6 R *The Evolution of Relationships*

The Portuguese man o' war (*Physalia physalis*) is neither Portuguese nor a warship. It looks like a jellyfish, but it's not. It's not even an animal. Instead it's a colony of individual animals. Each individual has a specialized job, fusing together in cooperation with the others to sail and sting their way through the ocean. Like a pirate ship made of jelly.

Each man o' war colony can last for a year or more. So the evolutionary biologist must ask, how did this cooperation evolve and how is it stable against cheaters? It is the same kind of question we have to ask about more familiar relationships among animals, like cooperating lions or breeding penguins. Or actual pirates.

In all of these examples, the evolutionary game to analyze is unlike those in the previous chapters. In the previous chapters, every example was a group of individuals who met, interacted once, and then never met again. There are animal societies like that. But primate groups and man o' war colonies are not like that. They persist and the same individuals interact many times in their lifetimes.

To model the evolution of these relationships, we need to learn how to model **repeated games**. In a repeated game, individuals form groups, interact potentially many times, and use past experience to modify group membership or adjust their own behavior towards other members of the group. The question we want to ask in this framework is: Which strategies lead to the evolution of lasting cooperative relationships?

Nearly all important game theory, as applied to humans and other animal societies, concerns repeated games. And unfortunately simple games, once repeated, are no longer so simple. The longer individuals interact, the larger the possible number of strategies. And even rare errors in the implementation or the interpretation of behavior can have huge effects. But we'll begin as always with something simple and understandable, before layering on the necessary complexity.

The iterated prisoner's dilemma

In a repeated game with uncertain end, reciprocity can be evolutionarily stable against non-cooperation. But it is never stable against other cooperative strategies.

The most studied repeated game is the **iterated prisoner's dilemma**. The simple game at its heart an additive prisoner's dilemma. Suppose in a pair of individuals each has the opportunity to help the other. Helping costs the helper c. But it produces $b > c$ for the helped. This is the resulting payoff matrix:

	Cooperate	Not
Cooperate	$b - c$	$-c$
Not	b	0

In this matrix, mutual cooperation is beneficial. But each individual still does better by switching to non-cooperate, because $b > b - c$ (left column) and $0 > -c$ (right column).

Now we embed this simple game in some repeating structure, so that the same pair of individuals interact more than once. There are many ways to assume repeat interaction. The structure of repetition itself influences the evolution of behavior. Some structures make it harder for cooperation to evolve. For example, a defined end point to interaction is unhelpful. If for example the pair interacts exactly 10 times, it can be as if the game was not repeated at all. Whatever the pair does in the first 9 plays, the 10th time neither individual has any (evolutionary) incentive to cooperate. It's just like a one-shot game at that point. Then it follows that in the 9th play, it is also a one-shot game, because neither player expects (evolutionarily) the other to cooperate in the 10th play. So there's no incentive to cooperate in the 9th either. And so on, all the way back to the first play.

If there is uncertainty about the end, things can be are different. So to repeat this game, we allow a pair of individuals to play once and then interact again with probability w. And then if they do interact again, they get the same chance w again of interacting again. And so on. This w is a constant chance of interacting again after each interaction, and it does not depend upon whether the individuals cooperate or not. Of course you are ready to object. It makes sense that the persistence of the relationship depends upon how the individuals treat one another. But we start here, analyze the consequences, and then consider something more dynamic in a later section.

When the pair has a constant chance w of interacting again after each game, it means that some pairs will interact only once, while others will interact many times. The probability of interacting only once is $1 - w$. The probability of interacting twice is $w(1 - w)$. The chance of interacting three times is $w^2(1 - w)$. And so on out to infinity. Yes, the individuals will die of natural causes before that. But unless w is very close to 1, or the time between interactions is very long, that isn't an event we need to worry about. If selection is strong, then the entire distribution of pair durations could matter (as explained in **1F**). But if selection is weak, then the average duration will give us a useful approximation. And this is what nearly everyone does, use the average. Later we'll consider what this misses, however.

The average number of interactions is $1/(1-w)$. There are several ways to calculate this average. You could proceed in the orthodox manner of just defining the expected duration using the probability distribution above. Let R be the average length of a relationship. Then R must satisfy:

$$R = \sum_{i=1}^{\infty} w^{i-1}(1-w)(i)$$

The probability $w^{i-1}(1-w)$ is the chance of persisting exactly i interactions. So the above just takes each i value, multiplies it (weights it) by the chance it happens, and then adds up all these values. That's the average. The expression above is an infinite series. But it's geometric, so it closes easily. If you don't know how to do that, it just means you are a normal and healthy person. I show you how in the box below. Don't worry. It's safe.

A much easier way to derive the mean $R = 1/(1-w)$ is to realize that since the chance the interaction continues is always the same, no matter how long a pair persists, the expected number of future interactions never changes. It seems weird, but it's true. A pair just meeting for the first time expects $1/(1-w)$ interactions. And so does a pair that has already interacted 100 times. They expect $1/(1-w)$ future interactions. This implies:

$$R = 1 + wR$$

Closing the geometric series. We want to take this infinite series and find a finite expression:

$$R = \sum_{i=1}^{\infty} w^{i-1}(1-w)(i) = (1-w)(1) + w(1-w)2 + w^2(1-w)3 + w^3(1-w)4 + \ldots$$

The trick with a **geometric infinite series** is to recognize that it is always embedded in itself, because every step of the series has the same future expected value. So what we do to close it is factor it so we reveal this fact. Then it closes almost instantly. Let's factor w out of every term after the first:

$$R = (1-w)(1) + w\big(\underbrace{(1-w)2 + w(1-w)3 + w^2(1-w)4 + \ldots}_{Q}\big)$$

Now the term labeled Q is not equal to R, but it is close. Each term just has 1 added to it. Like this:

$$R = (1-w)(1) + w\big(\underbrace{(1-w)(1+1) + w(1-w)(2+1) + w^2(1-w)(3+1) + \ldots}_{Q}\big)$$

All of the those $+1$ terms just sum to 1. Why? Because the series $(1-w)(1) + w(1-w)(1) + w^2(1-w)(1) + \ldots$ is the total probability distribution. It must equal 1. This gives us:

$$R = (1-w)(1) + w\big(1 + \underbrace{(1-w)1 + w(1-w)2 + w^2(1-w)3 + \ldots}_{R}\big)$$

And now the sequence in parentheses is equal to R. So we have $R = (1-w)(1) + w(1+R) = 1 + wR$. Solving yields $R = 1/(1-w)$.

Why? Because whatever value R takes, it must satisfy the equation above so that it has the property that the expectation never changes as the pair persists. It says that the pair interacts once (1) and then has chance w of having the same expectation that they just had. So solving the above for R and get $R = 1/(1 - w)$.

Unless we also introduce strategies that use information about past behavior, repeating the game makes no qualitative difference. For example, let ALLC be a strategy that always cooperates. In a pair of ALLC individuals, each will earn $b - c$ each time they interact. So the expected fitness change is $R(b - c)$. But a rare individual who never cooperates, call this NO-C, gets b every time it interacts with ALLC, so it earns Rb. This is still a prisoner's dilemma and cooperation is not stable.

But consider a simple contingent strategy that uses past behavior. Suppose a strategy that begins by cooperating and then copies its opponent's behavior in every interaction afterwards. This strategy is usually called Tit-for-Tat (TFT). Can TFT be evolutionarily stable against NO-C? A pair of TFT individuals always cooperate with one another, because they start by cooperating and then copy one another in each interaction afterwards. So each earns $R(b-c)$. A rare NO-C earns b on the first interaction with TFT, but then TFT copies the non-cooperation on the next interaction. So both individuals earn zero until the pair ends. So TFT can be evolutionarily stable against NO-C if:

$$R(b - c) > b$$

This is easy to satisfy, if w is large enough. Consider $w = 0.9$. Then $R = 10$. Even if $b - c$ is small, multiplying it by a lasting relationship can make up for the costs of cooperation. We can also express the condition above after substituting $R = 1/(1 - w)$, as:

$$wb > c$$

So if $b/c = 2$, then w just needs to be more than 0.5. That doesn't seem so demanding. Relationships don't even have to last that long, for reciprocity to pay.

But hang on. Tit-for-Tat is rather famous in game theory.[1] It reveals the basic logic of reciprocity and how it can maintain cooperation. But Tit-for-Tat itself is a bad strategy and it is never evolutionary stable. Consider a rare ALLC in a population of TFT. The rare ALLC also earns $R(b - c)$. So ALLC can increase. If there are enough of them, NO-C can increase as well. It's like the problem with retaliation in the chapter on conflict (**4H**). Tit-for-Tat tolerates strategies that do not guard the wall.

However it seems odd that TFT and ALLC receive exactly the same payoff, in the absence of NO-C. Surely these strategies differ in other ways?

[1]Robert Axelrod wrote a book talking about how great it is.

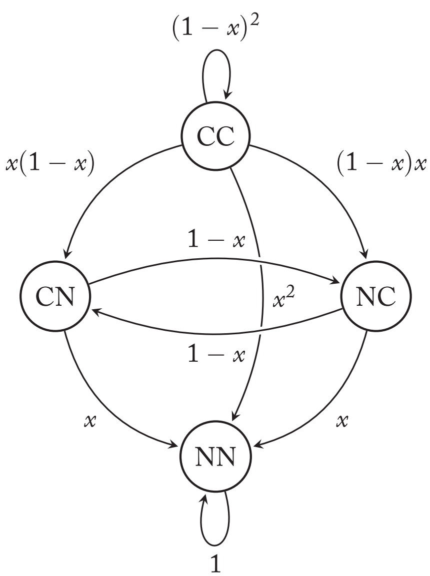
Little mistakes, large consequences

Errors matter in repeated games. If games are repeated enough, even rare errors can influence the dynamics of relationships.

Suppose an individual who intends to cooperate sometimes fails to do so. This is because cooperation requires action, and sometimes action goes wrong, no matter our intentions. Let x be the probability that an individual who intends to cooperation ends up non-cooperating instead. In game theory, this is called an **implementation error**. Implementation errors differentiate TFT and ALLC. Errors hurt both of these strategies. But TFT retaliates against an error by its opponent—it doesn't know it is an error. ALLC never retaliates.

To compute the expected payoffs in the presence of errors, we need to account for all the possible sequences of mistakes in every possible duration of the relationship. That sounds hard. But it is made easier by representing the problem as transitions between **intention states**. For a pair of TFT individuals, there are four possible pairs of intentions for each play of the prisoner's dilemma: (1) They both intend to cooperate (CC); (2) Partner intends to cooperate but focal does not (NC); (3) Focal intends to cooperate but partner does not (CN); (4) Neither intends to cooperate (NN). Implementation errors influence the transitions between these states. Let's represent it all with a diagram, which I explain below it.

Figure 6.1



Two Tit-for-Tats, in a game tree, F-E-U-D-I-N-G

The circles are the intention states and the arrows are possible transitions, labeled with probabilities. Start at the top. A pair in the CC state remains there only if neither commits an error. The probability neither commits an error is $(1-x)^2$. If both commit an error in the same play, the relationship moves to NN. This happens with probability x^2. For each state of the relationship (CC, NC, CN, NN), the same logic implies the probability of moving to any state (or staying in the same state) in the next play.

We can use this diagram to derive the expected payoff to a TFT paired with another TFT. The intuition is that we write an expression for the expected payoff at each state. We represent future payoffs by adding the other state expressions as needed. This is weird, I know. Consider the payoff for CC. Let V_{CC} be the expected payoff of a pair at the CC intention state. If neither individual commits an error, the focal TFT receives $b-c$ and then with chance w the pair interacts again at the same CC state. So the expected future payoff is wV_{CC}. So the payoff is $b-c+wV_{CC}$, but only with chance $(1-x)^2$, the chance neither commits an error. With the same logic, we can build the full expectation at CC and all of the other states. The box on the next page has the details.

It turns out that the expected payoff for TFT against TFT is:

$$V(\text{TFT}|\text{TFT}) = \frac{(1-x)(b-c)}{1-w(1-x)}$$

Errors reduce benefits (the numerator) and the effective duration of the relationship (denominator), because once the pair enters mutual non-cooperation (NN), there is no way out. But notice that when $x=0$, we recover the original $(b-c)/(1-w)$ payoff to sustained cooperation.

How does a rare ALLC do against TFT? Shown in the same box, ALLC receives:

$$V(\text{ALLC}|\text{TFT}) = (1-x)\frac{(1-xw)b-c}{1-w}$$

This looks more complicated, but it has the same structure. Errors reduce benefits. But in this case, they do not reduce the duration of the relationship (denominator), because ALLC never intentionally non-cooperates. So ALLC never gets trapped in NN like TFT does. Sometimes people say that ALLC is forgiving. But really it has no concept of a grudge in the first place.

TFT is now evolutionarily stable against ALLC when $(1-x)wb < c$. This is easier to satisfy when x is large. Errors hurt both TFT and ALLC. But TFT retaliates against ALLC's errors, while ALLC never retaliates for TFT's errors.

Errors help TFT against ALLC. But unfortunately they do the opposite with NO-C. Also in the box, I show that the condition for TFT to be stable against NO-C is $(1-x)wb > c$. This is the opposite of the previous condition. Errors hurt TFT relative to NO-C, because pairs of TFT end up feuding, when errors are common. So TFT is either stable against ALLC or NO-C, but never both.

Before you read too much into this analysis, notice that we assumed that implementation errors apply only to cooperation. Things can look quite different if

Calculating payoffs when there are implementation errors. To calculate the expected payoff for TFT when it is common, $V(\text{TFT}|\text{TFT})$, we need to write an expression for the expected payoff for each pair of *intended* actions. The first is mutual cooperation, CC. The payoff is:

$$V_{\text{CC}} = \underbrace{(1-x)^2(b-c+wV_{\text{CC}})}_{\text{neither makes a mistake}} + \underbrace{x(1-x)(b+wV_{\text{CN}})}_{\text{focal fails to cooperate}} + \underbrace{(1-x)x(-c+wV_{\text{NC}})}_{\text{partner fails to cooperate}} + x^2(0)$$

On the far right, the last term is just zero, because if ever both TFT simultaneously fail to cooperate, with probability x^2, they will non-cooperate until the relationship ends. We need similar expressions for what happens at V_{CN} and V_{NC}. These are easier, because if there is no mistake the pair just moves back-and-forth from CN to NC forever. If there is a mistake, then there is mutual non-cooperation and the relationship is essentially over.

$$V_{\text{CN}} = (1-x)(-c+wV_{\text{NC}}) + x(0) \qquad V_{\text{NC}} = (1-x)(b+wV_{\text{CN}}) + x(0)$$

Now we have three simultaneous equations and three unknowns, V_{CC}, V_{CN} and V_{NC}. If you aren't going to use some software like Mathematica to do this, I suggest solving for V_{CN} and V_{NC} first, they are like mirrors of one another and not so complicated. Then solve for V_{CC}. You should make sure you know how to solve such systems in software like Mathematica however, because eventually systems of equations get too big for a sane person to waste time on doing algebra. Of course you still need to be confident with algebra, since you need to manipulate the solutions and understand them. So practicing on this three equation system might be useful. The solutions are:

$$V_{\text{CN}} = (1-x)\frac{w(1-x)b-c}{(1-w(1-x))(1+w(1-x))} \qquad V_{\text{NC}} = (1-x)\frac{b-w(1-x)c}{(1-w(1-x))(1+w(1-x))}$$

$$V_{\text{CC}} = \frac{(1-x)(b-c)}{1-w(1-x)}$$

Since a pair of TFT individuals begin at CC, $V(\text{TFT}|\text{TFT}) = V_{\text{CC}}$ and we are done. Notice that in the limit of $x \to 0$, we recover the original mutual cooperation payoff, $(b-c)/(1-w)$. To figure out when TFT is evolutionarily stable, we now need $V(\text{NO-C}|\text{TFT})$ and $V(\text{ALLC}|\text{TFT})$. First:

$$V(\text{NO-C}|\text{TFT}) = (1-x)b + x(0) + w(0) = (1-x)b$$

We require $V(\text{TFT}|\text{TFT}) > V(\text{NO-C}|\text{TFT})$. This simplifies to $(1-x)wb > c$. For ALLC, we just need V_{CC} and V_{CN}, because ALLC always intends to cooperate.

$$V_{\text{CC}} = \underbrace{(1-x)^2(b-c+wV_{\text{CC}})}_{\text{no mistakes}} + \underbrace{(1-x)x(-c+wV_{\text{CC}})}_{\text{partner (TFT) error}} + \underbrace{x(1-x)(b+wV_{\text{CN}})}_{\text{focal (ALLC) error}} + \underbrace{x^2(0+wV_{\text{CN}})}_{\text{both error}}$$

$$V_{\text{CN}} = (1-x)(-c+wV_{\text{CC}}) + x(0+wV_{\text{CN}})$$

Now two equations and two unknowns. Solving for V_{CC}:

$$V_{\text{CC}} = (1-x)\frac{(1-xw)b-c}{1-w} = V(\text{ALLC}|\text{TFT})$$

We require $V(\text{TFT}|\text{TFT}) > V(\text{ALLC}|\text{TFT})$ for TFT to be stable against ALLC. This simplifies to $(1-x)wb < c$, which is the *opposite* of the condition for TFT to be stable against NO-C. We can reexpress these conditions in terms of x. For TFT to resist ALLC, we require $x > 1 - c/(wb)$. So if x is large enough, then rare ALLC cannot invade. Nice! But if x is large enough to keep out ALLC, then it is also large enough to allow rare NO-C to invade.

we also allow errors for non-cooperation. How can a non-cooperator end up accidentally cooperating? Suppose cooperation means "stinting," limiting harvest of a scarce resource, to allow it to regenerate. If both individuals stint, then both can harvest more in the future. Non-cooperation means harvesting. But harvesting can fail, because it requires action. Maybe the resource is a pond with fish, and the non-cooperator is bad at fishing. In the practice box below, I ask you to analyze this situation.

The details of the payoffs can change things as well. The default prisoner's dilemma is *additive*. There are no synergies when multiple individuals cooperate. This can also change things. In the second practice problem, I ask you to analyze one example.

Practice: When cooperation means not doing. Suppose implementation errors happen instead to non-cooperation intentions—an individual who intends to non-cooperate (N) instead cooperates (C) with chance y. Rederive the stability conditions for TFT against ALLC and NO-C. If you are feeling lucky, allow at the same time the previous chance x that intent to cooperate (C) results in non-cooperation (N). What qualitative conclusions change?

Practice: Synergistic cooperation. Suppose the payoff when both individuals cooperate is now $B - c$ instead of $b - c$, where $B > b > c$. Analyze the iterated game again. Does this change any of the qualitative conclusions?

Learning and mistakes

Some strategies try to learn which behavior, cooperation or non-cooperation, results in higher rewards. These strategies can do better than TFT in the presence of implementation errors.

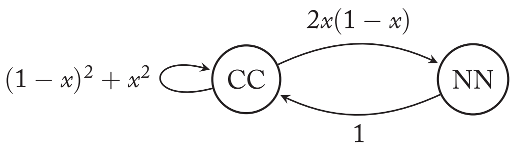
Another way to think about the iterated prisoner's dilemma, especially for animals like primates, is that individuals do arrive with fixed strategies for how to behave. Rather they are learning dynamically how to make a relationship work or whether to abandon it. Algorithmically, learning strategies are still strategies, so we don't need any new tricks (yet) to study a simple example.

Consider a learning strategy named **Pavlov**. Pavlov has an aspiration level for the benefit it derives from an interaction. If it receives a payoff at or above that level, it keeps its behavior (cooperation or non-cooperate). Otherwise it changes behavior. Suppose the aspiration level is set to $b - c$, the payoff to mutual cooperation (CC). So Pavlov doesn't change when it receives $b - c$ or greater, which includes b, the payoff to tricking an opponent (NC). If it receives instead 0 (NN) or $-c$ (CN), it switches behavior. So you see Pavlov is not a nice, moralistic strategy like TFT. It is just learning how to get a satisfactory payoff.

First consider how Pavlov behaves in the absence of implementation errors. If two Pavlov individuals begin by cooperating, they will receive $b - c$ and stay with cooperation. If they begin instead with mutual defection, they will receive 0 and switch to CC, where they'll stay. So Pavlov does well when common. That's a good start.

Now let's figure out how they do when there are implementation errors. As in the previous section, suppose x is the probability that an individual who intends to cooperate instead non-cooperates. Let's draw a finite-state diagram again, to help us understand how this calculation is built.

Figure 6.2



A pair of Pavlov individuals: Meet, play, love

Begin with the CC node. If no mistake it made, the pair remains at CC. But they also remain at CC if *both* make a mistake. Why? These are implementation errors. To the players know when they make a mistake. So when both make a mistake, they play NN, which results in zero for both. So Pavlov decides zero is bad, and then both play C on the next turn. When only one makes a mistake, they play CN or NC. Either way, one of the Pavlov individuals receives b and the other $-c$. The one who received b stays at N. The one who received $-c$ changes to N. So the next turn, they play NN. But then they immediately revert back to CC. Pavlov recovers from mistakes very quickly, because it is trying to find a good payoff. In the box on the next page, I show that when Pavlov is common, it receives on average:

$$V(\text{Pavlov}|\text{Pavlov}) = \frac{(1-x)(b-c)}{(1-w)(1+2wx(1-x))}$$

This is a bit of a mess. But notice that when $x = 0$, we recover $(b-c)/(1-w)$, the payoff to sustained cooperation. Errors reduce benefits (top), as well as the effective duration of cooperation (bottom).

We want to compare this payoff to a rare ALLC. Why? Because failing to keep out rare nice strategies was the basic flaw of TFT, remember. And errors didn't help. ALLC will never intend to non-cooperate. So we only have to consider the CC and CN nodes. In the box, I show that rare ALLC cannot invade common Pavlov. The reason is that ALLC lets Pavlov exploit at the CN node. Pavlov likes getting b, and ALLC will just keep paying $-c$, until it makes a mistake. Then both get zero, and Pavlov decides to cooperate again. This phenomenon highlights that Pavlov is not really a "cooperative" strategy in the typical sense. It just wants a good enough payoff, $b-c$ or more. And it is happy to exploit a partner to get it.

Pavlov's weakness is NO-C. It will keep trying to cooperate with NO-C, bouncing from CN to NN and back, until the relationship ends. If errors are not too common, Pavlov can be stable against rare NO-C, but it is not as easy as with TFT, which just stops cooperating with NO-C right away. See the box for the details.

Partner choice

Strategies that leave a non-cooperative partner to search for a new one must cope with the cost of search and the market for partners.

The iterated prisoner's dilemma can be interpreted in a variety of ways. We can see it as each individual having a single partner and sicking with for $R = 1/(1-w)$ plays. Or we can see it as each individual having many partners all at once. For

Pavlovian payoffs. The calculations are analogous to the TFT example in the previous section. We need only two equations for this calculation:

$$\begin{aligned} V_{\text{CC}} &= (1-x)^2(b-c+wV_{\text{CC}}) + x(1-x)(b+wV_{\text{NN}}) \\ &\quad + (1-x)x(-c+wV_{\text{NN}}) + x^2(0+wV_{\text{CC}}) \\ V_{\text{NN}} &= 0 + wV_{\text{CC}} \end{aligned}$$

Solving for V_{CC} gives us:

$$V_{\text{CC}} = \frac{(1-x)(b-c)}{(1-w)(1+2wx(1-x))} = V(\text{Pavlov}|\text{Pavlov})$$

This doesn't look so nice. But we want to compare it to the payoffs for rare invading strategies. A rare ALLC visits CC and CN nodes. These are the implied expressions for each:

$$\begin{aligned} V_{\text{CC}} &= (1-x)^2(b-c+wV_{\text{CC}}) + x(1-x)(b+wV_{\text{CN}}) \\ &\quad + (1-x)x(-c+wV_{\text{CN}}) + x^2(0+wV_{\text{CC}}) \\ V_{\text{CN}} &= (1-x)(-c+wV_{\text{CN}}) + x(0+wV_{\text{CC}}) \end{aligned}$$

Solving for V_{CC}, we find:

$$V_{\text{CC}} = \frac{(1-x)(b(1-w(1-x)) - c(1-w(1-(3-x)x)))}{(1-w)(1-w(1-(3-x)x))} = V(\text{ALLC}|\text{Pavlov})$$

This expression is even worse. We ask when $V(\text{Pavlov}|\text{Pavlov}) > V(\text{ALLC}|\text{Pavlov})$, and the answer is whenever $b > c$. ALLC cannot invade. What about are NO-C? We need only consider the NC and NN nodes.

$$\begin{aligned} V_{\text{NC}} &= (1-x)(b+wV_{\text{NN}}) + x(0+wV_{\text{NC}}) \\ V_{\text{NN}} &= 0 + wV_{\text{NC}} \end{aligned}$$

Solving or V_{NC}:

$$V_{\text{NC}} = \frac{(1-x)b}{(1-w)(1+w(1-x))} = V(\text{NO-C}|\text{Pavlov})$$

These expressions are not so nice to work with, so I suggest plotting $V(\text{NO-C}|\text{Pavlov})$ and $V(\text{Pavlov}|\text{Pavlov})$ as functions of x. As long as errors aren't too common, and b is large enough, Pavlov can be stable against both NO-C and ALLC.

example, in a group of baboons, every adult interacts repeatedly with every other adult. Some pairs sustain cooperation. Others do not. When non-cooperation means not interacting, as in the default prisoner's dilemma game, the result will look like **partner choice**: An individual will cooperate with only a subset of the other adults.

As a general model of partner choice, the iterated prisoner's dilemma is badly flawed. First, often individuals cannot partner with everyone. Relationships are special and help is limited, and so animals must choose who to help. Second, pairs who sustain cooperation may want to interact more often, so to maximize the benefits of the relationship. To explore these ideas, we need some way to model dynamic partner choice.

Imagine now that individuals interact an iterated prisoner's dilemma with only one partner at a time. Each time step, each individual can decide to interact or to dissolve the pair and search for a new partner. Relationships can end for other reasons, and the w probability of persistence remains. However the relationship ends, search takes time. There is a chance s each time step that a new partner is found. So on average is takes $S = 1/(1 - s)$ periods to find a new partner. Yes, a constant chance s is odd—what if every other individual in the local community is already in a relationship? But we start easy.

Lifespan is not infinite, and this makes search costly. Time searching is time not interacting, and time is limited. Let λ be the probability an individual survives at each time step. So the average lifespan is $L = 1/(1 - \lambda)$.

The next step is to calculate fitness. What is the lifetime fitness contribution of these pairwise relationships? Suppose TFT is common. Any pair of TFT individuals still earns $R(b - c)$. And then each TFT spends on average S turns searching for another partner. So an individual's lifespan is divided up into segments of length (on average) $R + S$. In a lifespan of length L, there will be (on average) $L/(R+S)$ of the these segments. In each segment, the individual receives (on average) $R(b - c)$ increments of fitness. Whew. So in all the lifetime fitness contribution is:

$$V(\text{TFT}|\text{TFT}) = \frac{L}{R + S}R(b - c)$$

Does partner choice help TFT resist invasion by NO-C? A rare NO-C individual will interact once with each TFT partner, earning b, before the relationship is terminated by the TFT partner. Then S turns of search before the next victim is found. So NO-C earns:

$$V(\text{NO-C}|\text{TFT}) = \frac{L}{1 + S}b$$

So TFT is evolutionarily stable when:

$$\frac{L}{R + S}R(b - c) > \frac{L}{1 + S}b$$

This condition is not so easy to interpret, because it depends strongly upon both R and S. I'm going to plot it, which will help. But first I want to manipulate it so you can see how we can extract insight from these results.

Recall that in the absence of partner choice TFT is evolutionarily stable when $R(b-c) > b$. This implies that the critical value of R is $R > b/(b-c)$. Let's compare this condition to our new one. Expressing the partner choice stability condition in terms of R:

$$R > \frac{Sb}{(1+S)(b-c)-b}$$

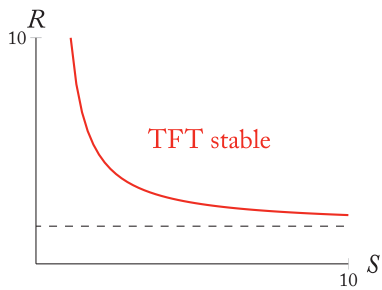
Is this ever easier to satisfy than the previous condition? We can ask the oracle of algebra:

$$\frac{b}{b-c} > \frac{Sb}{(1+S)(b-c)-b}$$

This simplifies to $c < 0$, which is never true. Partner choice never improves the stability of TFT.

It will help to plot the critical value of R as a function of S. This means to plot $R = Sb/((1+S)(b-c)-b)$. Above this curve, TFT is stable against rare NO-C. Below it, NO-C invades. Here's what it looks like:

Figure 6.3



The above is drawn for $b = 2$ and $c = 1.2$. The dashed horizontal line is $R = b/(b-c)$, the value required for TFT to be stable in the absence of partner choice. Above the red curve, TFT is stable. So you see that partner choice, in this model, never helps. The reason is because partner choice sets rare un-cooperative individuals free to victimize new partners. When a TFT sticks with NO-C, the NO-C can't extract benefits from anyone else.

Don't jump to the conclusion that partner choice never helps. There are lots of dynamics missing from this basic model. For example, when TFT is common the only process that will free up new victims for NO-C is the exogenous w end of relationships. So the expected search time should be a function of w. But this model serves as another example of how careful, logical statements of assumptions help us explore social causes in strategies contexts.