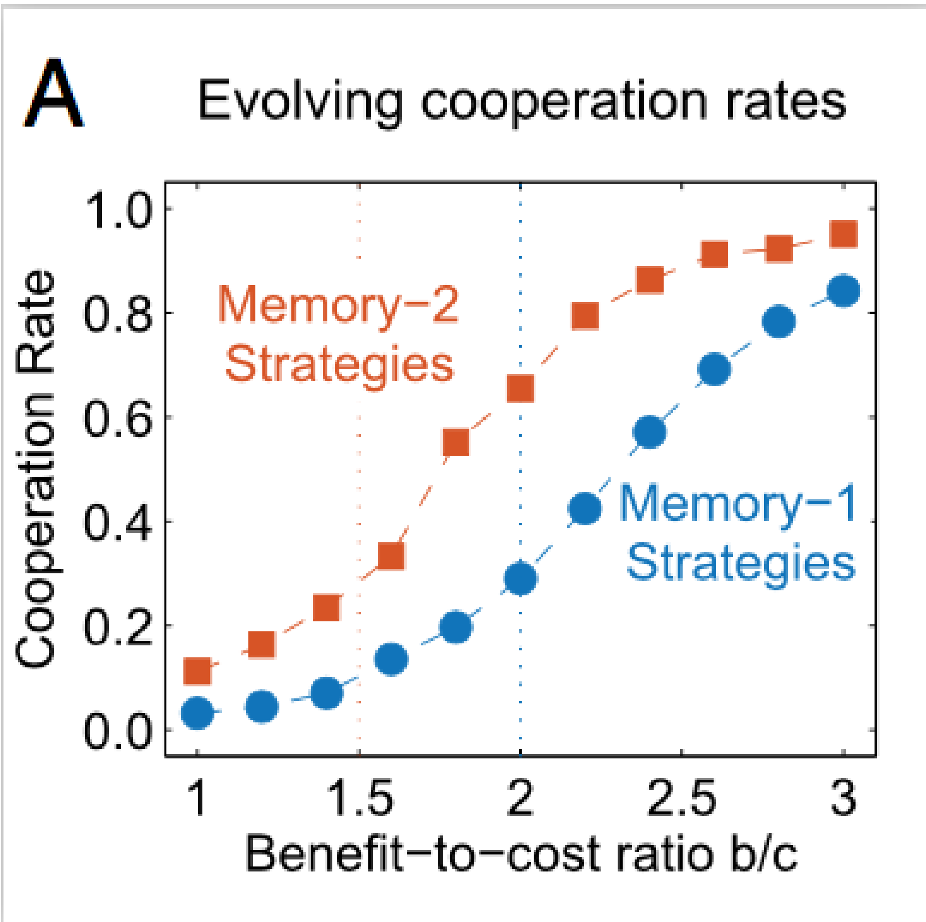

Figure 4: Theoretical dependence on b_1, no noise, with $T = 10^7, \beta = 2$

Hence I am reasonably confident that my code runs correctly.

4.3 Full Strategy Space

Running the simulation again over the full strategy space with two possible games yielded:

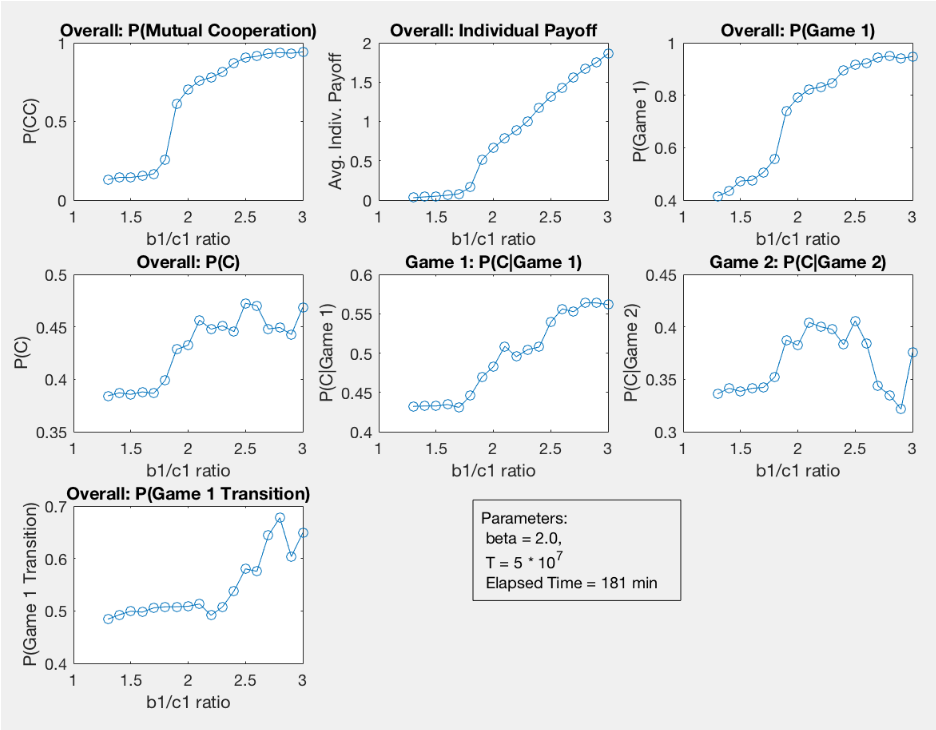

Figure 5: Dependence on b_1, with noise level $\epsilon = 0.01, T = 5 * 10^7, \beta = 2$

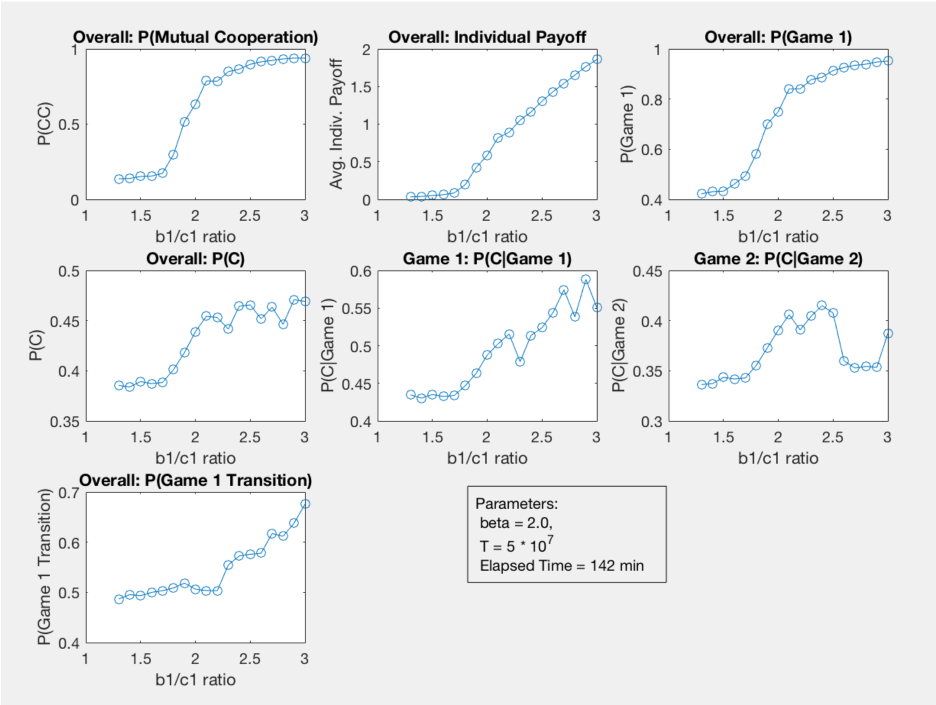

Figure 6: Dependence on b_1, with noise level $\epsilon = 0.01, T = 5 * 10^7, \beta = 2$

From top-down and left-right, the graphs depict as a function of b_1:

- long-term average probability of being in a CC state (given by strategy dynamics of the population at each timestep),
- long-term average individual payoff,
- long-term probability of being in a Game 1 state (given by strategy dynamics of the population at each timestep),
- on the level of strategies (not their interactions), the probability of cooperation broken down into Game 1, Game 2, and overall,
- on the level of strategies (not their interactions), the probability of preferring Game 1

The similarity between the independent simulation runs gives confidence that the results are replicable. A few noticable trends:

- Across the two games, mutual cooperation begins to take root earlier: in Game 1 alone, at $b_1 = 2$ we had $P(CC) \approx 20\%$ while in the two-game scenario $P(CC) \approx 50\%$. Theoretically, without noise, we know $WSLS$ is a Nash equilibrium when $b > 2c = 2$.
- As the benefit of cooperation in Game 1 rises, strategies tend to prefer to transition to Game 1. The probability of Game 1 seems to be increasing to high levels, so Game 1 seems to become more predominant as b_1 increases. (The random mutations built into the simulations prevent full homogeneity.)
- The probability of cooperating, given we are in Game 2, dips between $b_1 = 2.5$ and $b_1 = 3$. Perhaps this value marks a threshold where Game 1 is so much better that it is not worth it to peter about in the low-stakes game but to instead try to force the counterparty to opt for Game 1 if it is to receive any payoff at all. Or it could be due to the fact that as most of the encounters are in Game 1, the strategies' actions in Game 2 become irrelevant (do not affect payoff), and so mutations in Game 2 actions are no longer being significantly selected against.

4.4 Robust Strategies

In order to investigate what robust, fully cooperative strategies might look like, I ran several simulations with $b_1 = 2, T = 5 * 10^7, \beta = 2$.

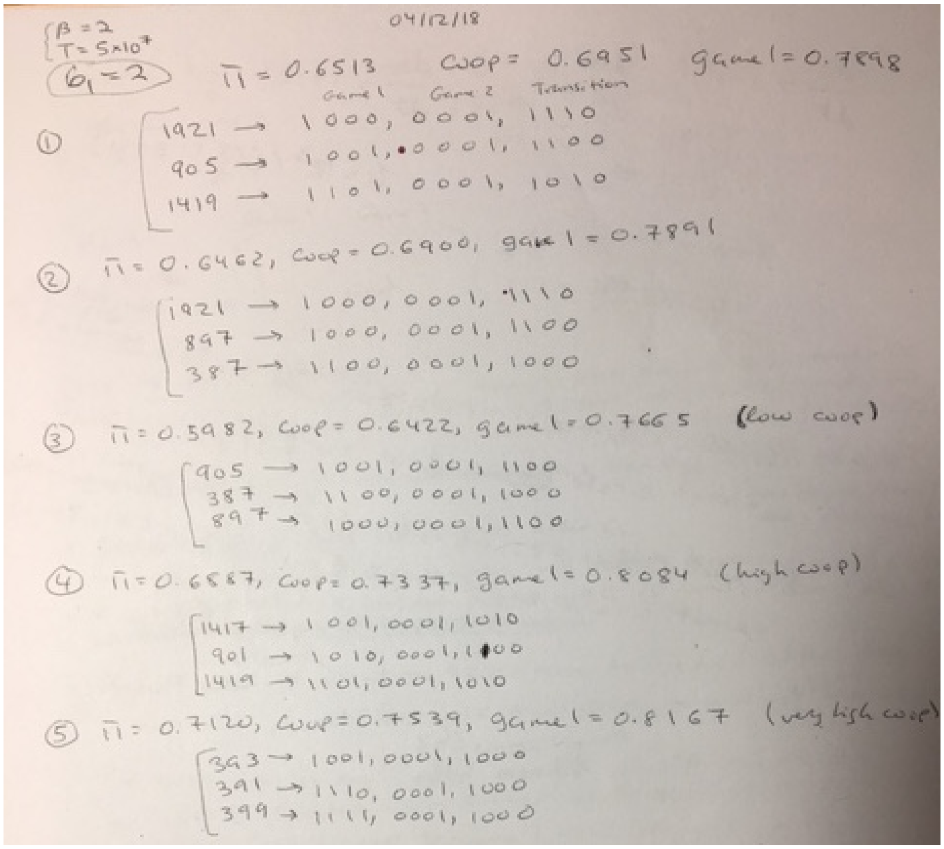

Figure 7: Most frequent strategies, with $b_1 = 2, \epsilon = 0.01, T = 5 * 10^7, \beta = 2$

I noticed that variants of strategy $A = [1000, \quad 0001, \quad 1000]$ were most frequent. In words, strategy A transitions to Game 1 only if the prior round was a CC, cooperates in Game 1 only if the prior round was a CC, and cooperates in Game 2 only if the prior round was a DD. Most importantly, strategy A has the nice property of coopertaing against itself.

5 Analytical Work

Theorem 1. *Strategy* $A = [1000, \quad 0001, \quad 1000]$ *is a subgame perfect equilibrium if* $2b_1 - b_2 > c$.

Note that $b_1 > b_2$, so this requirement for mutual cooperation is easier to meet than the corresponding threshold for a single high-stakes IPD game - $2b_1 - b_2 > 2c$ vs. $b_1 > 2c$. Thus, surprisingly, the existence of lower-stakes interactions lowers the barrier for mutually cooperative strategies to become Nash equilibria.

Proof. According to the one-shot deviation principle, a strategy is a subgame perfect equlibirium iff there are no profitable one-period deviations in all subgames. (Wikipedia, One-Shot Deviation

Principle)

Let π_{AA} denote the payoff our strategy achieves playing against itself, and let $\pi_{\tilde{A}A}$ denote the payoff the one-time deviation strategy acheives versus strategy A.

As is standard in the case of infinite horizon games, I use a discount factor $0 < \delta < 1$ to discount the value of future payoffs.

1. Subgame state: CC (Game 1).

   Since transition preferences are unaffected, the next round will be in Game 1. Baseline: since the next round will be in Game 1, and the last round was a CC, strategy A cooperates next round. One-shot deviation: strategy $\tilde{A}$ defects next round. Subsequent trajectories:

Round	Game	Result	Payoff
1	1	CC	$b_1 - c$ ⟲

(a) Baseline Payoff (π_{AA})

Round	Game	Result	Payoff
1	1	DC	b_1
2	2	DD	0
3	2	CC	$b_2 - c$
4	1	CC	$b_1 - c$ ⟲

(b) Single Deviation Payoff ($\pi_{\tilde{A}A}$)

   Hence the baseline payoff is $\pi_{AA} = \sum_{i=0}^{\infty}(b_1 - c) * \delta^i = (b_1 - c) * \frac{1}{1+\delta}$. Note that the expected number of rounds is $\frac{1}{1+\delta} = \sum_{k=1}^{\infty} k * \delta^{k-1} * (1 - \delta)$ (first round always occurs; δ can be interpreted as the probability of having a next round). So the average payoff per round is $\overline{\pi}_{AA} = b_1 - c$, as expected (pun intended). More importantly,

   $\pi_{AA} - \pi_{\tilde{A}A} = -c + (b_1 - c) * \delta + (b_1 - b_2) * \delta^2$. In the limit as $\delta \to 1$, we have that $\pi_{AA} - \pi_{\tilde{A}A} = -c + (b_1 - c) + (b_1 - b_2) = 2b_1 - b_2 - 2c$. So $\pi_{AA} > \pi_{\tilde{A}A}$ when $\boxed{2b_1 - b_2 > 2c}$.

2. Subgame state: CD (Game 1). Note that the subsequent trajectories are the same after DC (Game 1), DD (Game 1), CC (Game 2), CD (Game 2), and DC (Game 2).

   Since transition preferences are unaffected, the counterparty will prefer Game 2, and so the next round will be in Game 2. Baseline: since the next round will be in Game 2, and the last round was not DD, strategy A defects next round. One-shot deviation: strategy $\tilde{A}$ cooperates next round.

Round	Game	Result	Payoff
1	2	DD	0
2	2	CC	$b_2 - c$
3	1	CC	$b_1 - c \circlearrowleft$

(c) Baseline Payoff (π_{AA})

Round	Game	Result	Payoff
1	2	CD	$-c$
2	2	DD	0
3	2	CC	$b_2 - c$
4	1	CC	$b_1 - c \circlearrowleft$

(d) Single Deviation Payoff ($\pi_{\tilde{A}A}$)

At each step, $\pi_{AA} \geq \pi_{\tilde{A}A}$, so $\pi_{AA} \geq \pi_{\tilde{A}A}$ independent of the values of b_1, b_2, and c. In the limit of no discounting, $\delta \to 1$, $\pi_{AA} = c + \pi_{\tilde{A}A}$.

3. Subgame state: DD (Game 2).

   Since transition preferences are unaffected, the counterparty will prefer Game 2, and so the next round will be in Game 2. Baseline: since the next round will be in Game 2, and the last round was DD, strategy A cooperates next round. One-shot deviation: strategy $\tilde{A}$ defects next round.

Round	Game	Result	Payoff
1	2	CC	$b_2 - c$
2	1	CC	$b_1 - c \circlearrowleft$

(e) Baseline Payoff (π_{AA})

Round	Game	Result	Payoff
1	2	DC	b_2
2	2	DD	0
3	2	CC	$b_2 - c$
4	1	CC	$b_1 - c \circlearrowleft$

(f) Single Deviation Payoff ($\pi_{\tilde{A}A}$)

$\pi_{AA} - \pi_{\tilde{A}A} = -c + (b_1 - c) * \delta + (b_1 - b_2) * \delta^2$. In the limit as $\delta \to 1$, $\pi_{AA} - \pi_{\tilde{A}A} = -c + (b_1 - c) + (b_1 - b_2) = 2b_1 - b_2 - 2c$, so again, $\pi_{AA} \geq \pi_{\tilde{A}A}$ when $\boxed{2b_1 - b_2 > 2c}$.

Now we consider one-shot deviations in transition strategy.

1. CC transition strategy deviation.

   - Case 0: note that this one-shot transition strategy deviation does not affect the subsequent trajectory unless the prior round was a CC.
   - Case 1: the prior round was CC (Game 1).

     Baseline: both strategies prefer Game 1 and so transition to Game 1; since we will be in Game 1 after a prior CC, strategy A cooperates. One-shot deviation: strategy $\tilde{A}$ does not want to transition to Game 1, so the next round will be in Game 2. Since we are in Game 2 after a prior CC, strategy $\tilde{A}$ defects next round.

Round	Game	Result	Payoff
1	1	CC	$b_1 - c$ ↻

(g) Baseline Payoff (π_{AA})

Round	Game	Result	Payoff
1	1	DC	b_1
2	2	DD	0
3	2	CC	$b_2 - c$
4	1	CC	$b_1 - c$ ↻

(h) Single Deviation Payoff ($\pi_{\tilde{A}A}$)

As we have seen previously, $\pi_{AA} \geq \pi_{\tilde{A}A}$ when $\boxed{2b_1 - b_2 > 2c}$.

- Case 2: the prior round was CC (Game 2).

  Baseline: both strategies prefer Game 2. One-shot deviation: strategy $\tilde{A}$ prefers to transition to Game 1, but the counterparty does not, so the next round will be in Game 2 regardless and the one-shot deviation has no affect. Here $\pi_{AA} = \pi_{\tilde{A}A}$.

2. CD, DC, or DD transition strategy deviation.

   Baseline: both strategies prefer Game 2. One-shot deviation: strategy $\tilde{A}$ prefers to transition to Game 1, but the counterparty does not, so the next round will be in Game 2 regardless and the one-shot deviation has no affect. Here $\pi_{AA} = \pi_{\tilde{A}A}$.

Therefore, given $2b_1 - b_2 > 2c$, no one-shot deviation is profitable, and so the strategy $A = [1, 0, 0, 0, \quad 0, 0, 0, 1 \quad 1, 0, 0, 0]$ is a subgame perfect Nash equilibrium if the condition $2b_1 - b_2 > 2c$ is met. □

In the future, I hope to determine whether strategy A is the only subgame perfect Nash equilibrium.

6 Future Work

Possible future avenues of research:

- larger strategy domain, for example: allow for strategies whose transition preferences distinguish between whether the previous round result was in a high- or low-stakes interaction; expanding beyond pure or memory-1 strategies.

- introduce different dynamics underlying the transition to Game 1, for example: only one player needs to prefer game 1 to transition to Game 1, or the probability of Game 1 is stochastic. In general, the dynamics could be modeled by defining an arbitrary underlying function f.

- investigate the existence of different subgame perfect Nash equilibria.

7 Appendix: Code Walkthrough

7.1 Main Files

- `get_stationary_dist.m`

  Given two strategies, computes their stationary distribution.

- `get_stats.m`

  Given the benefit/cost of cooperation in Games 1 and 2, and two strategies, computes their respective payoffs, fraction cooperation, and fraction of time in Game 1 when they play against each other.

- `precompute_stationary_dist.m`

  Precomputes a 4096 by 4096 matrix containing the stationary distributions for the $2^{12} = 4096$ possible matchups between pure memory-1 strategies with noise level ϵ.

- `lookup_stats.m`

  Given the benefit/cost of cooperation in Games 1 and 2, two strategies, and the matrix of stationary distributions, computes the respective payoffs, fraction cooperation, and fraction of time in Game 1 for both strategies when they play each other.

- `run_evolution_simulation.m`

  Runs the evolution simulation for T timesteps given $b1$ (value of cooperation in Game 1), selection pressure β. If input do_plots $= true$, it draws plots of the population's average payoff, fraction cooperation, and fraction of time in Game 1 as a function of time. Calculates the mean of the population's average payoff, fraction cooperation, and fraction of time in Game 1 over the simulation, as well as the top 3 strategies, expressed as the number they represent in binary.

- `test_b1_dependence.m`

  Given the number of timesteps T and the selection pressure β, draws a plot of the population's average payoff, fraction cooperation, and fraction of time in Game 1 as a function of $b1$, where $b1$ ranges from 1.3 to 3 in 0.1 increments.

8 References

1. Nowak, Martin, and Karl Sigmund. "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game." Nature 364.6432 (1993): 56.

2. Nowak, Martin A. ”Five rules for the evolution of cooperation.” science 314.5805 (2006): 1560-1563.
3. Wikipedia, One-Shot Deviation Principle, and
4. Game Theory 101, Youtube Series, by William Spaniel