

Computational Examination of Strategies for Play in IDS Games

Steve Kimbrough, Howard Kunreuther, Kenneth Reisman

2/20/2011

1. Introduction

This document is meant to serve as a repository for work product related to ongoing investigations of interdependent security (IDS) games, focusing broadly on computational modeling of strategic interactions. Being a repository, the document will continue to grow as work on the project proceeds.

IDS games are multi-agent games where each agent must decide whether or not to invest in risk-mitigating measures. Each agent knows that even if she fully protects herself by investing in such measures, she may still incur losses due to one of the other agents who chose not to invest. Many significant social problems fit into an IDS framework, including investing in airline or port security, taking vaccinations against disease, installing computer security software to protect from viruses, and divisions of firms undertaking risky investments that could cause the entire firm to become insolvent. A key social policy question that arises in all of these examples is how to induce enough agents invest in risk-mitigating measures so that Pareto optimal levels of investment are achieved.

In this document we report the results and analysis of computational simulations involving repeated IDS games with two players. We assume that strategies in these games can be represented as algorithms, and we employ a series of computational tournaments to investigate the performance of a diverse set of strategies under varying conditions. Our aims are to investigate what kinds of strategies do well in these games, and to understand what kinds of social policies promote increased investment in risk-mitigation.

2. IDS Games

2.1. Formal characterization

We follow Kunreuther & Heal (2003) in characterizing 2-player IDS games formally. The payoffs for the stage game are shown in Table 2.1. The interpretation is as follows. The cost of investing in security measures is c . If a player does not invest, there is a probability p that the player will incur a direct loss. A player that invests may still incur

an indirect loss if the other player does not invest. The probability of such an indirect loss is q . The cost associated with any direct or indirect loss, if it should materialize, is L . We assume that agents make decisions simultaneously.

	S	N
S	$-c, -c$	$-c - qL, -pL$
N	$-pL, -c - qL$	$-pL - (1 - p)qL, -pL - (1 - p)qL$

Table 2.1: Expected outcomes associated with investing (S) or not investing (N) in security. Payoffs are shown with Row on the left and Column to the right.

This results in the following payoffs. In the case where both players invest (S, S), the payoff for each player is certain to be $-c$. If Row does not invest but Column does invest (N, S), then Row's expected payoff is $-pL$, the probability of a direct loss times the cost of the loss. In this case, Column's expected loss is $-c - qL$, the cost of investing in protective measures minus the expected loss due to contagion. If Row does invest but Column does not (S, N), then the payoffs are the same with the roles reversed. Finally, if neither player invests (N, N) then there are two sources of expected loss for each player. First, a direct loss may occur, in which case the expected loss is pL . Second, an indirect loss might occur. We are assuming that an indirect loss will not occur if a direct one does. Thus, the probability of a direct loss not occurring and an indirect loss occurring is just the product of these two quantities: $(1 - p)q$ and the associated expected loss is $(1 - p)qL$. Summing everything up we get the entries shown for (N, N) in Table 2.1.

The Stochastic Prisoner's Dilemma (SPD) is a special case of the IDS Game. An IDS game becomes an SPD game when $pL + (1-p)qL > c > pL$ so that (N, N) is a dominant solution, but both individuals would be better off if they had decided to invest in protection (S, S). While the realized payoffs in an SPD game vary stochastically from round to round, the *expected* payoffs in each round correspond to the fixed payoffs in a Deterministic Prisoner's Dilemma (DPD) game.

2.2. Empirical studies

Two laboratory studies have investigated behavior in 2-player IDS games.¹ Kunreuther et al. (2009) investigated three parameterizations of the Kunreuther & Heal (2003) model:

- (i) $p=q=.2, c=12, L=50$
- (ii) $p=q=.4, c=12, L=25$
- (iii) $p=q=.6, c=12, L=19$.

¹ Other empirical studies (Hess, Holt, and Smith, 2007; Shafran, 2010) have investigated variations of the IDS model with more than 2-players.

These three parameterizations are all instances of SPD games and they have roughly the same expected payoffs (see, for example, Table 2.2).

	S	N
S	-12, -12	-22, -10
N	-10, -22	-18, -18

Table 2.2: Expected payoffs for $p=q=.2$, $c=12$, $L=50$.

In addition, Kunreuther et al. investigated play in a DPD game with fixed payoffs (Table 2.3) similar to the expected payoffs in their SPD games.

	S	N
S	-12, -12	-22, -10
N	-10, -22	-16, -16

Table 2.3: Payoffs for the DPD game investigated in Kunreuther et al (2009).

Kunreuther et al. investigated two different conditions for the SPD games: a full-feedback condition and a partial-feedback condition. In the full-feedback condition, players were able to see what actions their counterparts had taken in the previous round, whether a loss has occurred, and a random integer r that was used to generate the outcomes.² In the partial-feedback condition, players were not told what actions their counterparts had taken in the previous round, but they were able to see whether a loss had occurred and the random integer r . Hence, while players in the full-feedback condition were always aware of their counterpart's action, players in the partial-feedback condition were able to infer this information only under certain circumstances.

Gong, Baron, Kunreuther (2009) conducted a different laboratory study of 2-player IDS games. They investigated SPD games with the parameterization $c=45$, $p=.4$, $q=.2$, $L=100$. This results in the expected payoffs shown in Table 2.4.

² To compute whether a loss has occurred, a random integer r is drawn from the uniform distribution $[1,100]$ after each round. If $r \leq 100p$, then both players experience a loss if at least one player has not invested. If $100p < r \leq 100[p + (1 - p)q]$, then both players experience a loss if both players have not invested. If $100[p + (1 - p)q] < r$, then no loss occurs.

	S	N
S	-45, -45	-65, -40
N	-40, -65	-52, -52

Table 2.4: Expected payoffs for $c=45, p=.4, q=.2, L=100$

In addition, they investigated play in a DPD game with the fixed payoffs identical to those in Table 2.4.

2.3. Settings used in this study

In this document we investigate a specific parameterization of the 2-player IDS model: $p=q=.2, c=12, L=50$. This was one of the three SPD parameterizations investigated in Kunreuther et al. (2009). The expected payoffs for these parameters are shown in Table 2.2. In our computational simulations, players are matched in round robin fashion with all other players plus their twin. Each game consists of 20 repeated rounds, and each pairwise match consists of 6000 replicated games. Simulation software was written in the python language.³

We conduct computational simulations under both full-feedback and partial-feedback conditions. In our full-feedback condition, agents are given information about their counterpart's action in the prior round, as well as whether there has been a loss. In our partial-feedback condition, agents are informed only whether there has been a loss. Unlike the partial-feedback condition in Kunreuther et al (2009), we do not provide the random number used to generate outcomes as information to the two agents. Thus, if an agent does not invest and a loss subsequently occurs, the agent would not be able to discern whether the loss was due to its own decision or to the decision of its counterpart—that is, the agent would not be able to discern whether its counterpart has invested. An agent could infer that its counterpart has not invested only in the case where the agent *does* invest in a given round and a loss subsequently occurs.

3. Strategies to be explored

In this section, we detail the 26 strategies to be explored in our computational simulations. We distinguish three classes of strategies based on the type of exogenously provided information employed by each class.⁴ Class 1 strategies employ information about whether the partner has invested in the round. Class 2 strategies employ

³ Ideally, it would be desirable to run a larger number of replications than 6000. The number 6000 was chosen due to computer memory limitations. The simulations reported here were performed on an Apple Powerbook Pro with a 2.4 GHz Intel Core Duo processor and 2 GB of RAM.

⁴ Many of the strategies and specifications here are drawn from Fudenberg, Rand, and Dreber (2010).

information about whether a loss has occurred in the prior round. Class 3 strategies do not employ any exogenously provided information. All three classes of strategies are applicable to the full-feedback condition (a total of 26 strategies). Only classes 2 and 3 are applicable to the partial-feedback condition (a total of 15 strategies).

In the following tables we provide a name, summary, and precise specification for each strategy. Many of the specifications employ finite-state machine diagrams. In these diagrams, the strategy begins each round in one of several possible states. Each state is represented by a circle and the strategy's action for that state is shown in the center of the circle. At the beginning of the game, the strategy is always in the *leftmost* state. The arrows between states indicate transitions for each possible message that may be received by the agent in a given round. Specifically, the message "S" implies that the partner invested, "N" implies that the partner did not invest, "L" implies that a loss occurred, "~L" implies that a loss did not occur, and "All" implies that a state transition should be taken regardless of the message.

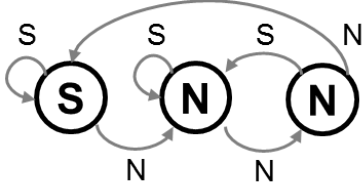
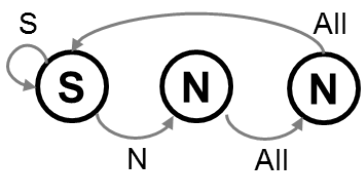
To illustrate how these diagrams work, consider TFT. TFT begins the round 1 of the game in the leftmost state, and thus will play "S" in the first round. Beginning in round 2, TFT receives information about the counterpart's action in the prior round: either "S" if the counterpart invested, or "N" if not. Notice that there is an arrow labeled "S" directed from the leftmost state back to itself. This implies that if the counterpart invested in the prior round, then TFT will remain in the same state and play "S" again. Notice also that there is an arrow labeled "N" directed from the leftmost state to the rightmost state. This implies that if the counterpart did not invest in the prior round, then TFT changes to the rightmost state and will play "N" instead. Similarly, if TFT is in the rightmost state and the counterpart plays "N" then TFT remains in that state, whereas if the counterpart plays "S" then TFT will revert to the leftmost state.

The process is similar for strategies in class 2, except that the strategies receive either message "L" (a loss has occurred) or "~L" (no loss has occurred) rather than "S" or "N". For class 3 strategies, the sequence of state transitions is predetermined and does not depend on the messages received at all.

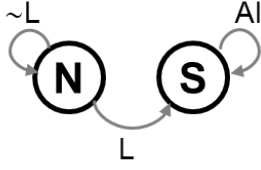
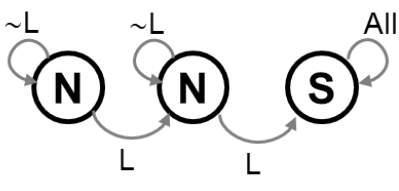
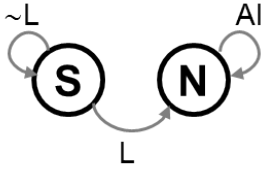
Class 1: Strategies that depend on partner's decision in the prior round

<i>Name</i>	<i>Summary</i>	<i>Specification</i>
TFT	Tit For Tat	

TF2T	Tit For 2 Tats	
DTFT	Same as Tit For Tat, but do not invest on first move	
2TFT	2 Tits for 1 Tat	
2TF2T	2 Tits for 2 Tats	
Grim	Invest until partner does not invest, then never invest	
Grim2	Invest until partner does not invest twice in a row, then never invest	
PTFT	Invest on first move. Each time partner does not invest, then shift to a different action	

2PTFT	Similar to PTFT, but do invest for at least two rounds before shifting back to invest	
T2	Invest in first move; each time I invest and partner defects, then do not invest for exactly two moves before investing again	
FictitiousPlay	Invest in first round. Then perform action that would maximize payoff for next round, assuming that the partner's most frequent action in all prior rounds will be played in the next round.	Difficult to represent as a finite state machine. See the appendix for sample code.

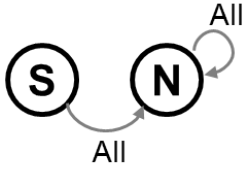
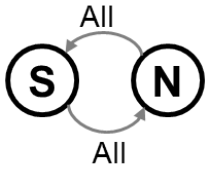
Class 2: Strategies that depend on whether a loss has occurred in the prior round

<i>Name</i>	<i>Summary</i>	<i>Specification</i>
AlwaysInvestAfter1Loss	Do not invest until a loss occurs, then always invest	
AlwaysInvestAfter2Losses	Do not invest until two losses occur, then always invest	
NeverInvestAfter1Loss	Invest until one loss occurs, then never invest	

NeverInvestAfter2Losses	Invest until two losses occur, then never invest	
Invest1AfterLoss	If a loss occurs, then invest in the next round; otherwise do not invest	
Invest2AfterLoss	If a loss occurs, then invest in the next two rounds; otherwise do not invest	
DoNotInvest1AfterLoss	If a loss occurs, then do not invest for the next round; otherwise invest	
DoNotInvest2AfterLoss	If a loss occurs, then do not invest for the next two rounds; otherwise invest	

Class 3: Strategies that do not incorporate additional information

<i>Name</i>	<i>Summary</i>	<i>Specification</i>
AlwaysInvest	Always invest	All
NeverInvest	Never invest	All

I-N	Invest on first move, then never invest	
Alternate	Invest on first round, then alternate on every round	
Random0.2	Invest with 0.2 probability	Invest with 0.2 probability
Random0.5	Invest with 0.5 probability	Invest with 0.5 probability
Random0.7	Invest with 0.7 probability	Invest with 0.7 probability

4. Results of play

In this section, we report the results of our baseline simulations under both full-feedback and partial-feedback conditions.

4.1. Full-feedback condition

For each strategy, we report total payoff per game as averaged across opponents and all replications. Payoffs for each round are scaled between zero and unity using the function $(\text{payoff}-lo)/(hi-lo)$, where hi is the highest possible payoff in a round (zero) and lo is the lowest possible payoff in a round (-62). As there are 20 rounds per game, the maximum possible total payoff is 20 and the minimum is zero. When sorted by decreasing average payoff per game, the rankings under full-feedback are as follows:

<i>Rank</i>	<i>Strategy</i>	<i>Average payoff</i>
1.	GRIM	15.65
2.	GRIM2	15.63
3.	2TFT	15.58
4.	NeverInvestAfter1Loss	15.55
5.	2TF2T	15.53
6.	2PTFT	15.52
7.	TF2T	15.52
8.	TFT	15.51
9.	T2	15.44
10.	NeverInvestAfter2Losses	15.43
11.	PTFT	15.41
12.	DoNotInvest1AfterLoss	15.39

13.	DTFT	15.39
14.	DoNotInvest2AfterLoss	15.32
15.	AlwaysInvest	15.25
16.	FictitiousPlay	15.08
17.	I-N	15.08
18.	NeverInvest	15.06
19.	Alternate	14.99
20.	Random0.2	14.98
21.	Invest1AfterLoss	14.93
22.	Random0.5	14.90
23.	Random0.7	14.89
24.	Invest2AfterLoss	14.87
25.	AlwaysInvestAfter1Loss	14.83
26.	AlwaysInvestAfter2Losses	14.82

We see here that the best performing strategies are GRIM and GRIM2, but in fact the top 12 or so strategies all have broadly similar properties: they are “nice”—that is, unless provoked, they will always invest—but they will also retaliate sooner or later if their counterpart does not invest. These properties allow them to reap the benefits of cooperation while being difficult to exploit.

We can also examine the distribution of game outcomes for each strategy using boxplot diagrams (Figure 4.1). The boxplot diagrams can be understood as follows. For each strategy, we keep track of the total payoff per game for every game in every match. Since each strategy plays 26 matches (in the full feedback condition) with 6000 games per match, this results in $26 \cdot 6000 = 156,000$ data points for each strategy. We then plot the distribution of payoffs using a separate boxplot for each strategy. In the boxplots below, the upper line represents the maximum payoff in the distribution, the second line from the top represents the 25th percentile, the dark middle line represents the median, followed by the 75th percentile, and finally the minimum. Each white box represents the range between the 25th and 75th percentile payoffs for a given strategy.

In Figure 4.1 we can see that the white boxes are smaller for the top 12 performing strategies than for the others, and that the white boxes tend (roughly) to increase as we move from rank 1 to rank 26. This implies that there is typically less variance in payoff among the top performing strategies. We can also see that the median payoff among the top 12 strategies is quite similar, and that the media payoff is located at the top of the white box for these strategies, implying that they have similar performance and that most of the variation in their performance is on the downward side. This makes sense when we consider that the top 12 strategies are all nice strategies that, by default, invest in every round.

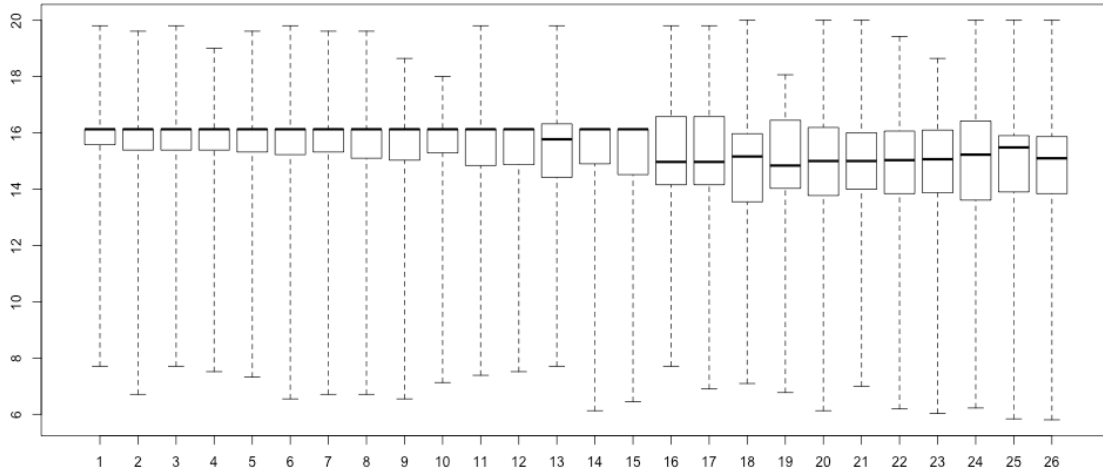


Figure 4.1: Boxplot of simulation results for baseline, full-feedback condition. x-axis corresponds to rank in decreasing order of average score per game, y-axis corresponds to total payoff per game.

In addition, a table showing average payoffs per game for each pair of strategies is included in the appendix.

Partial-feedback condition

When sorted by decreasing average payoff per game, the rankings under partial-feedback are as follows:

<i>Rank</i>	<i>Strategy</i>	<i>Average payoff</i>
1.	NeverInvest	15.38
2.	I-N	15.37
3.	NeverInvestAfter1Loss	15.33
4.	Random0.2	15.24
5.	Invest1AfterLoss	15.21
6.	NeverInvestAfter2Losses	15.12
7.	Invest2AfterLoss	15.09
8.	DoNotInvest1AfterLoss	15.09
9.	Alternate	15.08
10.	Random0.5	15.06
11.	DoNotInvest2AfterLoss	14.98
12.	Random0.7	14.97
13.	AlwaysInvestAfter2Losses	14.92
14.	AlwaysInvestAfter1Loss	14.86
15.	AlwaysInvest	14.85

Clearly, the partial-feedback condition makes a significant difference to the results, favoring strategies that rarely invest over those that frequently do so. The top two performing strategies under partial-feedback, NeverInvest and I-N, did not rank in the top 50% for the full-feedback simulation. The only strategy that performs relatively well under both full-feedback and partial-feedback conditions is NeverInvestAfter1Loss.

When we examine the distribution of game outcomes for each strategy using boxplot diagrams (Figure 4.2), we see a different story than in the full-feedback condition: there appears to be little correlation between payoff variance and performance. Also, there is surprisingly little correlation between median payoff and performance. This suggests that average payoff does not give us a representative summary of the performance of these different strategies; when ranked by median payoffs, or other statistical measures of performance, the ordering of the strategies would be quite different.

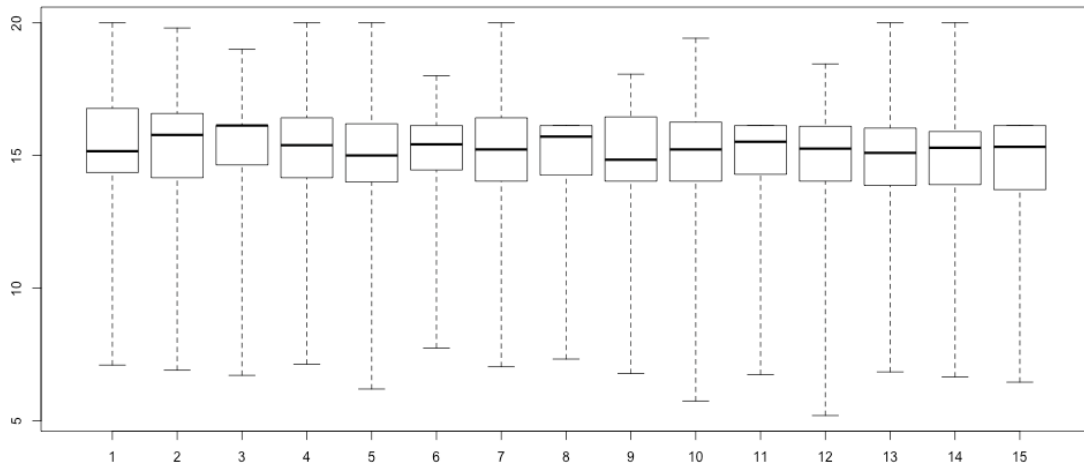


Figure 4.2: Boxplot of simulation results for baseline, partial-feedback condition. x-axis corresponds to rank in decreasing order of average score per game, y-axis corresponds to payoffs per game.

5. Risk profile transformation

The baseline simulations were conducted under an assumption of risk neutrality. In particular, the baseline simulations assume that that an agent's utility function, which maps from realized payoffs to utility, is $U(x) = (x-lo)/(hi-lo)$. The parameters hi and lo represent, respectively, the highest possible payoff in a round (zero) and the lowest possible payoff in a round (-62). In this section, we investigate how the results of the baseline simulation would differ under the assumptions of risk aversion and of risk proneness. As the payoffs in IDS games are always negative, we select risk averse and risk prone utility functions that are suitably defined over the domain $(\infty,0]$.

To illustrate risk aversion, we employ the utility function $U(x) = [(x-lo)/(hi-lo)]^c$ with $c = 1/2$. The value of c here is chosen arbitrarily as the midpoint of the range $(0,1)$, since any value in this range will result in a concave utility function defined over the negative domain. A plot of this function is labeled as A in Figure 5.1.

To illustrate risk proneness, we employ the utility function $U(x) = [(x-lo)/(hi-lo)]^c$ with $c = 3/2$. The value of c here is chosen from the low end of the range $(1, \infty)$, since any value in this range will result in a convex utility function defined in the negative domain. A plot of this function is labeled as C in Figure 5.1.

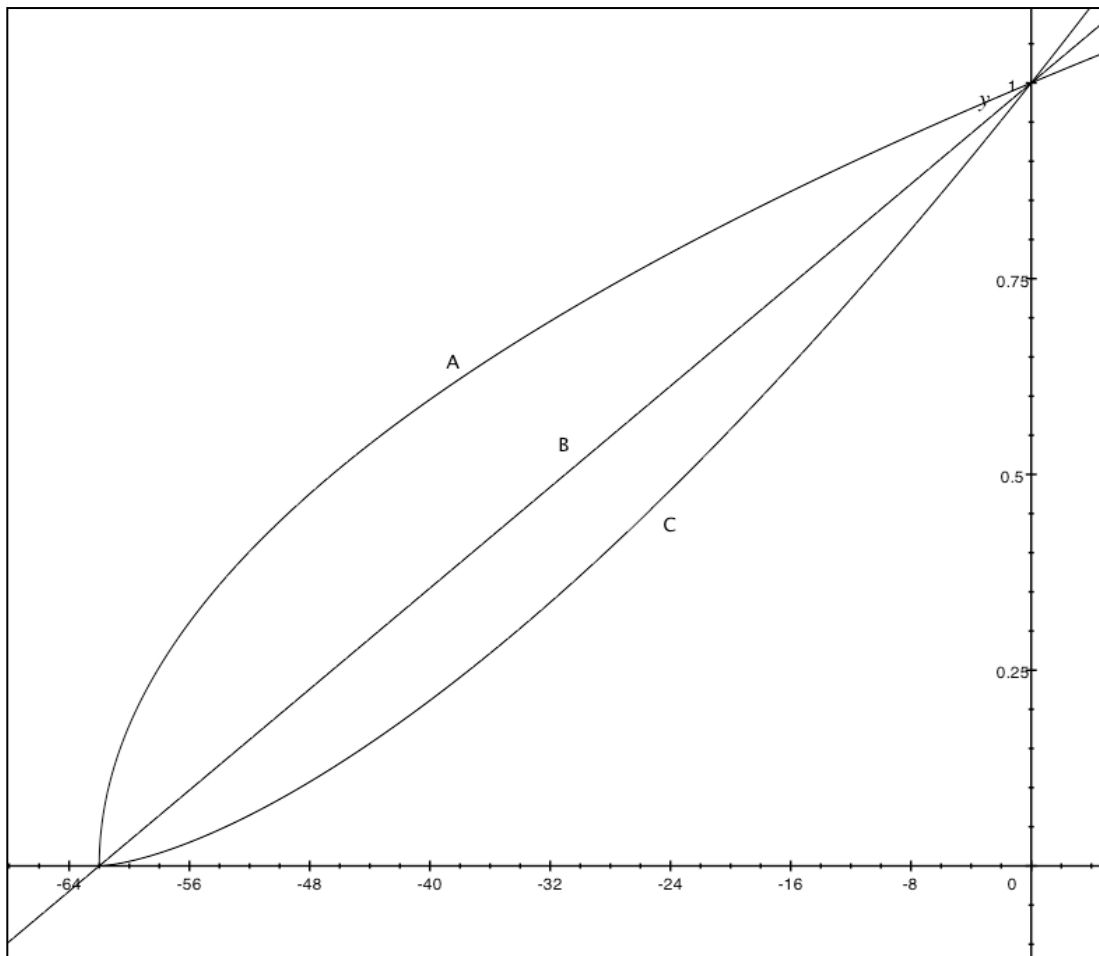


Figure 5.1: Illustrative utility functions. (A): $U(x) = [(x-lo)/(hi-lo)]^{1/2}$; (B) $U(x) = (x-lo)/(hi-lo)$; (C) $U(x) = [(x-lo)/(hi-lo)]^{3/2}$

5.1. Risk averse utility function

The complete rankings for the risk averse, full-feedback condition are as follows (the corresponding boxplot diagram is shown in Figure 5.2):

<i>Rank</i>	<i>Strategy</i>	<i>Average utility</i>
1.	GRIM2	17.34
2.	GRIM	17.33
3.	2TFT	17.32
4.	2TF2T	17.29
5.	TF2T	17.28
6.	NeverInvestAfter1Loss	17.27
7.	TFT	17.26
8.	2PTFT	17.22
9.	T2	17.18
10.	NeverInvestAfter2Losses	17.17
11.	DoNotInvest1AfterLoss	17.13
12.	PTFT	17.12
13.	DoNotInvest2AfterLoss	17.05
14.	DTFT	17.00
15.	AlwaysInvest	16.98
16.	FictitiousPlay	16.60
17.	I-N	16.59
18.	NeverInvest	16.57
19.	Alternate	16.51
20.	Random0.2	16.48
21.	Random0.7	16.46
22.	AlwaysInvestAfter1Loss	16.45
23.	Random0.5	16.42
24.	Invest1AfterLoss	16.41
25.	AlwaysInvestAfter2Losses	16.39
26.	Invest2AfterLoss	16.35

We see here that the results for the risk averse, full-feedback condition are broadly similar to those for the baseline, full-feedback condition despite slight differences in the ordering of strategies.

The complete rankings for the risk averse, partial-feedback condition are as follows (the corresponding boxplot diagram is show in Figure 5.3):

<i>Rank</i>	<i>Strategy</i>	<i>Average utility</i>
1.	NeverInvestAfter1Loss	16.97
2.	NeverInvestAfter2Losses	16.80

3.	NeverInvest	16.79
4.	I-N	16.79
5.	DoNotInvest1AfterLoss	16.78
6.	Random0.2	16.69
7.	DoNotInvest2AfterLoss	16.67
8.	Invest1AfterLoss	16.64
9.	Alternate	16.63
10.	Random0.5	16.57
11.	Invest2AfterLoss	16.54
12.	Random0.7	16.54
13.	AlwaysInvest	16.53
14.	AlwaysInvestAfter2Losses	16.47
15.	AlwaysInvestAfter1Loss	16.47

We see here that the results for the risk averse, partial-feedback condition differ substantially from those in the baseline, partial-feedback condition. The top performing strategy by a significant margin is now NeverInvestAfter1Loss—a nice strategy—rather than NeverInvest. NeverInvest is still among the top performers and AlwaysInvest is still among the worst, but, in general, it appears the assumption of risk aversion is more conducive to cooperation than that of risk neutrality. This is what one would expect given that a loss is viewed more negatively as a person becomes more risk averse.

5.2. Risk prone utility function

The complete rankings for the risk prone, full-feedback condition are as follows (the corresponding boxplot diagram is shown in Figure 5.4):

<i>Rank</i>	<i>Strategy</i>	<i>Average utility</i>
1.	GRIM	14.42
2.	NeverInvest	14.40
3.	FictitiousPlay	14.37
4.	I-N	14.37
5.	DTFT	14.36
6.	GRIM2	14.29
7.	2TFT	14.26
8.	2PTFT	14.21
9.	NeverInvestAfter1Loss	14.20
10.	Random0.2	14.17
11.	TFT	14.15
12.	Invest1AfterLoss	14.11
13.	2TF2T	14.11
14.	TF2T	14.07

15.	T2	14.05
16.	PTFT	14.02
17.	NeverInvestAfter2Losses	13.98
18.	Alternate	13.94
19.	Invest2AfterLoss	13.93
20.	DoNotInvest1AfterLoss	13.91
21.	Random0.5	13.85
22.	DoNotInvest2AfterLoss	13.80
23.	AlwaysInvest	13.70
24.	Random0.7	13.67
25.	AlwaysInvestAfter2Losses	13.63
26.	AlwaysInvestAfter1Loss	13.49

The above results differ significantly from those for the baseline, full-feedback condition. NeverInvest now ranks in 2nd place rather than in 18th place and, in general, hostile strategies tend to perform considerably better than in the baseline simulation. Again, this is what one would expect given that a loss is viewed less negatively as a person becomes more risk prone.

The complete rankings for the risk prone, partial-feedback condition are as follows (the corresponding boxplot diagram is show in Figure 5.5):

<i>Rank</i>	<i>Strategy</i>	<i>Average utility</i>
1.	NeverInvest	14.76
2.	I-N	14.70
3.	Random0.2	14.46
4.	Invest1AfterLoss	14.41
5.	Invest2AfterLoss	14.18
6.	NeverInvestAfter1Loss	14.12
7.	Alternate	14.02
8.	Random0.5	14.01
9.	NeverInvestAfter2Losses	13.77
10.	AlwaysInvestAfter2Losses	13.76
11.	Random0.7	13.74
12.	DoNotInvest1AfterLoss	13.68
13.	AlwaysInvestAfter1Loss	13.53
14.	DoNotInvest2AfterLoss	13.52
15.	AlwaysInvest	13.33

The results above are broadly similar those in the baseline, partial-feedback condition in that hostile strategies tend to outperform nice strategies. Indeed, nice strategies (such as NeverInvestAfter1Loss) tend to perform even worse under the assumption of risk proneness than under risk neutrality.

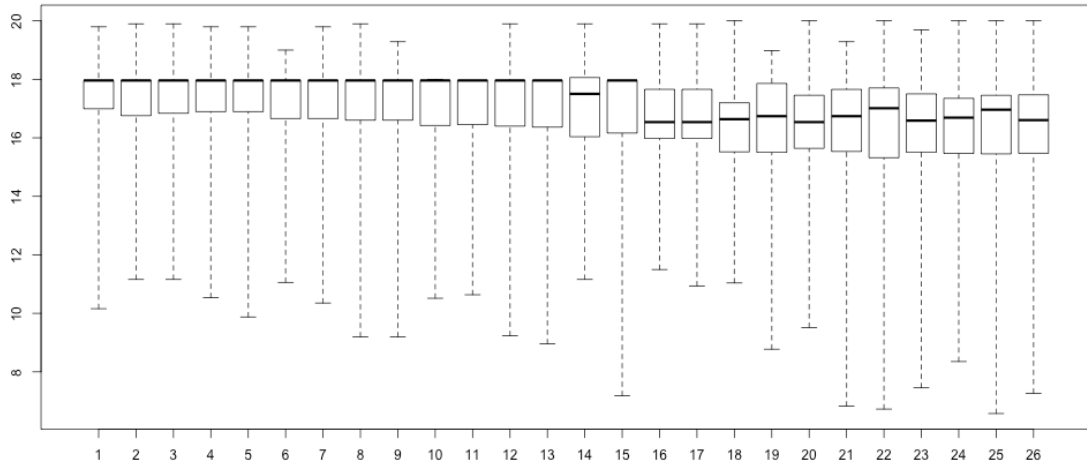


Figure 5.2: Boxplot of simulation results for risk averse, full-feedback condition. x-axis corresponds to rank in decreasing order of average utility per game, y-axis corresponds to average utility per game.

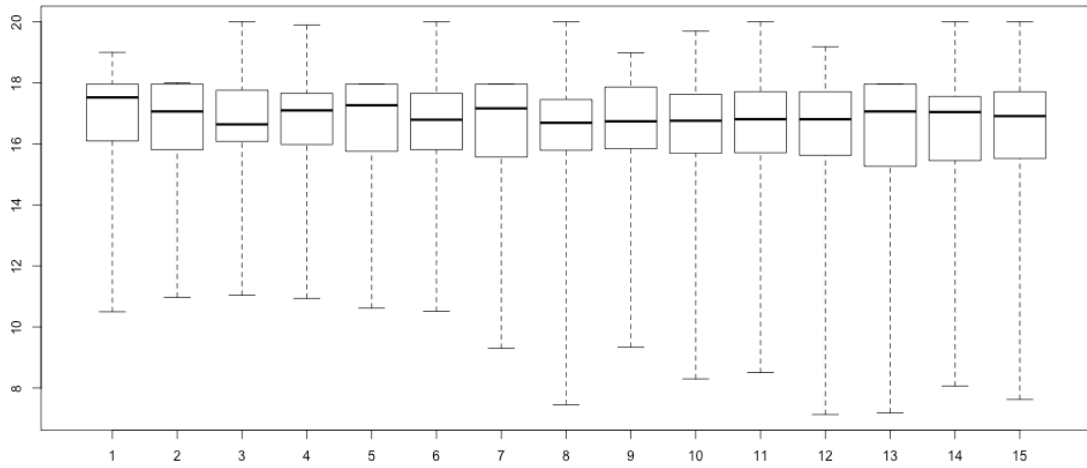


Figure 5.3: Boxplot of simulation results for risk averse, partial-feedback condition. x-axis corresponds to rank in decreasing order of average utility per game, y-axis corresponds to average utility per game.

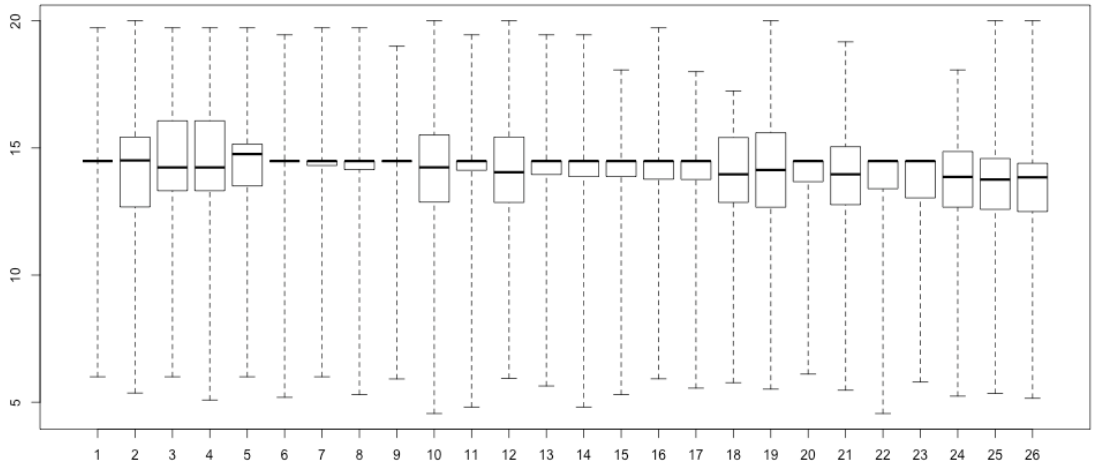


Figure 5.4: Boxplot of simulation results for risk prone, full-feedback condition. x-axis corresponds to rank in decreasing order of average utility per game, y-axis corresponds to average utility per game.

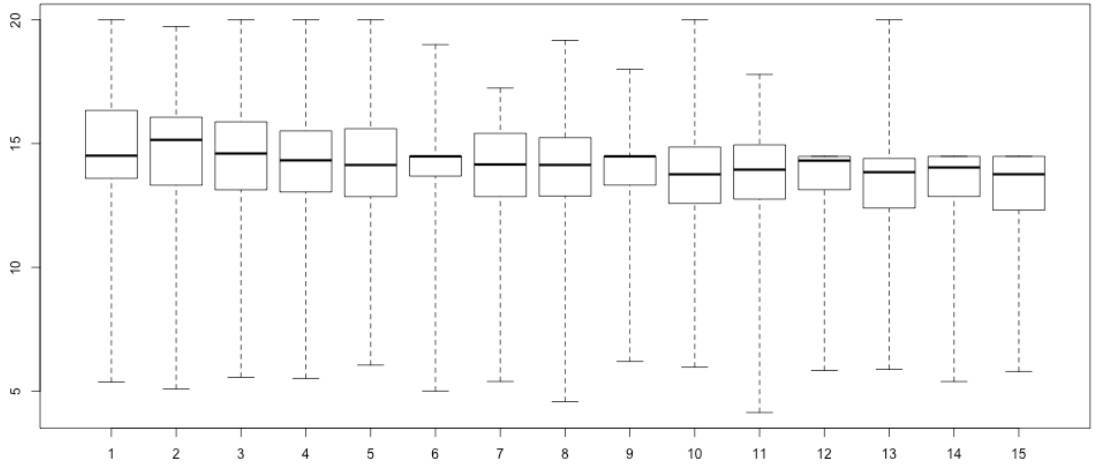


Figure 5.5: Boxplot of simulation results for risk prone, partial-feedback condition. x-axis corresponds to rank in decreasing order of average utility per game, y-axis corresponds to average utility per game.

6. Discussion

Some clear patterns emerge from the foregoing computer simulations. In the baseline full-feedback condition the top performing strategies tend to be nice, but difficult to exploit. This finding is broadly consistent with prior analyses of Prisoner's Dilemma

games (Axelrod, 1984). The results are substantially different in the baseline partial-feedback condition, where the top performing strategy is NeverInvest. What explains this change in result? There are two factors to consider. First, the change in information available to agents (from full to partial feedback) makes it more difficult for agents to discern partners that are nice from those that are not. This tends to favor strategies that are hostile. Second, as we shift from full to partial feedback, there is an accompanying change in the strategic landscape: there are fewer nice strategies among the 15 strategies included in the partial-feedback condition than among the 26 strategies in the full-feedback condition. This, too, tends to favor strategies that are hostile. In future work, it would be illuminating to vary the proportion of nice strategies in the partial-feedback condition to understand how this influences outcome.

The only strategy that performs relatively well in both the baseline full-feedback and baseline partial feedback conditions is NeverInvestAfter1Loss. The success of this strategy can be explained because it is one of the few strategies that is both nice and hard to exploit under both conditions (many of the strategies with these characteristics in the full-feedback condition are not practicable under partial feedback). The success of NeverInvestAfter1Loss is all the more notable because it outperforms some of the other nice but hard to exploit strategies in the full-feedback condition, such as TFT, even though it operates on less information.

The finding that partial-feedback favors strategies that rarely invest over those that frequently do so highlights the importance of providing feedback if we want individuals to invest. From a policy perspective, this suggests the need for monitoring (e.g., third party inspections and mechanisms for public accountability) to encourage investment. It also suggests that for very low probability events, strategies that do not invest may dominate those that do due to the more limited feedback.

Our investigation into the effect of different utility functions in section 5 shows that risk attitude has a large effect on the ranking of strategies. The effects are particularly pronounced in the partial feedback condition. Under the assumption of risk aversion the top two strategies (NeverInvestAfter1Loss and NeverInvestAfter2Losses) are both nice, whereas under the assumption of risk proneness the top five strategies are all hostile. Thus, in the partial feedback condition, risk attitude in itself appears sufficient to shift equilibrium play from hostile to nice: strongly risk averse agents will tend to invest, while strongly risk prone agents will not. In the full feedback condition, risk aversion has little effect on the ranking of strategies (i.e., the top performers are all nice) but risk proneness significantly improves the performance of hostile strategies (i.e., the top two performers are GRIM and NeverInvest).

In all, the simulation results suggest that risk attitude is at least as important a determinant of strategic choice as the different informational conditions. In the behavioral experiments conducted by Kunreuther et al. (2009) with the same parameters as those investigated in our simulations ($p=q=.2$, $c=12$, $L=50$) human subjects invested in security

only approximately 25% of the time. The simulation results here suggest that one explanation for why human subjects invest so rarely may be that many subjects are risk prone. Yet, more investigation is needed before we can use the results of our simulations to interpret behavior. For example, the ranking of strategies used in our simulation assumes that agents are interested in maximizing expected utility, but it is likely that individuals use choice rules other than this.

The simulations conducted so far cover only one parameterization of the 2-player IDS model. It is not yet clear to what extent the patterns that have emerged will generalize to other sets of parameters, to variations in the composition of strategic landscape, or to other changes in assumptions. In future work, it would be illuminating to extend the present analysis in several ways. First, it would be desirable to conduct a sensitivity analysis to understand how the results of the simulation would change as one varies the parameters and the composition of the strategic landscape. In particular, it would be interesting to explore parameterizations with low values of p and/or q to test the hypothesis that cooperation becomes more difficult to maintain as loss events become rarer. Second, it would be interesting to investigate a larger set of risk averse and risk prone utility functions (in particular, utility functions that are less extreme and more behaviorally plausible than those investigated here). On a related note, it would be interesting to examine choice rules other than maximizing expected utility. Third, it would be revealing to conduct DPD simulations with the strategies investigated here and to compare the results with those of the SPD simulations.

References

- Axelrod, Robert (1984). *The Evolution of Cooperation*. New York: Basic Books.
- Fudenberg, Drew, David Rand, Anna Dreber (2010). "Slow to Anger and Fast to Foregive: Cooperation in an Uncertain World". Unpublished manuscript.
- Gong, Min, Jonathan Baron, & Howard Kunreuther (2009), "Group Cooperation under Uncertainty", *Journal of Risk and Uncertainty*, 39:251-270.
- Hess, Rachel, Charles Holt & Angela Smith (2007). "Coordination of Strategic Responses to Security Threats: Laboratory Evidence." *Experimental Economics*, 10: 235-250.
- Kunreuther, Howard and Geoffrey Heal (2003). "Interdependent Security". *Journal of Risk and Uncertainty*, 26: 231-249.
- Howard Kunreuther, Gabriel Silvasi, Eric Bradlow, Dylan Small (2009), "Bayesian Analysis of Deterministic and Stochastic Prisoner's Dilemma Games", *Judgment and Decision Making*, Vol. 4, No. 5, August 2009, pp. 363–384.
- Shafran, Aric (2010). "Interdependent Security Experiments", *Economics Bulletin*, 2010, 30(3), pp. 1950-62.

Appendix

Baseline (risk-neutral), full-feedback pairwise matrix

The following matrix displays the average game outcome for Row when paired with Column.

Baseline (risk-neutral), partial-feedback pairwise matrix

The following matrix displays the average game outcome for Row when paired with Column.

	AlwaysInvestAfter1Loss	AlwaysInvestAfter2Losses	NeverInvestAfter1Loss	NeverInvestAfter2Losses	Invest1AfterLoss	Invest2AfterLoss	DoNotInvest1AfterLoss	DoNotInvest2AfterLoss	AlwaysInvest	NeverInvest	I-N	Alternate	Random0.2	Random0.5	Random0.7
AlwaysInvestAfter1Loss	15.72	15.03	14.73	15.72	13.66	14.07	15.95	16.12	16.30	13.07	13.24	14.70	13.73	14.69	15.34
AlwaysInvestAfter2Losses	15.96	15.38	14.35	15.32	13.87	14.36	15.86	16.10	16.44	13.26	13.38	14.88	13.91	14.85	15.48
NeverInvestAfter1Loss	15.84	15.44	16.13	16.13	14.50	14.89	16.13	16.13	16.13	13.83	13.97	15.04	14.37	15.07	15.52
NeverInvestAfter2Losses	15.54	15.06	16.13	16.13	14.12	14.45	16.13	16.13	16.13	13.59	13.65	14.74	14.02	14.77	15.25
Invest1AfterLoss	16.25	15.81	14.59	15.39	14.59	15.05	15.75	16.16	16.70	13.79	13.92	15.28	14.43	15.27	15.81
Invest2AfterLoss	16.12	15.57	14.53	15.41	14.34	14.82	15.76	16.12	16.58	13.59	13.74	15.11	14.20	15.11	15.70
DoNotInvest1AfterLoss	15.57	15.08	16.13	16.13	13.98	14.26	16.13	16.13	16.13	13.46	13.57	14.73	13.92	14.73	15.30
DoNotInvest2AfterLoss	15.48	14.90	16.13	16.13	13.75	14.04	16.13	16.13	16.13	13.20	13.34	14.61	13.75	14.62	15.21
AlwaysInvest	15.34	14.56	16.13	16.13	13.41	13.78	16.13	16.13	16.13	12.90	13.06	14.51	13.54	14.53	15.16
NeverInvest	16.43	16.05	14.80	15.46	14.93	15.38	15.74	16.24	16.75	14.18	14.30	15.47	14.70	15.46	15.98
I-N	16.34	15.95	14.93	15.53	14.92	15.33	15.76	16.18	16.76	14.14	14.28	15.45	14.65	15.41	15.98
Alternate	15.87	15.38	14.80	15.77	14.22	14.66	15.77	16.11	16.46	13.53	13.70	15.15	14.15	15.03	15.61
Random0.2	16.24	15.83	14.72	15.46	14.63	15.14	15.75	16.19	16.68	13.93	14.02	15.29	14.45	15.28	15.83
Random0.5	15.91	15.44	14.82	15.73	14.21	14.63	15.77	16.11	16.42	13.54	13.68	15.02	14.12	14.96	15.58
Random0.7	15.72	15.10	15.11	15.92	13.91	14.33	15.90	16.10	16.32	13.28	13.46	14.80	13.88	14.79	15.41

Risk-averse, full-feedback results

The following matrix displays the average game outcome for Row when paired with Column.

Risk-averse, partial-feedback pairwise matrix

The following matrix displays the average game outcome for Row when paired with Column.

	AlwaysInvestAfter1Loss	AlwaysInvestAfter2Losses	NeverInvestAfter1Loss	NeverInvestAfter2Losses	Invest1AfterLoss	Invest2AfterLoss	DoNotInvest1AfterLoss	DoNotInvest2AfterLoss	AlwaysInvest	NeverInvest	I-N	Alternate	Random0.2	Random0.5	Random0.7
AlwaysInvestAfter1Loss	17.43	16.72	16.28	17.33	15.20	15.63	17.63	17.79	17.92	14.57	14.71	16.29	15.28	16.28	16.96
AlwaysInvestAfter2Losses	17.54	17.04	15.82	16.77	15.42	15.90	17.44	17.65	17.86	14.81	14.90	16.42	15.47	16.41	17.00
NeverInvestAfter1Loss	17.26	16.79	17.96	17.96	16.04	16.33	17.96	17.96	17.96	15.54	15.67	16.54	15.96	16.56	17.03
NeverInvestAfter2Losses	17.17	16.55	17.96	17.96	15.65	15.94	17.96	17.96	17.96	15.22	15.29	16.32	15.60	16.35	16.91
Invest1AfterLoss	17.47	17.11	16.12	16.76	16.17	16.52	17.06	17.40	17.81	15.48	15.58	16.77	16.02	16.72	17.14
Invest2AfterLoss	17.43	16.95	16.00	16.78	15.90	16.34	17.11	17.44	17.80	15.21	15.33	16.58	15.79	16.59	17.09
DoNotInvest1AfterLoss	17.30	16.76	17.96	17.96	15.51	15.80	17.96	17.96	17.96	15.04	15.17	16.31	15.49	16.32	16.95
DoNotInvest2AfterLoss	17.23	16.59	17.96	17.96	15.28	15.62	17.96	17.96	17.96	14.74	14.89	16.18	15.31	16.24	16.90
AlwaysInvest	17.08	16.21	17.96	17.96	14.93	15.35	17.96	17.96	17.96	14.37	14.55	16.16	15.07	16.18	16.88
NeverInvest	17.52	17.26	16.39	16.85	16.48	16.79	17.04	17.39	17.74	15.96	16.04	16.85	16.32	16.84	17.21
I-N	17.42	17.15	16.51	16.93	16.44	16.72	17.09	17.38	17.78	15.89	16.06	16.87	16.26	16.81	17.22
Alternate	17.32	16.86	16.34	17.23	15.83	16.18	17.22	17.49	17.87	15.14	15.34	16.96	15.72	16.54	17.07
Random0.2	17.48	17.15	16.28	16.86	16.21	16.60	17.09	17.43	17.82	15.64	15.71	16.73	16.06	16.71	17.15
Random0.5	17.35	16.92	16.34	17.18	15.79	16.17	17.23	17.55	17.84	15.17	15.29	16.52	15.70	16.48	17.05
Random0.7	17.29	16.66	16.68	17.50	15.48	15.89	17.48	17.68	17.90	14.85	15.03	16.37	15.44	16.36	16.98

Risk-prone, full-feedback pairwise matrix

The following matrix displays the average game outcome for Row when paired with Column.

Risk-prone, Partial-feedback pairwise matrix

The following matrix displays the average game outcome for Row when paired with Column.

	AlwaysInvestAfter1Loss	AlwaysInvestAfter2Losses	NeverInvestAfter1Loss	NeverInvestAfter2Losses	Invest1AfterLoss	Invest2AfterLoss	DoNotInvest1AfterLoss	DoNotInvest2AfterLoss	AlwaysInvest	NeverInvest	I-N	Alternate	Random0.2	Random0.5	Random0.7
AlwaysInvestAfter1Loss	18.21	17.91	18.47	18.70	17.78	17.83	18.69	18.74	18.76	17.70	17.76	18.27	17.91	18.24	18.46
AlwaysInvestAfter2Losses	18.46	18.14	18.62	18.85	17.96	18.07	18.87	18.92	18.94	17.91	17.97	18.38	18.12	18.43	18.64
NeverInvestAfter1Loss	17.95	17.75	18.33	18.33	17.68	17.76	18.33	18.33	18.33	17.65	17.68	18.00	17.79	18.02	18.14
NeverInvestAfter2Losses	17.79	17.56	18.33	18.33	17.50	17.56	18.33	18.33	18.33	17.47	17.50	17.88	17.63	17.90	18.05
Invest1AfterLoss	18.52	18.22	18.62	18.91	18.09	18.18	18.88	18.94	19.00	18.00	18.09	18.47	18.21	18.47	18.69
Invest2AfterLoss	18.48	18.20	18.62	18.87	18.03	18.09	18.84	18.92	18.94	17.96	18.01	18.50	18.16	18.48	18.67
DoNotInvest1AfterLoss	17.79	17.56	18.33	18.33	17.52	17.55	18.33	18.33	18.33	17.46	17.51	17.91	17.64	17.89	18.07
DoNotInvest2AfterLoss	17.76	17.54	18.33	18.33	17.48	17.53	18.33	18.33	18.33	17.44	17.50	17.88	17.62	17.88	18.06
AlwaysInvest	17.74	17.50	18.33	18.33	17.44	17.50	18.33	18.33	18.33	17.40	17.41	17.86	17.61	17.87	18.05
NeverInvest	18.58	18.31	18.69	18.93	18.16	18.19	18.90	18.99	19.01	18.08	18.14	18.54	18.25	18.54	18.74
I-N	18.55	18.25	18.67	18.92	18.14	18.20	18.88	18.94	18.97	18.03	18.12	18.52	18.23	18.52	18.72
Alternate	18.17	17.90	18.50	18.65	17.80	17.85	18.63	18.64	18.68	17.75	17.79	18.20	17.93	18.21	18.40
Random0.2	18.45	18.12	18.56	18.81	18.01	18.08	18.82	18.82	18.89	17.93	17.96	18.39	18.11	18.41	18.62
Random0.5	18.18	17.91	18.48	18.64	17.80	17.86	18.64	18.64	18.66	17.75	17.80	18.21	17.91	18.21	18.40
Random0.7	18.00	17.73	18.40	18.50	17.70	17.70	18.51	18.51	18.54	17.61	17.66	18.07	17.79	18.07	18.25

Specification of FictitiousPlay

The following code in the Python programming language implements FictitiousPlay as a class:

```
class FictitiousPlay(object):

    def __init__(self):
        self.history = []

    def decide(self, partner_invested, c, p, q, L):
        """Decision function called in each round. Accepts parameters
        corresponding to whether the partner has invested in the
        previous round, as well as the values of c, p, q, L from
        the IDS model. Returns True or False depending on whether
        agent will invest in this round."""
        # Return true in first round
        if len(self.history) == 0:
            return True
        # Keep track of history from prior rounds
        self.history.append(partner_invested)
        # compute percentage of times partner has previously invested
        a = average(self.history)
        # Return True iff this maximizes expected payoff in next round
```

```
return a*(-1*c) + (1-a)*((-1*c)+(q*(-1*L))) >= \  
a*(p*(-1*L))+(1-a)*((p*(-1*L))+((1-p)*q*(-1*L)))
```