MIT 14.11: Problem Set 2

Fall 2013

Due in class on Tuesday, October 15th. If you are working with a partner, you and your partner may turn in a single copy of the problem set. You will find Chapter 4 in Martin Nowak's *Evolutionary Dynamics* very useful for this problem set. Please show your work and acknowledge any additional resources consulted.

1 Replicator Dynamic of Some 2-Strategy Matrix-Form Games

For this problem, consider the following payoff matrix:

$$\begin{array}{cc} & \begin{array}{cc} \mathbf{A} & \mathbf{B} \end{array} \\ \begin{array}{c} \mathbf{A} \\ \mathbf{B} \end{array} & \begin{pmatrix} r & s \\ t & p \end{pmatrix} \end{array}$$

where $r, p, s, t > 0$. You may find the following definitions useful:

- The fraction of the population that plays strategy A is x_A. Similarly, the fraction of the population that plays strategy B is x_B.
- The *state* is (x_A, x_B).
- The payoffs to those playing A are $f_A(x_A, x_B) = x_A r + x_B s$. Similarly, the payoffs to those playing B are $f_B(x_A, x_B) = x_A t + x_B p$. As you can see, both payoffs depend on the fraction of population playing A and B.
- The replicator equation is $\frac{dx_A}{dt} = x_A\left(f_A(x_A, x_B) - f(x_A, x_B)\right)$ where $f(x_A, x_B) = x_A f_A(x_A, x_B) + x_B f_B(x_A, x_B)$ are the average payoffs in the population.
- The *steady state* of the replicator equation are $\{x_A | \frac{dx_A}{dt} = 0\}$. That is, x_A is a steady state if, once there, we stay there.
- The *asymptotically stable steady state* of the replicator equation are

$$\{x_A | \exists \bar{\epsilon} > 0 \text{ s.t. } \forall \epsilon < \bar{\epsilon} \text{ if } x_A + \epsilon < 1 \text{ and } x_A - \epsilon > 0, \frac{d(x_A + \epsilon)}{dt} < 0 \text{ and} \frac{d(x_A - \epsilon)}{dt} > 0\}$$

  These are steady states where if we move to a nearby state, due to, say, some perturbation, the replicator dynamic moves us back towards that steady state.

1.1 Prisoners' Dilemma

In a prisoners' dilemma, $t > r > p > s$. Assume this relationship holds and answer the following:

1. For what values of x_A is x_A growing?

2. What are the steady states?

3. What are the asymptotically stable steady states?

4. Compare the steady states and asymptotically stable steady states to the mixed and pure Nash equilibria.

1.2 Coordination Game

In a coordination game, $r > t$ and $p > s$. Assume these relationships hold and answer the four questions in 1.1.

1.3 Hawk-Dove

In a Hawk-Dove game, $t > r$ and $s > p$. Assume these relationships hold and answer the four questions in 1.1.

2 Robustness of the Replicator Dynamic

When modeling a particular phenomenon, we often simplify the analysis by assuming payoffs are entirely determined by the game we're considering. In practice, fitness may be primarily determined by other factors and only slightly influenced by the game. Nonetheless, with sufficient time, evolution or learning will typically lead us to steady states that are Nash equilibria. In this problem, you'll show this.

Start by considering again the payoff matrix for a coordination game, where $r > t$ and $p > s$. Now, let's scale each of these. Specifically, replace r with $a + rb$, s with $a + sb$, t with $a + tb$, and p with $a + pb$, where $a > 0, b > 0$, a is arbitrarily large and b is arbitrarily small. This transformation is naturally interpreted as a situation in which fitness is only slightly effected by the coordination game. We will now show that evolutionary dynamics work much the same way, with the only difference that it might take longer to arrive at a steady state.

1. What are the pure and mixed Nash equilibria?

2. Write down the replicator equation for this game in terms of x_A and the transformed payoffs.

3. For what values of x_A is x_A growing?

4. What are the steady states?

5. What are the asymptotically stable steady states?

3 Numerical Estimation of Replicator Dynamics in Hawk-Dove-Bourgeois Game

In the above problems, we were able to solve the dynamics analytically. For games with more than two strategies, it is often hard to do so, and we rely instead on numerical estimations (*a.k.a.* computer simulations). In this problem, we will demonstrate and explore this technique.

Begin by considering the following payoff matrix for the Hawk-Dove-Bourgeois game:

$$\begin{array}{c} \\ \text{Hawk} \\ \text{Dove} \\ \text{Bourgeois} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Hawk} & \text{Dove} & \text{Bourgeois} \end{array} \\ \begin{pmatrix} \frac{v-c}{2} & v & \frac{3v-c}{4} \\ 0 & \frac{v}{2} & \frac{v}{4} \\ \frac{v-c}{4} & \frac{3v}{4} & \frac{v}{2} \end{pmatrix} \end{array}$$

For the duration of this problem, let $v = 2$ and $c = 3$.

1. Write down the formula for the replicator equation for each strategy.

2. Now, let's start coding. Randomly initiate the population with some frequency of hawks, doves, and bourgeoisie. You'll want to use a random number generator that draws from the uniform distribution to choose your initial frequencies. *A complete answer to this question includes both the code you used to generate the initial frequencies and output presenting the resulting frequencies. Don't forget to comment your code!*

3. The replicator equation determines how these population frequencies change from period to period. Use the equations you wrote down for part 1 to determine how the frequency of each strategy changes of time, and where the population frequencies appear to stabilize. Summarize this by graphing the frequency of each strategy over time. *A complete answer to this question includes both this graph and the code which generated it.*

4. Repeat step 3 ten times with different, randomly drawn initial frequencies. Graph any time trajectories that stabilize at qualitatively different population frequencies. Also graph any time trajectories that don't appear to stabilize. *A complete answer to this question is comprised of these graphs.*

5. Categorize the different possible "outcomes" found above. These should give you all asymptotically stable steady states, and, if they exist, any dynamics that don't converge. *A complete answer to this question includes code that generates these categories. In your comments or in prose attached to your response, please indicate how these categories compare to the Nash equilibria of this game.*

6. We will now estimate how frequently each of these outcomes occur. To do this, we need to start at lots and lots of initial conditions, each time letting the replicator dynamic take its course and classifying the outcome. We'll keep track of each outcome so that we can summarize the proportion of times we end up at each outcome after we're done running the simulation.

   Specifically, build a loop that runs steps 2, 3, and 5 10,000 times, and keeps track of the outcomes.[1] After your loop, add code that summarizes the proportion of trials that stabilized at each outcome, aggregated across all 10,000 trials. *A complete answer to this question includes both these summaries and the code that generated them.*

[1]Note: By sampling 10,000 initial conditions from the uniform distribution, we're essentially guaranteed to cover the entire range of potential initial population frequencies.

14.11 Insights from Game Theory into Social Behavior
Fall 2013