Laboratory Experiments With the

Prisoner's Dilemma

Prisoner's Dilemma Concept by Merrill Flood and Marvin Dresher
Prisoner's Dilemma Computer Tournament by Robert Axelrod
Programmed by Philippe Mathieu - http://www.lifl.fr/IPD/ipd.frame.html

Geology 200 - Evolutionary Systems
James Madison University
Lynn S. Fichter and Steven J. Baedke

Introduction

Now, finally, we get to the application of Artificial Life to the human dimension, and the Prisoner's Dilemma. There are a number of concepts developed in the past 40-50 years which have been momentous in helping us understand human behavior and history. Of these, the theories revolving around game theory (invented by Johnny von Neuman) and the Prisoner's Dilemma are certainly some of the most important. It is hard to imagine a facet of human relationships this analysis does not bear directly on, ranging from interpersonal relationships (all of you play the Prisoner's Dilemma in some form with friends, family, and significant others whether you are aware of it or not), social interactions, business, politics, war. It is so pervasive, and so important it is hard to exaggerate its importance.

The Prisoner's Dilemma was created in 1950 by Merrill Flood and Marvin Dresher at the RAND (Research and Development) Corporation as an abstract way of analyzing relationships. But it was not until Robert Axelrod at the University of Michigan attacked the problem in the late 1970's that a significant breakthrough occurred. Axelrod, was an early member of a discussion group set up and chaired by John Holland. And Holland is one of the founders, movers, and shakers in Artificial Life studies. Because of this influence Axelrod approached the Prisoner's Dilemma as an Alife research project.

Thus, because of all these connections we can in a class like this bring together all this abstract theory about chaos, complexity, and artificial life and apply it directly to our own lives. The analyses we are going to do in these experiments are fascinating in their own right. But after you come to understand them at a deeper level through these experiments they have the potential of changing the way you look at and understand the world. Maybe they will even change your behavior.

This is a large, subtle, and in some ways unwieldy subject. We could have spent the entire class just exploring the Prisoner's Dilemma and its implications. Suffice it to say we have (can) not. But we hope we can peak your interest enough to pursue it on your own. Some of these experiments today are really neat.

If you want to pursue this topic more, we recommend starting with the two books below. If you are in history, political science, or any study which concerns the relationships between or among people, or any living organisms for that matter, you really need to understand these ideas.

- Poundstone, William, 1992, Prisoner' Dilemma: Anchor Books, Doubleday, 294 pages, ISBN 0-385-41580-X
- Axelrod, Robert, 1984, The Evolution of Cooperation, Basic Books, 241 pages, ISBN 0-465-02122-0.

The Prisoner's Dilemma

Imagine you have to interact with the same person over and over. Your relationship to this person is neutral; they are not relatives, friends, or foes - just people you do business with. Yet you want to get ahead as fast and as far as you can interacting with this person, and you are willing to be deceitful if it will help. What is more, so do/will they. ***What is the best strategy you could take?***

But there is an economics to this relationship. Not all interactions are equally profitable. Both of you have the choice of cooperating or defecting each time you meet, with the payoffs of different interactions shown in the table below.

		Column Player	
		Cooperate	**Defect**
Row Player	**Cooperate**	**R=3, R=3** Reward for mutual cooperation	**S=0, T=5** Sucker's payoff, and temptation to defect
	Defect	**S=0, T=5** Sucker's payoff, and temptation to defect	**P=1, P=1** Punishment for mutual defection

There is a convention of symbols here: **T, R, P,** and **S** ("Terps")

- **T**= 5 points = temptation to defect, receives the most reward, and is always opposite S (sucker.)
- **R**= 3 points = reward for mutual cooperation.
- **P**= 1 point = punishment for mutual defection.
- **S**= 0 points = sucker's payoff. One cooperates (the sucker) while the other takes the temptation (T) to defect.

Notice that the way this is set up **T**emptation to defect must be better than **R**=cooperation, which must be better than **P**unishment. That is:

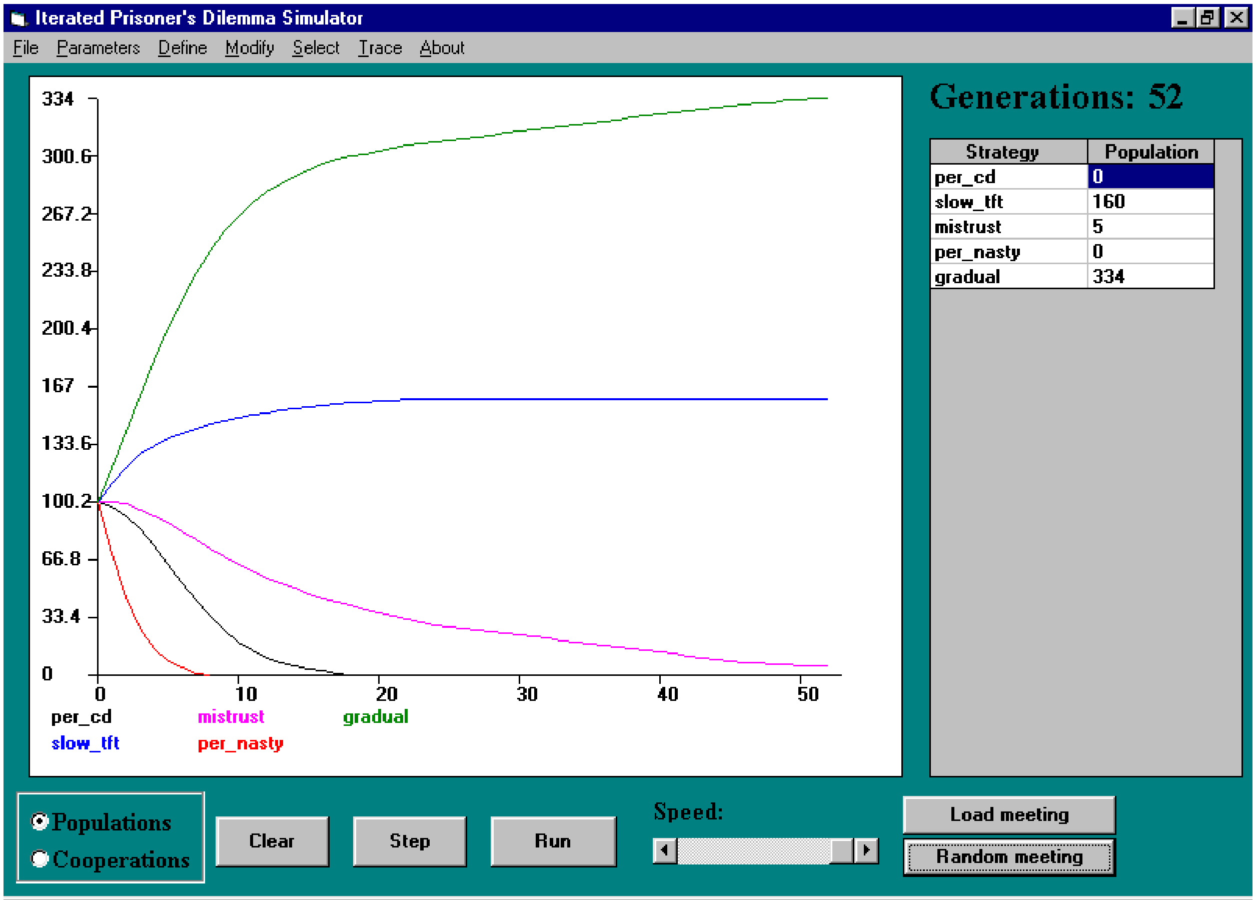
$$T>R>P>S$$

Similarly, it is useful to add the following restrictions to prevent an advantage to playing alternatively cooperation and defection (a strategy in the experiments called Per_CD - "Periodic Cooperation and Defection.")

$$2R>T+S$$

The rules here are that you both must make your decisions "simultaneously" without being able to talk to, come to an agreement with, or know what the other will do. But, you also have a record of all past interactions so you can analyze past behavior to help decide your present move.

The issue is, is there a universal solution? Is there one best strategy which always pays off in the long run, which makes you more successful. Thanks to Phillipe Mathieu we have a very nice computer program to experiment with. It is available on the Web at this address mithieu@lifl.fr (screen capture below) The page also has a very good compact discussion of the Prisoner's Dilemma (Mathieu is French so ignore some of the grammatical stumbles.)


Iterated Prisoner's Dilemma Simulator
File Parameters Define Modify Select Trace About
Generations: 52
Strategy Population
per_cd 0
slow_tft 160
mistrust 5
per_nasty 0
gradual 334
334
300.6
267.2
233.8
200.4
167
133.6
100.2
66.8
33.4
0
0 10 20 30 40 50
per_cd
slow_tft
mistrust
per_nasty
gradual
Populations
Cooperations
Clear
Step
Run
Speed:
Load meeting
Random meeting

Experiment One - Prisoner's Dilemma
Some Basic Strategies

Opening the Prisoner's Dilemma

- Prisoner's Dilemma is available in the Geology Department computer lab, Miller 232 and as a program you can download from the web to your own computer at the following address.

  http://www.lifl.fr/IPD/ipd.frame.html Get the WINPRI version.
- Click the "PrisonDil" icon in the Evolutionary Systems folder on the desk top.

Exploring the Prisoner's Dilemma

Observe that the program does not seem to be able to graph more than 10 strategies at a time even though it can calculate more than that. For example, go to SELECT: SUBTRACT ALL: ADD ALL: OK and it will conduct an ecological tournament among all 15 classic strategies, and show their final scores in the upper right, but plot only the first 10 strategies in the list.

- ***Try this,***
  - When the "Iterated Prisoner's Dilemma Simulator" is up, click **RANDOM MEETING** in the lower right hand corner. Each time you click **RANDOM MEETING** it will load several strategies and calculate a Simulated Ecological Evolution where successful strategies become more numerous and unsuccessful ones less numerous. The names of the strategies and their final population sizes are shown in the upper right of the screen.
  - Go to the lower left and click **COOPERATION** and it will show you a running total of cooperative moves made by all players during the simulation.
- Note the following:
  - Click **CLEAR** at the bottom and then **STEP** to see the game played out step-by-step.
  - Go to **MODIFY** along the top row and you see the list of classic strategies available in the program. All can be modified by clicking the **MODIFY** button. Try it just to see what is there. ***Note, each strategy is described here.***
  - In **DEFINE** you see blank versions of different categories of strategies. You can use these to build you own strategies.
  - Under **TRACE** you get various statistics of the game.
    - **MATRIX** is a matrix diagram showing the outcome of interaction between all pairs of strategies.
    - **TRACE** gives you the population numbers corresponding to the graph.
    - **MEETING** shows you step-by-step every interaction between two players, and their final scores. If you want to know why one player does better than another this is where you can find out.
  - Under **PARAMETERS** you can change how may points each move is worth (other than the 5, 3, 1, 0), how long a tournament runs, and other parameters.

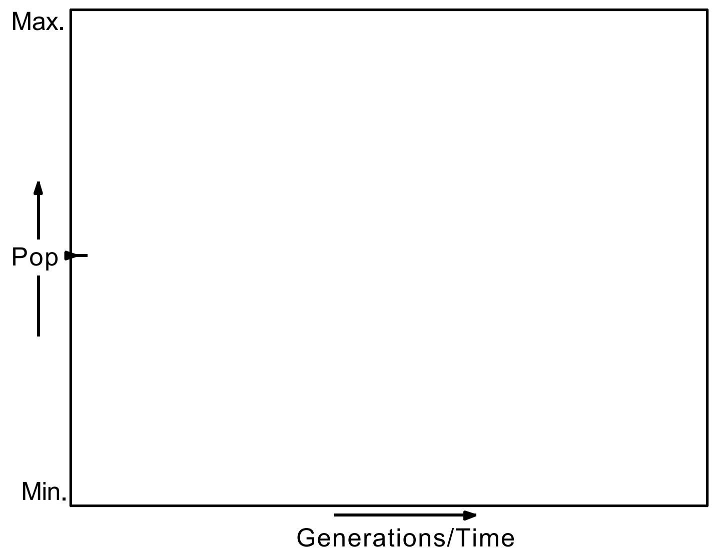
1. Checking Out the Homework.

- We hope you attempted to graph the strategies we gave you in the homework. Even if you are way off, it is valuable to get a sense of how well you see these "interpersonal" relationships (since inevitably you are going to be (have been) involved in similar interactions.)
- Repeat the homework using the computer program. Sketch the computer drawn curves over top of yours, or redraw them below, identifying them by a different color or some other way.
  - L **OBSERVE:** the computer graphs the actual population sizes rather than just "max/min". Make adjustments as best you can.
- If your graphs differ significantly from the computer drawn ones, try to explain why you drew yours the way you did.

1. Experimental Record One:

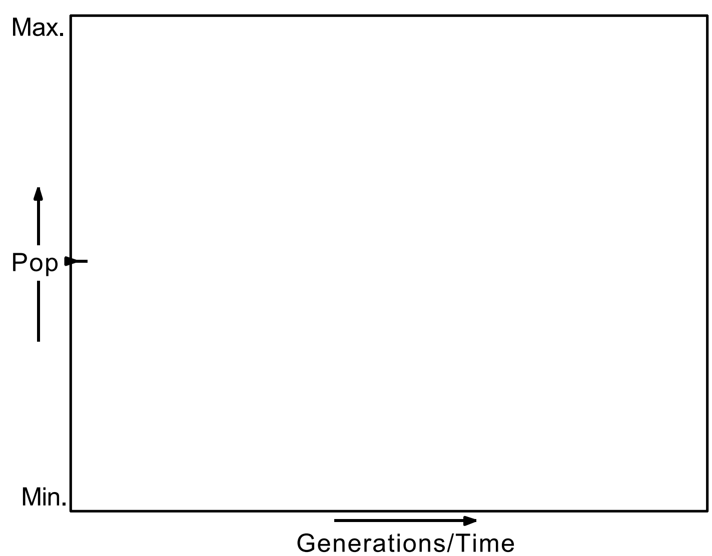
PER_CD - (Periodic CD). Alternates cooperation and defection.

GRADUAL - Cooperates until opponent defects. Then, after 1st defection it defects once and cooperates twice (D C C). With 2nd defection by opponent it defects twice and cooperates twice (D D C C). Then with each successive defection of opponent it adds one more defection of its own (D D D C C) etc.



PER_KIND - (Periodic_Kind).
Plays periodically CCD, CCD, CCD, etc.

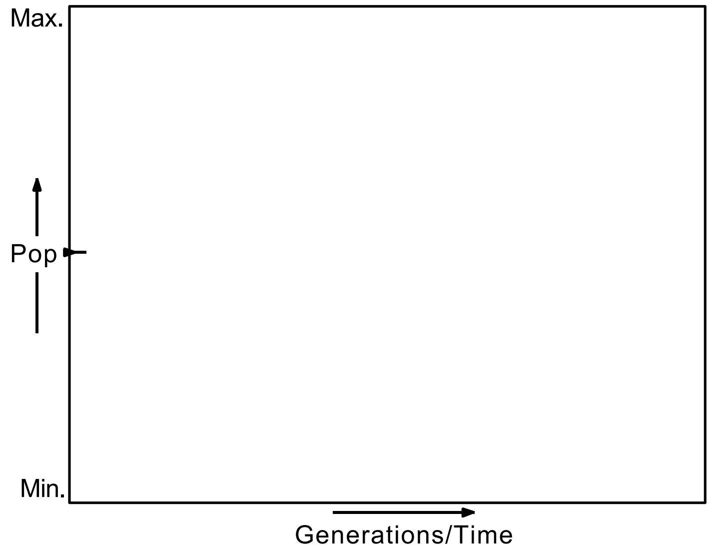
PER_NASTY - (Periodic_Nasty).
Plays periodically DDC, DDC, DDC, etc.



COOPERATE [C] - Always cooperate

DEFECT [D] - Always defect

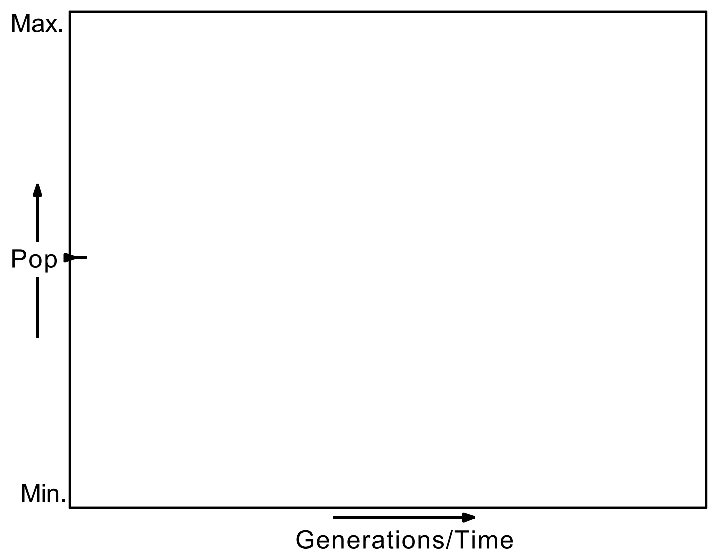
TIT_FOR_TAT - Cooperate on first move, then play what the opponent played in the last move.



COOPERATE [C] - Always cooperate

MISTRUST - Defects on first move, then plays opponents move C for C, and D for D.

SPITE - Cooperates until the opponent defects, then it always defects.



2. The Golden Rule (***"Do unto others . . . "***) is essentially the strategy "Always Cooperate." It is a nice aphorism, but does it work in the real world? Pit Cooperate against each of the other strategies one-on-one and record results in the table below.

2. EXPERIMENTAL RECORD TWO: THE GOLDEN RULE

Cooperation is Better Than These Strategies	Cooperation Works Well With These Strategies	Cooperation Works Poorly With These Strategies

Think about the strategies "Cooperation Works Poorly With". Explain, in functional terms, why Cooperation suffers so badly at the hands of these strategies. (IF you go to **TRACE: MEETING**, click on the two strategies, then click GO; it will show you a history of interaction between the two strategies.)

3. Which Strategy is Best? Obviously, there are a lot of combinations and permutations among these strategies. We could mix and match all day and night and not explore them all. So, let's try this.

1. Go to **SELECT: SUBTRACT ALL,** then **ADD** the first 10, one by one. Record your results in the table on the next page.

There is also a problem subtracting individual strategies when a lot of strategies are selected. So, . . . to save your self some grief, go to SELECT: SUBTRACT ALL: OK and remove everything. Then add back what you want. I would do this even if you need to subtract only one strategy. Subtract them all, and add back what you need.

3. EXPERIMENTAL RECORD THREE: WHICH STRATEGY IS BEST?

The Top Two-Three Strategies	The Bottom Three-Four Strategies

Is there anything in common among the TOP TWO OR THREE STRATEGIES that makes them so successful? Or, baring that, why do you think they are so successful? IF you want, go to TRACE: MEETING for more information.
Is there anything in common among the Three or Four Bottom Strategies that makes them so unsuccessful? Or, baring that, why do you think they are so unsuccessful?

Now do the same thing, only choose the bottom 10. This way the middle 5 overlap the two runs.

4. Nice, and Not Nice

- Some strategies are said to be ***"Nice"*** (never being the first to defect) while others are ***"Not Nice."*** (those which defect on first move, or will defect as much or more than they cooperate.) We could also classify ***"Mean"*** (those which try to exploit another strategy without provocation.)
  - Note: if a strategies' first move depends on what the other does, or a strategies' first move cannot be clearly determined, put it in the Non-nice category.

4. Experimental Record Four: Nice, and Not Nice.

The Nice Strategies	The Not-Nice Strategies	The Mean Strategies

Cooperate is a nice strategy, but it is easily exploited - taken advantage of - as Experimental Run 2 demonstrates. From your list of nice strategies, remove and list to the right the strategies which are ***"Nice, but not easily exploited."***	**Nice But Not Exploitable**

5. Gradual and Tit_For_Tat.

5. Experimental Record Five: Gradual and Tit_For_Tat.

One-on-one, record Gradual's response to each of the other 14 strategies.

Gradual Beats These Strategies	Gradual Does Well With These Strategies

One-on-one, record Tit_For_Tat's response to each of the other 14 strategies.

Tit_For_Tat Beats These Strategies	Tit_For_Tat Does Well With These Strategies

Specifically, in operational terms (i.e. how it actually takes place), how do Gradual and Tit_For_Tat beat the strategies they beat?

Specifically, in operational terms (i.e. how it actually takes place), why do Gradual and Tit_For_Tat do well with the strategies they do well with?

6. Nice Strategy Robustness

Robustness is strength or endurance; the ability to survive disruption. We want to see how robust Nice strategies are.

In your list above you probably came up with a lot of Nice strategies, but clearly they are not all equally effective. Tit_For_Tat is nice, but can't easily be exploited by Not-Nice or Mean strategies. So it does well. Cooperation is obviously nice, but does not fare well with Not Nice or Mean strategies.

There is also a problem subtracting individual strategies when a lot of strategies are selected. So, . . . to save your self some grief, go to SELECT: SUBTRACT ALL: OK and remove everything. Then add back what you want. I would do this even if you need to subtract only one strategy. Subtract them all, and add back what you need.

Our Strategy:

So, here is what we are going to do. Take either your first 10 strategies, or last 10 strategies from Experimental Run 3 and work with them alone.

1. Run the 10 and note the top 2 strategies, and the bottom 4 strategies.
2. Then remove the top two strategies and run it again.
3. Then remove the next two top strategies and run it again.

6. Experimental Record Six: Nice Strategy Robustness.

" The question is, how many Nice strategies do you have to remove before the originally Not Nice or Mean strategies begin to be successful? Or, do they ever?

Your 10 Strategies	Best Two	Bottom Four

Remaining 8 Strategies	**Best Two**	**Bottom Four**
Remaining 6 Strategies	**Best Two**	**Bottom Four**

Is there any point at which Not Nice or Mean strategies begin to become successful during the declining presence of the initially most successful strategies? When and how?

What, if any, other changes occur in how the strategies interact as the Nice strategies are removed?

Robustness of the Robustness: If you want to test the robustness of your conclusions in this one experiment, do the same analysis with the bottom 10, or with some random combination of 10 strategies. If you get similar results then there is greater probability these conclusions are universal.

What Should You Do? As abstract as all this analysis is, we would like to draw some lessons from it for our own behavior. ***So, in principle, under what conditions would it be best to adapt a "Not Nice" strategy, and when not?***

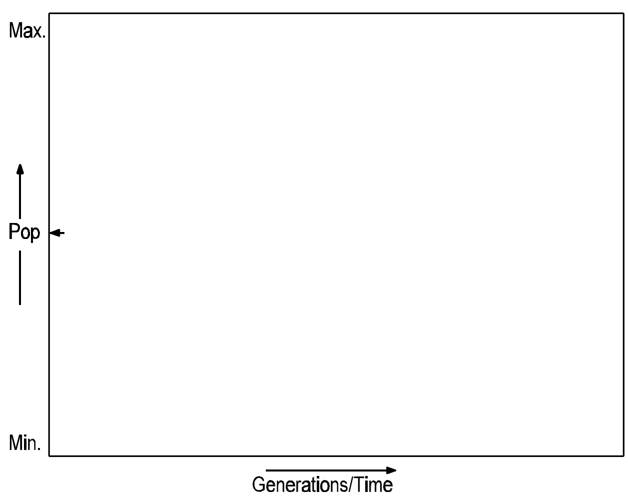
7. Random Meetings Tournaments

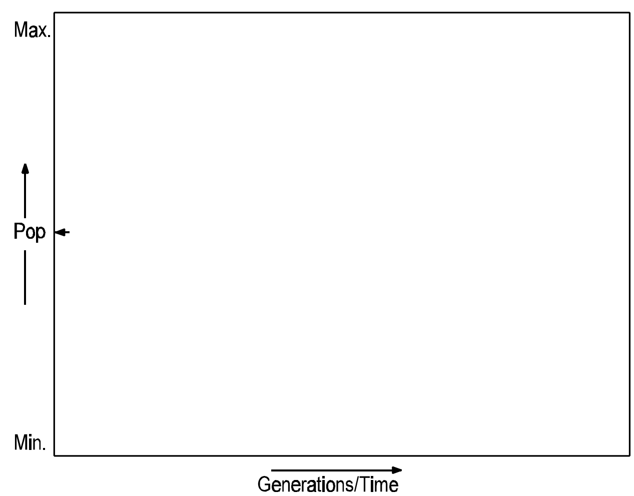
“ Want to find out how much you understand? This is a neat game to play.

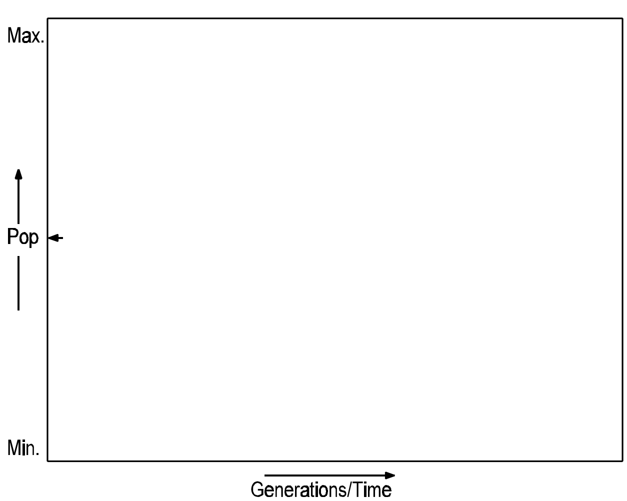
1. Go to **SELECT: SUBTRACT ALL: ADD ALL: OK.**
2. Click **RANDOM MEETING** in the lower right, and the program will load 3-5 strategies at random, calculate their interactions, and plot them.
3. Now, click **RANDOM MEETING**, and then as fast as you can the **CLEAR** button. This will prevent the strategies from calculating and plotting, but print the names of the selected strategies below the graph.
4. Examine the strategies and try to figure out which will do well and which poorly.
5. Click **RUN** and it will calculate the outcome, giving you instant feedback on your analysis.

7. Experimental Record Seven: Random Meetings Tournaments.

“ We have provided a few blank graphs in case you want to sketch out your analyses first.







8. Weird Stuff There is a possibility that in Experimental Run 5 or Experimental Run 7 you encountered some strange behavior in how the strategies interacted. We want to explore some of these here.

“ Run a simulation with the either of the following 6 or 8 strategies.

Ø Cooperate	Î Cooperate
Ù Defect	Ï Defect
Ú Random	Ð Random
Û Soft_majo	Ñ Soft_Majo
Ü Per_CD	Ò Per_CD
Ý Slow_TFT	Ó Mistrust
Þ Mistrust	
ß Pavlov	

Compared to most outcomes you have seen so far, doesn't this just blow your mind? These are obviously very unstable situations with their out-of-phase oscillations, and we found them by accident.

One is almost inclined to think that information flow is too fluid here (“r” is too high), a chaotic attractor. It is analogous to the three body problem we examined with the GALAXY experiment. Think about what this would be in a real life situation - what a turbulent period of history to live in ! (Reminds me of the French Revolution.)

But analyzing and understanding what is going on would not be easy here - there are too many interacting strategies (called degrees of freedom, 1 degree for each strategy). But there have been periods of history like this - revolutionary times when all is chaos, and no one understands what is going on, or knows how to rectify the situation.

Now, revolutions are sometimes necessary - like mass extinctions. Old, ineffective strategies can get intrenched, and new innovative strategies cannot get established, and the only way to break the logjam is through revolution. But, there are other revolutionary times that are just destructive, because things have gotten out of balance, or Not Nice or Mean strategies have appeared in force. And so we want to understand these situations, often in retrospect, so we can prevent the same conditions from appearing again.

8. Experimental Record Eight: Weird Stuff.

“ Can you make sense out of what is happening here with these 6 or 8 strategies? Perhaps if we understood more about simpler systems we could better understand a more complex system. So, let's examine some simpler systems.

Load and plot these 3 strategies: } Soft_majo, } Per_cd, } Per_nasty.

“ Now plot just } Per_cd, vs. } Per_nasty.

“ Now plot just } Soft_Majo, vs. } Per_nasty.

? Do you think you would get the same response with } Soft_majo vs. } Per_cd?

We know from earlier studies that oscillations are not going to happen with 2 strategies. We need to add a 3rd strategy, and not just any third strategy. You can check this out by clicking **RANDOM MEETING** over and over. Most 3, 4, or 5 strategy combinations produce pretty straight forward outcomes.

Give it a try, click RANDOM MEETING over and over. How many produce oscillations?

But under the right conditions, the presence of a 3rd strategy has the ability to destabilize the other two. It is useless, of course, to ask which is doing the destabilizing. There is no blame here. It is like asking in the 3-body problem which star is causing the chaotic behavior. It is not any one of them, but the synergistic interaction of the 3 together - it is an emergent property.

Analysis: Return to these 3 strategies: } Soft_majo, } Per_cd, } Per_nasty.

Think about these interaction between Per_nasty and Per_cd.

Per_cd - C D C D C D C D C D C D C D C D C D C D C
Per_nasty - D D C D D C D D C D D C D D C D D C D D C
0 1 3 1 0 5 0 1 3 1 0 5 0 1 3 1 0 5 0 1 3 = 34
5 1 3 1 5 0 5 1 3 1 5 0 5 1 3 1 5 0 5 1 3 = 54

1. Per_cd defects on only 3 of Per_nasty's cooperations.
2. Per_nasty defects on 7 of Per_cd's cooperations.

ˆ Per_nasty is going to garner more points than Per_cd, and this shows up not only when Per_nasty beats out Per_cd in a one-on-one meeting, but also later in this tournament after Soft_majo has gone extinct.

But in the first 10–12 moves the roles are reversed. Per_cd thrives while Per_nasty suffers. Why? To understand, examine the interaction of Soft_majo and Per_nasty.

L Go to **TRACE: MEETING** and record how these two interact.

Soft_majo													
Per_nasty													

Per_nasty does not do well against Soft_majo because Per_nasty's strategy of defecting twice for every cooperation is D>C, and this causes Soft_majo to defect on every move after the first one. It doesn't pay to mess around with Soft_majo.

ˆ Initially Per_nasty is damaged by Soft_majo and declines up to the first 8-9 moves.

Now, we have to examine the relationship between Soft_majo and Per_cd.

Soft_majo													
Per_cd													

You observe that Per_cd does really well against Soft_majo, resulting in its population initially climbing very high. Conversely, Soft_major's population is declining.

But, by move 20 Soft_majo is extinct, so that now it is only Per_cd and Per_nasty . . . and in a two way face off between these two we know the outcome, Per_cd declines to extinction and Per_nasty maximizes.

Could we have figured all this out before the tournament? Maybe. But what about with 6 strategies, or 8, or a dozen, or a city full of merchants? Gives new meaning to the "Computational Viewpoint." Sometimes it is easier to just watch it, calculate it, and see what happens bottom-up.

9. Tournaments: More Weird Stuff.

Despite it being easier to just watch it happen, we humans often do not have that luxury. Some "tournaments" would just be too destructive, and history is full of situations we would rather have avoided. Likewise the future is full of new situations where it would really be helpful to understand ahead of time what might happen. This is the great power of Alife studies, they allow us to experiment bottom up. But that does not obviate the necessity of some bright, intelligent, educated person understanding why.

You are that bright, intelligent, educated person! And because of that we encourage you to work with classmates to analyze and understand these.

9. Experimental Record Nine: Tournaments: More Weird Stuff.

" We have provided several more oscillating tournaments for you to ponder, with tables for **Trace: Meetings** so you can do an analysis. Can you figure out why they behave as they do?

- Print out the tournament if you want to have a hard copy to analyze.
- Data sheets are on the next page.

Tournament #1 - Load and plot these strategies: Explain the oscillatory behavior.

- Soft_majo
- Per_cd
- Defect

Tournament #2 - Load and plot these strategies: Explain the oscillatory behavior.

- Per_nasty
- Tf2t
- Mistrust
- Per_cd

Tournament #3 - Load and plot these strategies: Explain the oscillatory behavior.

} Random
} Tf2t
} Defect
} Cooperate

Tournament #4 - Load and plot these strategies: Explain the oscillatory behavior.

- Per_kind
- Per_cd
- Slow_tft
- Random

Tournament #5 - Load and plot these strategies:

- Go to the lower right part of the screen and click "Load Strategies." Try out any of the strategies, but several from Sample 9.met through Sample 15.met have oscillatory meetings.

10. General Conclusions.

" The Prisoner's Dilemma has been an extremely insightful tool for understanding behavior, and its consequences. Axelrod in his book "The Evolution of Cooperation" spends some time drawing insightful conclusions from analysis of his tournaments.

So, what general conclusions for individual and collective human behavior can we draw from all this analysis? What practical advice can we take away from this that will reduce conflict, while maximizing the non-zerosumness of the world?

10. Experimental Record Ten: General Conclusions.

" Below is a list of aphorisms, some well known, some we made up or from Axelrod. ***These are practical decisions, not moral or ethical ones.***

G = Good Lesson **M** = Misses the Point

Circle your choice

G or **M** 1. It pays to show you are tough by defecting on the first move.

G or **M** 2. A complicated strategy which obscures what you are doing, and keeps others off guard usually works better than a simple strategy.

G or **M** 3. To forgive easily brings big rewards in the long run.

G or **M** 4. An occasional defection never hurt anyone, or, it pays to be sneaky.

G or **M** 5. Envy and greed, although not nice, produce greater rewards in the long run.

G or **M** 6. If your are clever you can usually get away with things.

G or **M** 7. It pays to retaliate, defection for defection, but not more.

G or **M** 8. Nice guys finish last.

G or **M** 9. If I help others increase their profits, I will get ahead myself.

(F (i (n (i (s (

Prisoner's Dilemma Classic Strategies

COOPERATE [C] - Always cooperate

DEFECT [D] - Always defect

RANDOM - Cooperates with probability of ½; also defects with probability of ½

GRADUAL - Cooperates until opponent defects. Then, after 1st defection it defects once and cooperates twice (D C C). With 2nd defection by opponent it defects twice and cooperates twice (D D C C). Then with each successive defection of opponent it adds one more defection of its own (D D D C C), always ending with 2 cooperations.

SOFT_MAJO - (Majo = majority). Cooperates on first move, then plays opponents majority moves. If C$D then cooperate. If D>C then defect.

TIT_FOR_TAT - Cooperates on first move, then plays what the opponent played in the last move.

PER_CD - (Periodic CD). Alternate cooperation and defection.

SLOW_TFT - (Slow Tit_For_Tat). Cooperates on first two moves, then if opponent plays two consecutive times the same move (CC or DD) play the opposite move (CC= D, and DD=C).

MISTRUST - Defects on first move, then plays opponents move C for C, and D for D.

PAVLOV - Cooperates if and only if both players opted for the same choice [C & C, or D & D] in the previous move. Otherwise, defect.

SPITE - Cooperates until the opponent defects, then it always defects.

TF2T - (Tit_For_2_Tats). Cooperates except if opponent has defected two consecutive times.

HARD_TFT - Cooperates on the first two moves, then defects it the opponent has defected at least one time in the past two moves, otherwise C.

PER_KIND - (Periodic_Kind). Plays periodically CCD, CCD, CCD, etc.

PER_NASTY - (Periodic_Nasty). Plays periodically DDC, DDC, DDC, etc.