From Incan Gold to Dominion:

Unraveling Optimal Strategies in Unsolved Games

A Senior Comprehensive Paper

presented to
the Faculty of Carleton College
Department of Mathematics
Tommy Occhipinti, Advisor

by
Grace Jaffe Tyler Mahony Lucinda Robinson

March 5, 2014

Acknowledgments

We would like to thank our advisor, Tommy Occhipinti, for constant technical help, brainstorming, and moral support. We would also like to thank Tristan Occhipinti for providing the software that allowed us to run our simulations, Miles Ott for helping us with statistics, Mike Tie for continual technical support, Andrew Gainer-Dewar for providing the LaTeX templates, and Harold Jaffe for proofreading. Finally, we would like to express our gratitude to the entire math department for everything they have done for us over the past four years.

Abstract

While some games have inherent optimal strategies, strategies that will win no matter how the opponent plays, perhaps more interesting are those that do not possess an objective best strategy. In this paper, we examine three such games: the iterated prisoner's dilemma, Incan Gold, and Dominion. Through computer simulations, we attempted to develop strategies for each game that could win most of the time, though not necessarily all of the time. We strived to create strategies that had a large breadth of success; that is, facing most common strategies of each game, ours would emerge on top. We begin with an analysis of the iterated prisoner's dilemma, running an Axelrod-style tournament to determine the balance of characteristics of winning strategies. Next, we turn our attention to Incan Gold, where we examine the ramifications of different styles of decision making, hoping to enhance an already powerful strategy. Finally, we unpack Dominion, attempting to uncover the most potent combination of cards that can lead to almost universal success.

Contents

1. Introduction

Tic-tac-toe is a solved game: it is always possible to prevent an opponent from winning. If one's opponent makes a mistake, it is possible to win the game, but even if one's opponent plays perfectly, one can always force a draw. In one sense, then, the opponent's decisions do not matter, because the outcome is always the same. On the other hand, those decisions matter very much, because they dictate the decisions the primary player must make to achieve that outcome. Our senior comprehensive project is a study of three unsolved games, in which the existence of perfect play is unproven and, for our games, ill-defined. The games (iterated prisoner's dilemma, Incan Gold, and Dominion) all require the player to make his own decisions based on the ones his opponents are making, but unlike tic-tac-toe, his responses will not necessarily force a particular outcome. There is no winning strategy for any of our games, nor are there strategies that can never lose; our goal was then to find good strategies. For each of our games, we coded multiple strategies, played them against each other, and adjusted the most successful ones in an effort to augment that success. The result was the development of strategies that are not provably best, but perform respectably well against a range of commonly encountered opponents.

2. Iterated Prisoner's Dilemma

The first game we studied was the iterated prisoner's dilemma. Our goal in studying this game was, given a list of twenty common strategies (see Appendix A), to develop a new strategy that performed better than any of those on the list. Our metric for success of a strategy was its cumulative score when played against each of the twenty common strategies, rather than the percentage of times it won individual games.

2.1 Introduction to Regular Prisoner's Dilemma

Iterated prisoner's dilemma is a variation on regular prisoner's dilemma. In the regular version, player A and player B are partners in crime and have been caught by the police. The police have enough evidence to charge the players on some, but not all, of their crimes. They place player A in one room and player B in another so that the two cannot communicate with each other. Player A must decide if he is going to cooperate with his partner (and not tell the police anything) or defect against his partner (and tell the police everything). At the same time, player B decides if she is going to cooperate or defect. If both players cooperate, the police will not have enough evidence and each player receives 1 year of jail time. If player A defects and player B cooperates, player A will go free for confessing and the police will use his confession to incarcerate player B for 5 years. If player A cooperates and player B defects, player A gets 5 years while player B goes free, and if both players defect, both receive 3 years. This information is commonly condensed into a payoff matrix, where the entry (X, Y) corresponds to (Player A's jail time, Player B's jail time):

	Player A Cooperates	Player A Defects
Player B Cooperates	$(1, 1)$	$(0, 5)$
Player B Defects	$(5, 0)$	$(3, 3)$

Table 2.1: Jail Sentences

As one might expect, the goal of prisoner's dilemma for an individual player is to serve the least possible amount of jail time, and it can be proven that the optimal move is always to defect.

Proposition 2.1.1. *It is always in a player's best interest, in terms of time they would spend in jail, to defect.*

Proof. Without loss of generality, consider the position of player A. Player B is either going to cooperate or defect. If player B is going to cooperate, player A will receive 1 year of jail time if he cooperates and no jail time if he defects, so player A receives less jail time by defecting. Then if player B cooperates, the best move for player A is to defect. If, on the other hand, player B is going to defect, player A will receive 5 years of jail time if he cooperates and 3 years if he defects, so player A receives less jail time by defecting. Then if player B defects, the best move for player A is to defect. Thus whatever player B does, the best move for player A is to defect. Similarly, whatever player A does, the best move for player B is to defect. □

If both players behave in a manner befitting their individual best interests, both players will defect and each will receive 3 years' jail time. In this situation, both will be worse off than if they had both cooperated, which would have resulted in each serving a lower sentence of 1 year. By Proposition 2.1.1, however, cooperating is in neither player's best interest. This paradox is the defining characteristic of the prisoner's dilemma.

The jail sentences are commonly redefined as generic rewards, so that instead of trying to obtain the least jail time, players try to obtain the highest score. The payoff matrix now looks like this, with (X, Y) now denoting (Player A's reward, Player B's reward):

	Player A Cooperates	Player A Defects
Player B Cooperates	$(3, 3)$	$(5, 0)$
Player B Defects	$(0, 5)$	$(1, 1)$

Table 2.2: Rewards

As in the original scenario, player A receives a higher score, no matter what player B does, by defecting, and player B receives a higher score, no matter what player A does, by defecting, too. But if both players defect, they both receive a lower score than if they had both cooperated, so we are in an analogous situation. This flipped version is more commonly used in analyzing strategies during iterated prisoner's dilemma, so that a higher score corresponds to a better strategy.

The above payoff matrix was the one we used throughout our study of prisoner's dilemma and iterated prisoner's dilemma, but it is possible to change the reward values as long as certain inequalities are upheld. The rewards are denoted as follows:

- R is the reward both players receive by both cooperating (here, $R = 3$)
- T is the reward player A receives by defecting when player B cooperates/that player B receives by defecting when player A cooperates (here, $T = 5$)
- S is the reward player A receives by cooperating when player B defects/that player B receives by cooperating when player A defects (here, $S = 0$)
- P is the reward both players receive by both defecting (here, $P = 1$)

In order for Proposition 2.1.1 to apply, we need $T > R$, so that the reward a player gets for defecting while his partner cooperates is greater than his reward for cooperating, too. We also need $R > P$, so that both players cooperating is a better outcome for each player than both players defecting. Lastly, we need $P > S$, so that if a player defects, it is better for her partner to also defect. Then whenever $T > R > P > S$, we are in a true prisoner's dilemma scenario, where Proposition 2.1.1 applies.

2.2 Introduction to Iterated Prisoner's Dilemma

The iterated prisoner's dilemma (IPD) consists of multiple rounds of the regular prisoner's dilemma, and the scores are calculated cumulatively. This gives the players the opportunity to learn about each other's past behavior and accordingly adjust their present decisions. The main difference between iterated prisoner's dilemma and regular prisoner's dilemma is the introduction of consequences. When a single round of prisoner's dilemma is played, players make their decisions, receive their scores, and are finished. In iterated prisoner's dilemma, players start in the same fashion: in the first round, they make their decisions and receive the scores for that round. Going into the next round, however, a player now knows what her partner did in the last one. If her partner defected, she may choose to punish him by defecting in this round, or to show forgiveness by cooperating. If her partner cooperated, she may choose to reward his loyalty by cooperating in this round, or try to take advantage of him by defecting. Every round now has consequences in later rounds. The knowledge of how players behaved in the previous rounds now serves as a tool for each player to predict their opponent's next move.

An interesting facet of the iterated version is that when there are a fixed number of rounds (or if the number of rounds has a fixed upper bound), it is possible to prove that the only reasonable move is to defect in each round, given rational players:

Proposition 2.2.1. *Assuming that there is a known upper bound on the number of rounds, that the primary player's opponent is playing in the most rational way, and that that opponent assumes the primary player is playing the most rational way, the optimal strategy is to always defect.*

Proof. Suppose players A and B are going to play iterated prisoner's dilemma with no more than n rounds. On that n^{th} game, the only reasonable choice for both players is to defect, because it is just a case of regular prisoner's dilemma, and there are no consequences down the line for their behavior, since this is the last round. This means that in the $(n-1)^{th}$ round, the players' actions still do not have consequences in later rounds, because they are already both going to defect in the last round. Once again, this is an isolated case of the regular prisoner's dilemma and thus by Proposition 2.1.1 the best strategy is to defect. The proof works by reverse strong induction: since players A and players B both defect in rounds k through n, then both should defect in round $k-1$. Then the optimal strategy for a player is to always defect. Note that this remains true even if the game lasts less than n rounds, as this proof shows that in each round 1 through n, the optimal strategy is to defect. □

Despite this result, always defecting turns out to be a poor strategy in a situation in which there are opponents that do not adhere to the principles outlined in Proposition 2.2.1. That is, if any given player is playing against a strategy that does not always defect, the optimal strategy is not necessarily to always defect. For example, if two players always cooperate with each other, they will each obtain higher cumulative scores than they do when they always defect against each other. In general, strategies that have the potential to cooperate more often—such as Tit for Tat, which, each round, either rewards its opponent for cooperating in the previous round by now cooperating, or punishes its opponent for defecting in the previous round by now defecting—tend to do better in tournament set-ups that include strategies that do not always defect than do strategies that adhere to the logic of Proposition 2.2.1.

A specific tournament set-up is often used to determine the overall efficacy of IPD strategies. The round robin tournament pits each strategy against every other strategy a certain number of times and then aggregates the results. Each tournament is generally run multiple times, after which scores are averaged, to mitigate the variability of random strategies. Robert Axelrod constructed the first IPD tournament in 1980 using this format, hoping to study the evolution of cooperation and the effect of reciprocity. Game theorists were invited to submit computer programs that played IPD using various strategies that followed the basic rules of the game. In this competition, as in our analysis, winners were determined based on the cumulative number of points achieved. The Tit for Tat strategy achieved the highest score in both this initial competition and in a subsequent competition [AD88].

Although most competitors entered single strategies to this competition designed to be objectively successful against most other strategies, another approach to these tournaments has been for a team to submit a large number of easily identifiable strategies. In addition, they submit one strategy that not only does generally well against unknown strategies but also can recognize the other strategies submitted by the same team. Once the two strategies have identified each other, the strong strategy commences to take optimal advantage of the other weaker strategies by constantly defecting while they constantly cooperate.

In iterated prisoner's dilemma, we want to maintain the paradox of regular prisoner's dilemma. Here, the equivalent is that constant mutual cooperation should be the most mutually beneficial strategy. In order to ensure this, we need another constraint on the reward values in addition to the previous ones: $2R > S + T$, or $R > (S+T)/2$. This guarantees that the score two players receive from always cooperating is better than the score that they receive from alternating between cooperating and defecting [PD12].

Proposition 2.2.2. *If $2R > S + T$, then mutual cooperation produces higher scores for each player than alternating between cooperating and defecting.*

Proof. Suppose players A and B are playing iterated prisoner's dilemma with 100 rounds and that $R \leq (T+S)/2$. If both players always cooperate, both players obtain a score of R points per round times 100 rounds, or 100R points. If in even-numbered rounds player A cooperates while player B defects, and in odd-numbered rounds player A defects while player B cooperates, then player A scores S points per round in the 50 even rounds and T points per round in the 50 odd ones, so player A scores $50S + 50T$ points. Similarly, player B scores T points per round in the 50 even rounds and S points per round in the 50 odd ones, so player B scores $50T + 50S = 50S + 50T$ points as well. Since we have assumed $R \leq (T+S)/2$, it follows that $100R \leq 100((T+S)/2) = 50T + 50S$, so both players score at least as many points using this alternating pattern of decisions as they do by always cooperating. Thus we insist on $2R > S + T$. □

To clarify this, with our standard numbers, if player A and player B mutually cooperate, then mutually defect, each player's score would accumulate to 4 over two rounds (3 from the mutually cooperate round plus 1 from the mutually defect round), giving each player an average score of 2 for each individual round. If, on the other hand, player A cooperates while player B defects, and in the following round each player makes the opposite decision, each player's score will accumulate to 5 over two rounds (5 from defecting when the other cooperates plus 0 from cooperating when the other defects), giving each player an average score of 2.5 per round. Finally, if both players mutually cooperate over the two rounds, each player will score 3 points each round; thus, it is more beneficial to mutually cooperate than alternate cooperating and defecting.

So far, four strategies for iterated prisoner's dilemma have been mentioned: always defecting, always cooperating, alternating in each round, and Tit for Tat. These represent only a fraction of possible strategies. Our goal in studying IPD was to find a strategy more successful than each strategy from a specific list of opponents. As in Axelrod's tournament, we coded each strategy and ran computer simulations of multi-game round robin tournaments. The following is an example of pseudocode for the alternating strategy:

```
• If it is the first round of play, return either "cooperate" or "defect" with equal probability
• If it is not the first round of play, return whatever was not returned in the previous
  round (if "cooperate" was returned in the last round, return "defect;" if "defect" was
  returned in the last round, return "cooperate")
```

It should be noted that many of the strategies in the list of opponents are not successful, so our optimal strategy must be able to take advantage of that.

2.3 Measuring a Strategy's Success

A successful strategy is one that is able to score a relatively high number of points against a majority of the other strategies in the tournament. We started with a list of twenty common strategies: for example, one strategy on our list cooperates and defects at random in every round. Such a strategy does not normally score well in tournaments, but is often included as a player anyway, since it provides a valuable benchmark: if another strategy does no better than this random strategy, it is a clear indicator that the former strategy is a poor one. See Appendix A for the list of strategies.

Strategies can be thought of as possessing certain characteristics. For example, the Tit for Tat strategy is trusting because it never defects before an opponent does, and the Spiteful strategy is unforgiving because it punishes a single defect in any round with constant defects for the remainder of the game. One of the keys to having a successful strategy is for it to not possess too much of a given characteristic [AD88] [Don86]. For example, Always Cooperate loses points by being too trusting and Always Defect loses points by not being trusting at all, while the Tit for Tat strategy, which won one of the original IPD tournaments, shows both elements of trust and punishment. In order to determine which of these strategies were successful, we first sorted them into those that took on parameters and those that did not. For strategies that take parameters (for example, Soft Spite which cooperates until an opponent defects, then defects N times in a row), we ran each of them against all of the strategies that did not involve parameters, using a spectrum of different parameter values. This allowed us to determine what parameters afforded each strategy the best score. We then fixed the parameters so that each strategy would score as many points as possible against the gamut of others. However, in the case of some strategies that take parameters and behave similarly to another strategy on the list that does not take a parameter, we fixed an arbitrary parameter value, as the optimal parameter value rendered the strategy identical to another in the list. For example, Random Pavlov behaves like Pavlov but makes a non-Pavlovian decision with probability p, and performs best when $p = 0$. But when $p = 0$, Random Pavlov and Pavlov are the same strategy, so we fixed p at 20% instead, 20% being high enough to distinguish Random Pavlov from Pavlov, but low enough for Random Pavlov to still perform decently. Additionally, we needed p significantly less than 50%, which would just be the Random strategy. Once our list of twenty strategies was set, we were able to play them against each other and determine how well each strategy was doing.

2.4 Specific Strategies and their Relative Success

In our investigation of the iterated prisoner's dilemma, we used a round robin tournament set-up, similar to the set-up mentioned earlier, to determine the most successful strategy in our pool. Each game lasted 1,000 rounds, and each game was played 5 times and the scores were averaged to account for variability amongst

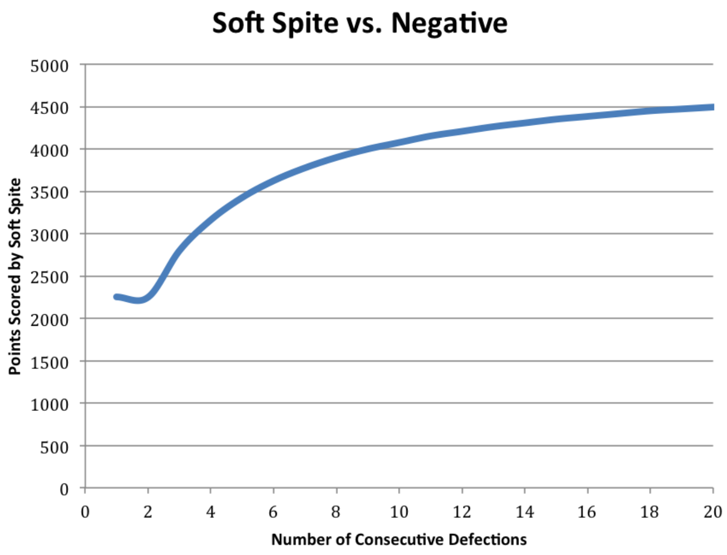


Figure 2.1: The spike in score as N increases reflects the increasing ability of Soft Spite to capitalize upon the combination of Soft Spite defecting and Negative cooperating that can continue for longer if N is larger, though the relative advantage of a larger N diminishes as N increases.

random strategies. With the scoring system we used, which is outlined above in Table 2.2, the theoretical maximum number of points any single strategy could obtain was 100,000, given that the maximum per game was 5,000, and each strategy was pitted against twenty other strategies. This maximum represents a round of games in which one player defects every time while each other player cooperates every time. In reality, the highest achievable aggregate score proved to be much lower (see Appendix B). This is because if the opposing strategy ever defects the maximum score is no longer 5,000; in our list of twenty there is only one that necessarily never defects. Thus 5,000 is rarely achievable. From our pool of strategies, the most effective strategies generally had an element of forgiveness, as well as the possibility of punishing strategies that are themselves too forgiving. That is, these strategies allowed for a return to cooperating, even if the opponent had defected during the game, but were also able, to a certain extent, to take advantage of those strategies that blindly cooperate.

After testing each strategy using our tournament set-up, the Gradual strategy obtained the highest score out of all strategies. This strategy is able to not only punish the other player for defecting, but also to exit a pattern of mutual defections (that other, less-forgiving strategies may get caught in against Tit for Tat, for example) by allowing two cooperates following a string of defects. This strategy not only allows for cooperation with more forgiving strategies, but it also cannot be taken advantage of by the more punishing strategies, and will essentially “salvage” a respectable number of points against those strategies which seek to exploit, rather than cooperate with, their opponents.

While Gradual obtained the highest score from the pool of twenty, it does not take a parameter and thus is difficult to tweak to obtain a more successful strategy. In our analysis of the optimal strategies, therefore, we focused primarily on the strategies whose behavior depends on a parameter. For these strategies, multiple runs were executed in order to find the optimal parameter for each strategy against the pool as a whole. These parameters were then used in the analysis of individual strategies. Three in particular obtain scores that are comparable with Gradual’s score against our list of twenty strategies. The next portion of this paper will be devoted to analyzing these three strategies. The first such strategy is the Soft Spite strategy, which cooperates until the opponent defects, then defects N times in a row.

Fig. 2.1 and Fig. 2.2 show the scores that Soft Spite obtains as the number of consecutive defects increases against Negative and Adaptive ($N = 8$).

Peaceful Tit for Tat, which behaves similarly to normal Tit for Tat but cooperates when Tit for Tat would defect with probability p, also achieved a similar score to Gradual. Fig. 2.3 and Fig. 2.4 display Peaceful Tit

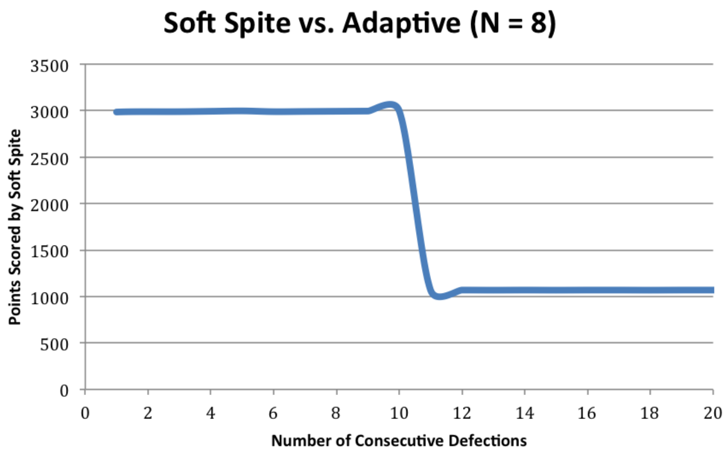


Figure 2.2: Soft Spite also does fairly well against Adaptive, set with the optimal parameter $N = 8$, until the number of defects that Soft Spite employs increases above 10. At that point, it will become more beneficial for Adaptive to defect every round, and the average score for each player in each round will be 1.

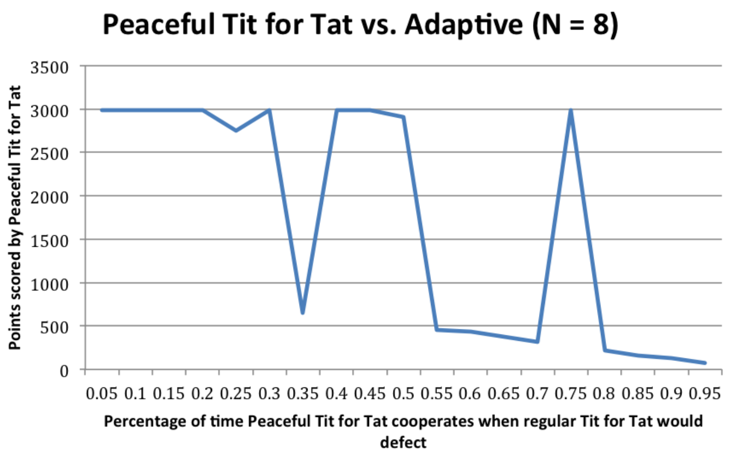


Figure 2.3: When Peaceful Tit for Tat closely resembles Tit for Tat, it does fairly well against Adaptive. The general decrease in score reflects the similarity to the Always Cooperate strategy, which does very poorly against Adaptive. We believe the spikes in the graph are due to the inherent randomness of Peaceful Tit for Tat.

for Tat's performance against Adaptive ($N = 8$) and Negative.

2.5 Analyzing Adaptive

The strategy that seemed most interesting and perhaps most easily improved upon was Adaptive. This strategy starts out with $N + 1$ cooperates followed by N defects. For its next move, it calculates its average score when choosing cooperate and its average score when choosing defect. It then compares the two averages and makes the choice that has gotten it a higher average score in the past. This process is repeated for the rest of the game. Adaptive relies on a parameter N, and while it performed decently with a wide range of N values, it was clearly most successful with a few very specific values for N. To determine the optimal value of N, we ran several simulations in which Adaptive played against all other strategies, varying the value of N

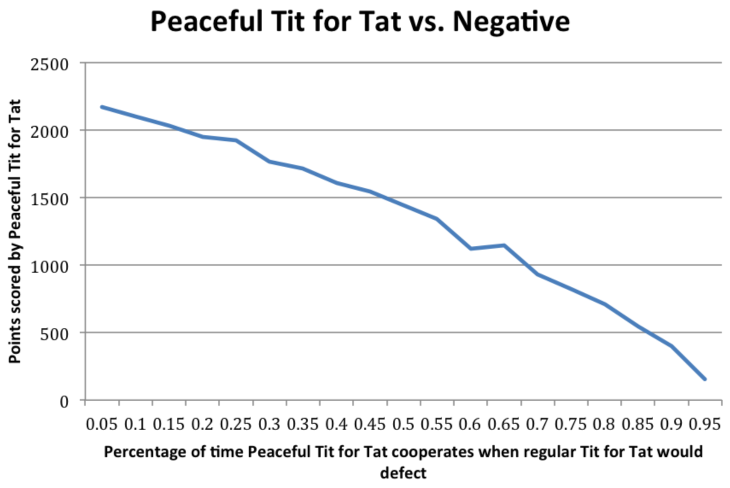


Figure 2.4: When Peaceful Tit for Tat closely resembles Tit for Tat, it is more successful against Negative, as it will mainly alternate between cooperate-defect and defect-cooperate, so its average score per turn will be close to 2.5. As the percentage of time that this strategy cooperates increases, Negative becomes more exploitative.

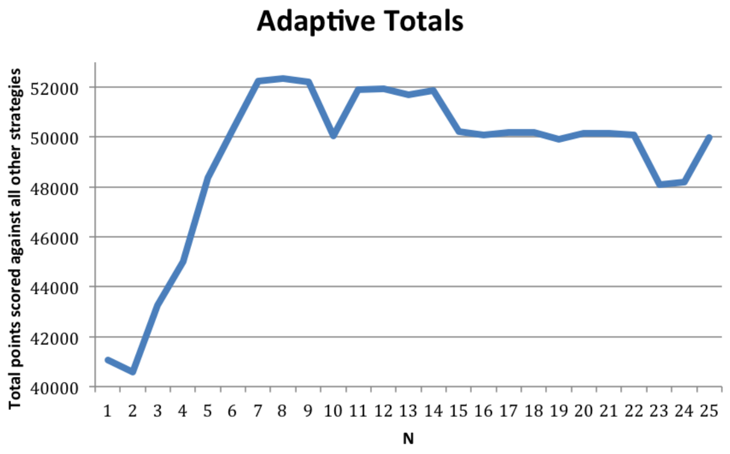


Figure 2.5: This is a graph of the totals of Adaptive's scores against all the other strategies for the range of parameters from 0 to 25. Larger parameters were not included because Adaptive performs steadily worse with larger parameters. As can be seen, the best scores occur at $N = 7$, $N = 8$, and $N = 9$, with $N = 8$ doing the best.

in each run. These test runs revealed that eight was the best number to assign to N, with seven and nine not far behind. Interestingly, these numbers did noticeably better than others that were not much larger or smaller, such as five, six, and ten. Why this occurred was not immediately apparent, but by graphing how well Adaptive did with each parameter from 1 to 100 against all of the other nineteen strategies (see Fig. 2.5), we were able to get a much better idea of the reasoning behind the success of these N values.

Against a plurality of the strategies, Adaptive's performance slightly decreased as N grew larger. This occurred against strategies where there was a clear optimal way of playing against them (for example, always defecting against Always Cooperate); the more quickly Adaptive learned what this optimal strategy was, the better it did. This suggested that plugging in lower numbers for N would make Adaptive more successful.

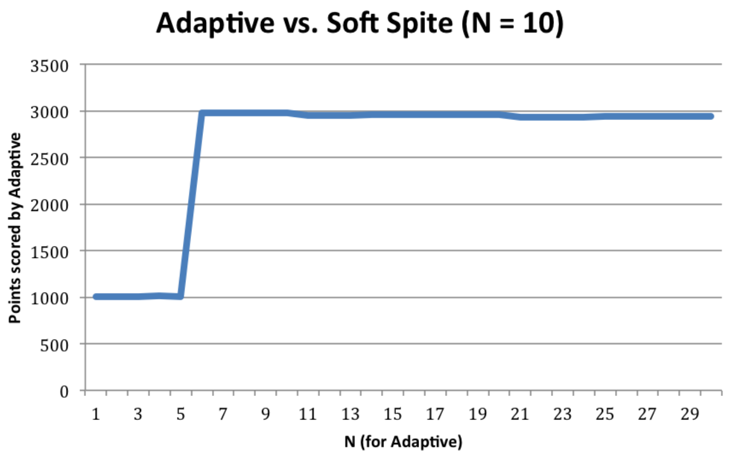


Figure 2.6: Soft Spite here uses a parameter of 10. For Adaptive, N values well below 10 score very poorly due to its early switch to defecting, which not only eliminates points gained from mutually cooperating, but also sets defects as the most successful play, so that the game will be played out with continuous mutual defection.

However, against the next largest group of strategies, Adaptive did very poorly with low values of N, then spiked to a much higher score around $N = 5$, followed by a slow decrease in success as N grew larger (see Fig. 2.6 for Adaptive's performance against Soft Spite).

These spikes occurred at $N = 3$ (Tit for Tat), $N = 4$ (Suspicious Tit for Tat), $N = 5$ (Probing Tit for Tat), and $N = 6$ (Soft Spite and Tit for Two Tats). In the case of Soft Spite (with the number of consecutive defects fixed at 10) though, Adaptive's score then climbed slowly until $N = 10$ before starting its descent, and with Probing Tit for Tat, the scores seemed to climb until about $N = 8$ or $N = 9$. This means that N values of six or less are not the optimal choices overall. As for the numbers just greater than nine, interesting things happen when Adaptive plays against Gradual and Hard Adaptive. Based on what Adaptive learns to do against Gradual, and what it and Hard Adaptive learn from each other, these parameter values do not produce as high of scores as $N = 7$, $N = 8$, and $N = 9$.

Consider the game between Gradual and Adaptive using a parameter of $N = 8$. Adaptive will begin with $N + 1 = 9$ cooperates, and thus so will Gradual, giving Adaptive an average score of $27/9 = 3$ when it cooperates. Then Adaptive will begin to defect. On the first defect Gradual will still cooperate because that is what Adaptive did in the last round, so Adaptive will score 5 points. In the next three rounds Gradual will defect once and cooperate twice while Adaptive defects all three times. This gives Adaptive an average score of $16/4 = 4$ when defecting. However, in the next four rounds Gradual will defect all four times because Adaptive has already defected four times, and Adaptive will also defect all four times to finish its trial period of $N = 8$ defects. For each of these rounds Adaptive will only gain 1 point. This brings its average score down to $20/8 = 2.5$ when defecting. After this Gradual will cooperate twice and Adaptive will go back to cooperating because $3 > 2.5$. The two strategies will continue cooperating for the rest of the game because Adaptive will maintain an average score of 3 points per round when cooperating, and Gradual only ever defects when Adaptive does, which will never happen after these initial rounds. In this scenario Adaptive ends up with a total score of 2996, which is very close to 3002, the optimal score when playing against Gradual, obtained by cooperating in every round except the last one.

Now consider the game between Gradual and Adaptive using the parameter $N = 10$. Again, in its trial period of cooperates Adaptive will average 3 points per round because Gradual will cooperate with it until it defects. Then Adaptive will defect and Gradual will cooperate, gaining Adaptive 5 points. The next seven rounds will look identical to the game where Adaptive used 8 for its N value, so it will have an average score of 2.5 when defecting. However, this time when Gradual cooperates twice, Adaptive will defect both times to finish its $N = 10$ defects. This gains Adaptive another 10 points for an average of $30/10 = 3$ points per round when defecting, creating a tie in average scores between cooperating and defecting, which is broken in favor

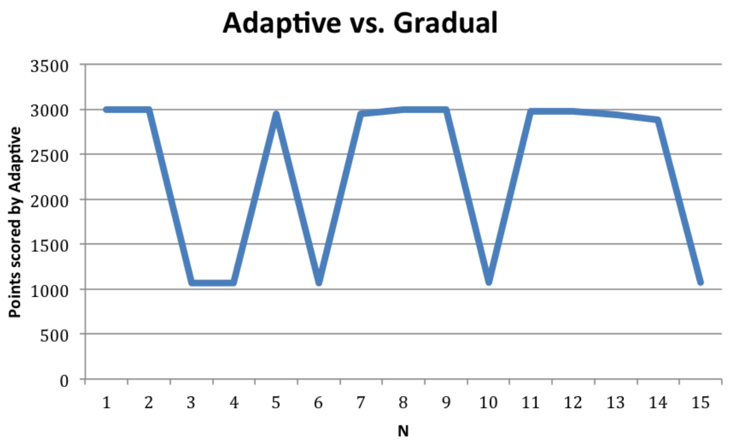


Figure 2.7: Most notable about this graph are the spikes in score for Adaptive. Such a spike is due to whether Adaptive will "decide" that cooperating or defecting is the best strategy.

of cooperating. But now Gradual will defect for the next ten rounds because Adaptive has defected a total of ten times. During these ten rounds Adaptive will first cooperate and get 0 points, bringing its average for cooperates down to $33/12 = 2.75$. Then it will defect because $2.75 < 3$ and gain 1 point, making its average for defecting $31/11 = 2.82$. It will then defect again bringing its average down to 2.66. Now $2.75 > 2.66$ so it will cooperate again, gain 0 points again and bring its cooperating average down to 2.54. This sort of pattern continues until after the tenth time Gradual defects, when Adaptive has 2.25 points per round when defecting and only 2.2 points per round when cooperating. Then Gradual performs its two obligatory cooperates while Adaptive continues defecting and gains 10 points bringing its defecting average back up to 2.56. This sends Gradual into another even longer string of defects, which further brings down Adaptive's average score when cooperating. In the end Adaptive gets a total score of 1070, much lower than the 2996 it acquired with the parameter of $N = 8$ (see Fig. 2.7).

2.6 Constructing the Optimal Strategy

To code a strategy that would cumulatively score more points than the twenty given others, we decided to use a "divide and conquer" approach. Our strategy, named Everything, can be thought of as a more discerning version of Adaptive. Whereas Adaptive uses an opening set of moves to determine whether cooperating or defecting nets it more points on average, Everything uses an opening set of moves to determine which strategy it is playing against in that particular game, then employ a tailor-made strategy to score the highest possible number of points against its opponent. The first step in coding Everything was crafting our opening salvo. Since Adaptive was one of the most successful strategies, we initially tried the opening used by Adaptive with its best parameter of $N = 8$, that is, eight cooperates followed by seven defects. However, some strategies we wanted to treat differently responded to this opening identically. After some minor tinkering, we found that seven cooperates followed by eight defects, and then five more cooperates allowed us to discriminate between opponents based on how we wanted to respond to them.

We then grouped the list of twenty strategies into opponents that we should always defect against, opponents we should always cooperate against, and opponents we should alternate against. Given the particulars of the opponents' strategies, employing one of these three strategies against our opponent after the first twenty moves would almost always net the best possible score (see Appendix C). Next, we found patterns that each of these groups followed in the first twenty moves and coded our strategy to recognize these patterns and employ the optimal strategy. For example, against any strategy that defected during moves 17, 18, and 19, our optimal choice was to always defect. The result was a strategy that fared far better against the list of twenty strategies than any of the strategies on the list itself, scoring approximately 60,000 points, whereas Gradual, the next best strategy, scored only 57,000.

There are a few ways in which our strategy could be improved, however. First, it is hard to categorize strategies with elements of randomness. For example, it is better to always cooperate against both Probing and Peaceful Tit for Tat and Random Pavlov, yet better to always defect against Random. The problem is that because of the randomness in these strategies' decisions it is hard to separate them. To minimize this problem, we decided to always cooperate against strategies that we did not recognize because the benefits of cooperating with the random Tit for Tats and Random Pavlov outweigh the damage of always cooperating with Random itself. This brings us to the potentially larger problem with our strategy; it was tailor-made to beat the specific strategies on our list of twenty. There is the possibility that if our strategy played against strategies similar to these twenty but with slight adjustments it could choose the wrong optimal strategy and end up losing out heavily. For example, a strategy that behaves identically to Tit for Tat during the first twenty moves, then defects for the rest of the game would obliterate our strategy, which is programmed to always cooperate against Tit for Tat. In the first twenty moves each strategy would garner 45 points, but in the remaining 980 moves Everything would get 0 points while the opponent would gain 5 points per turn for a total of 4945 points. However, as we mentioned earlier, our strategy is designed to beat common opponents, as opposed to more unorthodox strategies.

It may be possible to develop a mechanism for recognizing and differentiating between Probing Tit for Tat, Peaceful Tit for Tat, Random Pavlov, and Random. To recognize the Tit for Tats, for example, our strategy could calculate how many moves were identical to Tit for Tat, and if the percentage of the moves that were identical was above, perhaps, 80%, then our strategy could identify Peaceful or Probing Tit for Tat and proceed to always cooperate. On the other hand, if the twenty first moves of the opposing strategy were unrecognizable and did not have a high enough percentage to be Peaceful or Probing Tit for Tat, our strategy could calculate how evenly distributed the other strategy's moves were between cooperates and defects. That is, if the other strategy cooperated approximately 50% of the time, our strategy could recognize this as Random and consequently always defect.

However, this tactic has the potential to misidentify certain strategies at a high cost. For example, Spiteful differs from Tit for Tat by only four moves out of the initial twenty. If the necessary percentage of identical moves was 80%, then Spiteful could be wrongly identified as Probing or Peaceful Tit for Tat. Spiteful will continue to defect for the rest of the game, and as the optimal strategy against either random version of Tit for Tat is to always cooperate, our strategy would not earn any more points for the rest of the game. Similarly, Tit for Tat differs from Suspicious Tit for Tat by only one move, and it differs from Pavlov by only four moves. The optimal strategy against those strategies is again to always cooperate, so by misidentifying, our strategy would lose out on many more points in that regard. Our strategy is thus not perfect, but it has greatly outperformed the others that we considered.

A logical next step in the investigation of the iterated prisoner's dilemma would be to develop a pool of strategies not included in our tournament, then run our optimal strategy against this pool. Some strategies in this pool could simply be the same as in our tournament but with different parameters (for example, we could include Adaptive ($N = 6$)), while some would have to be completely different. One significant flaw that could be expected is that our strategy is set to always cooperate against unrecognizable strategies, which leaves it vulnerable to the possibility of other strategies easily taking advantage of it. One adjustment we might have to make, then, would be to have our strategy always defect, instead of always cooperate, against unknown strategies. By Proposition 2.2.1, this is the best choice to make against any unidentified opponent.

3. Incan Gold

The next game we studied was Incan Gold. Again, we were trying to find a successful strategy for the game of Incan Gold, but our measure of success had to change slightly due to differences between Incan Gold and the Iterated Prisoner's Dilemma. First, because Incan Gold has been studied in a formal setting significantly less than IPD, we did not have a list of twenty common strategies to use as a test base. We had to create our own set of strategies that we deemed reasonable based on playing Incan Gold ourselves. Second, because of the way Incan Gold is played, our metric for success was not the total number of points a strategy earned but rather the percentage of games a strategy won. Thus we ran simulations attempting to divine which of our reasonable strategies won most often and if, based on that, we could create even more successful strategies.

3.1 Introduction to Incan Gold

The game Incan Gold is based on a similar principle to IPD. At each turn, a player must choose one of two options: stay or leave. Also similar to IPD, the player's choice will be affected by the choices of her opponent(s). However, there are several key differences between the two games, chief among which are that in Incan Gold, the player's choice is influenced by factors other than her opponent's choice(s), and that these factors introduce an element of luck.

In this game, each player dons the role of an explorer searching for treasure in an ancient Incan temple [MF06]. There is a deck of cards, each showing either a treasure or a hazard. Treasure cards appear in different denominations ranging from one to seventeen gems, and there are five types of hazard: fire, spiders, zombies, rock fall, and snakes. Each turn the card on the top of the deck is revealed. If it is a treasure, all the players remaining in the temple split the amount specified on the card. If it is a hazard, no players get any treasure, and if two cards showing the same hazard are revealed in a round, all the players remaining in the temple are killed. After each card is revealed each player can either stay in the temple or leave with the treasure they have collected. Players' choices are shown simultaneously as in rock-paper-scissors. If a player leaves, he keeps the treasure he had collected up to that point for the rest of the game, but does not collect any more treasure for the remainder of the round. If a player stays, he may keep exploring the temple and collecting treasure, but at an increasing risk of being killed by hazards. If a player dies, he loses all the treasure he had amassed that round. The round ends when all players have either left or been killed, and the cards revealed in that round are shuffled back into the deck. Each game consists of five rounds in total, and the winner of the game is whichever player has procured the most gems.

It may occur during a round that a treasure card appears with a number that is not evenly divisible by the number of players. In this case, players use division with remainder to each collect an equal number of gems, and those leftover are placed on top of the card. When a player leaves, she takes with her all treasure that is on the cards. However, if two or more players leave on the same turn, they split the treasure on the cards. If this amount of treasure is not divisible by the number of players leaving, the same process is applied and the remainder is left on the cards. There are other optional rules to this game that we did not employ in our studies for simplicity's sake. These rules deal with a special type of treasure card, called an artifact, and the removal of ceratin hazards throughout the game.

The crux of Incan Gold is risk versus reward. At each turn the player has to decide whether to be conservative and leave with what she has, or risk losing her treasure in an attempt to gain more. As more hazards appear, the player is incentivized to leave because the risk of death increases, but as more of the other players leave, the player is incentivized to keep exploring, because any treasure found will be split between fewer people. On the other hand, as more treasures appear, the player may be incentivized to leave if there are many treasures left on top of the cards, but if she leaves at the same time as an opponent (or opponents), then they each get fewer treasures, which is beneficial to anyone who did not leave at that point. This is the clearest parallel between Incan Gold and Prisoner's Dilemma. If two players leave at once, they come out exactly even. If one player leaves and the other players stay, then the player who left gets all the treasure left on the table and the others get none. The difference is that by staying, the other players gain the opportunity to make more than the player(s) who left earlier.

One last important difference between Incan Gold and IPD is that the former is designed so that between two and eight players can play at once, whereas IPD must have exactly two players. Initially, we tested strategies against varying numbers of opponents, but to pin down our metric of success, we insisted on three player games. The third player adds further complexity to the game, taking us another step beyond IPD, but also significantly increases variability. The addition of a fourth player increased variability even more, to the point where running thousands of games did not yield consistent results, which is why we capped the number of players at three.

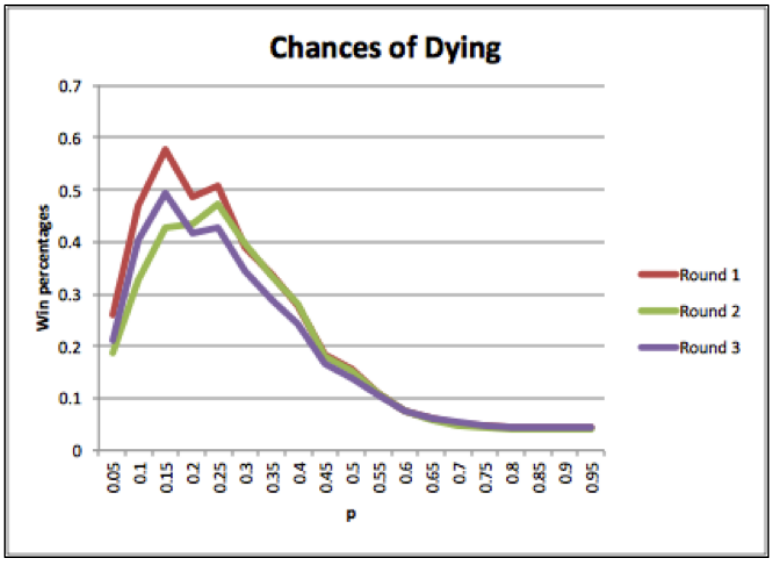
3.2 Constructing Basic Strategies

Unlike IPD, Incan Gold is not associated with a bank of well-known strategies that have been tested against each other in competition. We coded nine strategies for a computer to run based on how we made decisions during actual gameplay. For example, one strategy we employed was being the last player to leave during every round. This strategy is among the riskier but as a result has the potential for higher payoffs: though by staying in the game longer, a player is likely to face more hazards that could spell death, the player is also likely to earn more treasure, and with no opponents left in play, the last one to leave does not have to share the spoils. This strategy, fittingly named Last To Leave, was easy enough to code (if there are other players left, stay; else, leave), but was not very successful in terms of win percentages. If two copies of Last To Leave were playing a game together, neither would ever be willing to leave and each round would end the same way: both copies would eventually die, earning no treasure. The nine basic strategy descriptions can be found in Appendix D.

Six out of the nine strategies involved parameters, and as in IPD, we wanted to see which parameters did best. First, we needed to decide what "best" meant. In Incan Gold, a player wins by having the most gems at the conclusion of the five rounds, whether he is ahead by a single gem or by twenty, so we decided to look at the percentage of times a strategy won its games rather than its average score in them. By having a computer simulate hundreds of rounds of gameplay between different strategies using a range of parameters, we were able to estimate what constituted good parameter values. The strategy Chances of Dying, which stays until the probability that the next card flipped over will prove deadly is greater than p, did fairly well with $p = .25$; though other values of p may have done better against specific opponents, this value is the one that did reasonably well against all opponents. We fixed our parameters with these initial estimates, then one by one had each strategy that involved a parameter take on a range of values and play against the bank of those fixed-parameter strategies plus the strategies that did not involve a parameter. Each copy of a strategy with a given parameter played every combination (including duplicates of the same strategy) of possible opponents from the bank in 1,000 games, and we recorded its win percentage of all those games combined. For Chances of Dying, using $p = .15$ gave the highest win percentage against the bank. Our next step was to fix the parameters with these new values and repeat the process, so that each strategy would be forced to play against a better bank. We repeated the same process once more after this; though each repetition gave us more information about what the ideal parameter values should be, we felt three iterations of the process would give us reasonable approximations without taking away time from studying other methods of improving our strategies. After the three rounds, we found Chances of Dying's ideal parameter to be $p = .15$ (Fig. 3.1).

The highest point in each curve corresponds to the value of p that netted Chances of Dying the highest win percentage against its bank of opponents. The peaks shift from left to right and back because it plays a different bank of opponents each round. When strategies employ the parameter values assigned after round one, Chances of Dying does better when it plays a little riskier at $p = .25$, but when strategies employ the parameters assigned after round two, Chances of Dying wins more often using $p = .15$. The peak also shifts up and down because of the difference in the banks. In round two, Chances of Dying is playing strategies that are objectively better than those it played against in round one; rather than playing strategies that use our best guess at a good parameter, it plays strategies that use the best parameter determined in thousands of rounds of gameplay. Since Chances of Dying is playing against better opponents, its peak in round two is lower than its peak in round one, which was true of most strategies. We believe this indicates that those strategies that do not involve a parameter were able to win more often as their parameterized counterparts killed each other off. However, the peak in round three is higher than that in round two, because Chances of

Figure 3.1: Chances of Dying Win Percentages



Dying now plays against the same opponents who dropped in win percentages between round one and round two, so it is able to win more often at $p = .15$.

Surprisingly, the strategies that perform the best in Incan Gold are those that optimize the reward instead of focusing on the risk. That is, strategies like Chances of Dying and Seen Enough Evils, which calculate, in one way or another, how likely they are to die in the next round, do not do nearly as well as strategies like Seen Enough Gems, which set a bar for how much treasure they must collect before leaving (see Appendix E).

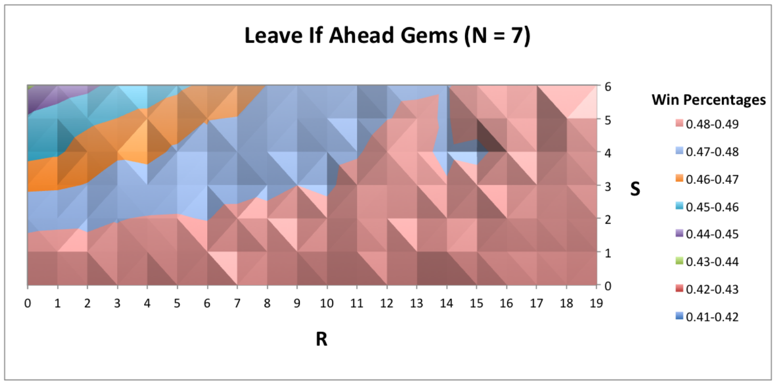
3.3 Optimizing the Basic Strategies

One significant difference between IPD and Incan Gold is the level of variability in the scores achieved by each strategy in different games. Incan Gold involves high degrees of chance and luck not present in IPD, to the extent that even when simulating a large number of games and then averaging the scores, the same strategy does not necessarily always win in Incan Gold. Because of this variability, it was more difficult to identify the one strategy that performed best and thus to find a basis for an optimal strategy as we had in IPD. We also could not use the "divide and conquer" method we had previously employed because the variability, as well as the increased use of strategies involving parameters, makes it impossible to determine which players are being faced in a given game. However, based on many test runs, we found that the Seen Enough Gems strategy generally had the highest win percentages versus the other strategies that we employed. In Seen Enough Gems, the player chooses to leave after collecting N gems. This strategy performs well for several values of N, though when playing against this particular pool of strategies, these successful values of N fall within a limited range. It is clear, for example, that if we set $N = 0$ and the player will leave after the first card is turned up each turn, the player will hardly have any time to collect any gems, and the win percentage is very low. On the other hand, if we set N to be much larger than 20, the win percentage drops drastically, as the player will nearly always be killed before seeing this many gems. The reasonable range for parameter values was thus between 4 and 16, and the optimal parameter for this strategy turned out to be $N = 7$ (see previous section for exact details of how we arrived at the optimal parameter).

It should be noted that there is a high degree of randomness in the game Incan Gold, whereas the game of IPD involves no randomness at all. Thus, we had to make certain assumptions in order to determine the optimal parameter for Seen Enough Gems. Most significantly, we assumed that the parameters for other strategies would be realistically set in a certain range and that when Seen Enough Gems played itself, it was playing a copy that had a fairly low, but reasonable, value for N.

After finding that Seen Enough Gems was the most successful strategy out of our pool, we attempted to construct an optimal strategy based on it. This initial optimal strategy operated on the assumption that any

Figure 3.2: Leave If Ahead Gems ($N = 7$) Win Percentages



player would want to behave differently depending on whether he is in the lead and if so, by how much. For example, if x is the standard number of gems a player waits to see, but he finds himself behind by twenty points, he may want to wait until he sees $x + 5$ gems to make up the discrepancy. On the other hand, if a player is ahead by two points, he may want to leave after seeing $x - 2$ gems in order to preserve his small lead. To construct the strategy, we initially set the value of N to be 7; that is, the player will leave after seeing 7 gems. This new strategy is programmed to behave identically to Seen Enough Gems in round 1. In rounds 2, 3, and 4, the player will behave differently depending on how many points separate him from the leader or, if this player is in the lead, the player in second place. This introduces a second parameter to the strategy; each round, the player determines whether he is within R points of the first or second place player. Once this has been determined, an adjustment may be made to the original parameter N. If he is up to R points behind first or ahead of second place, he does not alter N; that is, he plays identically to regular Seen Enough Gems within this window. If he is in first place and more than R points ahead of second, he becomes more conservative, leaving after seeing $N - S$ gems, where S is a third parameter. If he is behind first place by more than R points, he becomes riskier, leaving after seeing $N + S$ gems. This new strategy thus takes three parameters: N, S, and R. In the final round, the strategy is completely different: the player will leave only if they are in the lead. Since this is the last round, leaving any earlier ensures certain loss.

The next step in constructing this strategy was to determine the optimal parameters for S and R (N remained fixed at 7). To do this, we ran trials, using the usual tournament set-up, varying both S and R. Given that S represents the change in how many gems the player must collect before leaving, the maximum value for S had to be 7, since the value $N - S$ cannot be negative. Thus, the possible values for S were 0 through 7. No such restrictions were present for the value of R, though, and thus the possible values for R were 0 through 19. Hypothetically, R has no necessary upper bound, but realistically, being behind or ahead by more than 19 points should not be treated any differently from being behind or ahead by exactly 19 points. The two situations are both rare and roughly equally dramatic, so we opted to treat them as the same case. We ran simulations for each possible combination of R, S, and pair of opponents. Based on the data we obtained, the optimal parameter values were $R = 18$ and $S = 5$. That is, when the player is ahead of second place by more than 18 points, she leaves after collecting $N - 5$ gems, and when the player is behind first place by more than 18 points, she leaves after collecting $N + 5$ gems. Otherwise, she leaves after collecting the usual N gems.

Using these parameters, this strategy does better than the basic version of Seen Enough Gems with the same N. This is shown in the fact that when $S = 0$, Leave If Ahead Gems is identical to Seen Enough Gems, and, as above, the optimal parameter for S is not 0. See Fig. 3.2 for the graph of Leave If Ahead Gems with varying R and S.

This strategy, however, proved to not necessarily be the optimal one. We used a chi-squared test to determine the statistical significance of our results, and given the number of trials we initially used, the Leave If Ahead Gems 7 (18, 5) strategy could not be proven to be the best strategy of those tested. When tested against the average score of all versions of Leave If Ahead Gems 7 using $S = 0$ (i.e. the average win percentage of Seen Enough Gems 7), the chi-squared test revealed a $p - value$ of .31, which very clearly does not disprove the null hypothesis that the two strategies are equally successful. This indicated that we had not run enough trials to accurately state whether the optimal strategy was in fact the best of all possible parameter combinations. Additionally, since we could not reject the null hypothesis when comparing Leave If Ahead Gems 7 to the base strategy, Seen Enough Gems 7, we could not say if we had even found a better strategy at all.

Thus, we proceeded to run further simulations using ten times as many trials as in the original tournaments. Additionally, given the fact that we determined the optimal value for N by running Seen Enough Gems against other strategies with varying parameters, and thus the optimal value of $N = 7$ that we initially found was relative to the parameters used in other strategies, we chose to consider other values for N, running simulations of our new strategy for $N = 6$, $N = 8$, and $N = 9$. The new data that we obtained from these new trials revealed interesting trends both between trials using different N values and within trials of the same N value.

The first notable result from our simulations of 10,000 trials was that $N = 7$ did not appear to be the best base parameter after all. When comparing the trials for $N = 7$ and $N = 8$, we used the chi-squared test again to determine the statistical significance of our results, using average values for each set of trials as our data point. The $N = 8$ trials were more successful than the $N = 7$ trials; this data is backed up by a $p - value$ of 0.0225. The $N = 9$ trials were more successful than the $N = 8$ trials; the $p - value$ here was 0.0185, so we conclude with 98.15% confidence that using $N = 9$ is better than using $N = 8$. Due to time constraints, we stopped running trials after $N = 9$, so it is possible that a higher value of N, perhaps $N = 10$ or $N = 11$, could produce better results.

Within the $N = 9$ trials, when comparing the optimal strategy, which occurred with $R = 12$ and $S = 2$, to the original Seen Enough Gems 9 strategy, the chi-squared test produced a $p - value$ of 0.064, which is slightly higher than the value commonly used to reject the null hypothesis. On the other hand, when comparing between $N = 9$ trials with different values for R and S, we had more difficulty definitively finding the optimal strategy. For example, Leave If Ahead Gems 9 $(12, 2)$ had a win percentage of 48.998, while Leave If Ahead Gems 9 $(13, 2)$ had a win percentage of 48.952; an average win percentage is 33.3, as there are three players competing in each trial, so each version certainly does well, but we wanted to conclude that the higher scoring version actually does better. Our null hypothesis in this case was that Leave If Ahead Gems 9 $(12, 2)$ and Leave If Ahead Gems 9 $(13, 2)$ are equally successful strategies. Given the number of trials run to produce these figures, the $p - value$ found in the chi-squared test was 0.55, hardly low enough to reject the null hypothesis. Thus, while we can say with some confidence that $N = 9$ is best parameter when $N \leq 9$ for our new strategy, we cannot say the same about $R = 12$ and $S = 2$. However, it is likely, though not certain, that this strategy is indeed better than Seen Enough Gems with $N = 9$.

Given the uncertainty surrounding the data collected for this strategy, especially the fact that the win percentages for the base strategies of Seen Enough Gems for each parameter were so comparable to the scores for our new Leave If Ahead Gems strategies, our final step in analyzing Incan Gold was to investigate other ways to tailor a more successful strategy. The next strategy that we designed included another parameter such that the original S value is split into $S1$ and $S2$, where $S1$ is the amount the strategy adjusts by if it is ahead, and $S2$ is the amount it adjusts by if it is behind. Additionally, we allowed $S1$ and $S2$ to take both positive and negative values. That is, we abandoned the assumption that the strategy will perform better if it is always more risky when behind by a certain amount and more conservative when ahead by a certain amount. Due to time constraints, this round of simulations featured 100 round games, instead of 1,000 or 10,000. We also chose to vary $S1$ and $S2$ by increments of 2, from -8 to 8, to limit the number of trials necessary while maintaining a large range of values.

Because of the low number of trials used in this analysis, it is difficult to pinpoint the absolute best parameters, although the apparent best strategy had the values $R = 2$, $S1 = 4$, and $S2 = 2$. That is, when this strategy is ahead by 2 or more points, it waits to collect 13 gems before leaving, instead of 9, and when it is behind by 2 or more points, it waits to collect 11 gems before leaving, again instead of 9. To gather a more comprehensive picture of the data trends, we considered the 100 strategies that got the lowest scores and the 100 strategies that got the highest scores out of the thousands of strategies tested. For each set of 100 strategies, we averaged the $S1$ and $S2$ values to attempt to determine in what situations it is better to be risky or conservative. Table 3.1 details our results.

As we can see from Table 3.1, the strategies that tend to be more risky are the ones that perform the best overall. One caveat about this data is that the average decision change size was taken without consideration of when these decisions are being made. That is, there is the potential of a relation between how far ahead or

	Lowest 100 Strategies	Highest 100 Strategies
$S1$	-4.03960396	1.108910891
$S2$	-4.396039604	4.138613861

Table 3.1: Best and Worst Strategies

behind this player is and how much the strategy changes. However, a cursory glance at the data suggests this is not true, at least superficially. Consider, for example, the following two strategies: Leave If Ahead Gems ($R = 1, S1 = 2, S2 = 6$) and Leave If Ahead Gems ($R = 13, S1 = 2, S2 = 6$). The former strategy wins 49.259% of the time, while the latter wins 49.26% of the time. These two strategies start varying their decisions at very different points; what appears to matter is that they increase the number of gems they will wait for instead of decreasing that number. This example perhaps indicates that the trend seen in the above table of more risky strategies performing better is correct.

There is clearly much more analysis that can be done regarding Incan Gold. This final consideration of another set of strategies revealed perhaps an even more robust possibility for a dominant strategy than our first attempt. Further analysis might involve weeding out strategies in this final test that were quite unsuccessful, then running more trials in order to obtain a clearer picture of what types of decisions lead to successful strategies.

4. Dominion

The final game we investigated was Dominion, a complex, card-based strategy game. Our goal in studying Dominion was not to find one optimal strategy, as we strived to do with Incan Gold and IPD, because the possible cards and variations of the game are so high in number. Instead, we attempted to discover several different successful strategies, each potentially successful in a different situation.

4.1 Introduction to Dominion

Official Dominion rules can be found at `riograndegames.com/getFile.php?id=348` [Vac08]. Briefly, each player in a group of 2-4 starts with an identical deck of ten cards and must decide which additional cards to buy of those available; in our studies, we limited every game to two players. The goal of the game is to collect the most victory points, which are obtained through the purchase of victory cards. This leads us to the three main types of cards: treasure, action, and victory. The denominations of treasure cards are Copper, Silver and Gold, and each has a distinct worth and cost. The worth of a card is how many coins a player is allowed to spend for each copy of the card she has in her hand; Copper is worth 1, Silver, 2, and Gold, 3 (so a player with two Silvers and a Copper in her hand may buy a card costing up to 5 coins). The cost of a card is how many coins are needed in hand to gain a copy of that card; Copper costs nothing, Silver, 3, and Gold, 6 (so the same player with 5 coins can afford more Copper or Silver but not Gold). With sufficient amounts of treasure in hand, a player can also buy action cards. Between the basic Dominion set and various expansions, there are over a hundred different action cards, but we limited our study to 23 actions, carefully choosing cards with a wide range of abilities (see Appendix F). During human gameplay, exactly ten distinct actions are available for purchase in any given game, but in our research we relaxed this standard to study en masse how actions cards interact with each other. If a player has an action card in hand, she may play her action prior to making a purchase. For example, if she has the Smithy action card, playing it allows her to add the top three cards from her deck into her current hand, ideally giving her more treasure to buy more expensive cards. A Market card, on the other hand, lets her turn up one extra card, play another action, make an extra purchase, and adds one coin to the total amount provided by the treasure coins already in her hand.

The last main type of card is victory cards: Estates, Duchies and Provinces. During gameplay, these cards are not worth any coins and do not provide any actions, but at the conclusion of gameplay, the player with the highest number of victory points wins. An Estate costs 2 coins and is worth 1 victory point, a Duchy costs 5 coins and is worth 3 victory points, and a Province costs 8 coins and is worth 6 victory points. Though it may be tempting to stock up on the more affordable Estates and Duchies, in practice these cards clog a deck and are nowhere near as useful as Provinces. Curse cards are a fourth variety that we encountered in our studies but which do not necessarily appear in every game. Certain actions allow a player to curse his opponents by giving each one a Curse card worth -1 victory point. We used only the most basic Curse-related card, Witch, an action which lets a player add two cards off the top of his deck to his hand and forces each of his opponents to add a Curse card to their own decks.

Each player starts with three Estates and seven Coppers, shuffles this deck of ten, and draws off the top five to form their hand. Each turn, a player is entitled first to one opportunity to play an action (one action), and then to one opportunity to make a purchase (one buy). During the first turn, no player has any action cards in hand, so everyone starts with a buy. Certain action cards may give a player an extra action or buy (see description of Market above). After a player has exhausted her actions, she concludes her turn with her buy(s). The hand she has just played with, including any cards she has added to her hand from her deck via actions and any new cards she has acquired via buys, go into a discard pile next to her deck, and she draws the next five cards off the top of her deck to form her new hand. When there are fewer than five cards remaining in her deck (that is, all but four or fewer are in the discard pile), she shuffles the discard pile and the cards remaining in the old deck, if any, to form her new deck, then draws off the top five cards to form her new hand.

Generally, once a card appears in a player's deck, it is there for the rest of the game. Hence, a Gold costs more coins to buy than it is worth in a single turn; the repeated use of the same Gold card for 3 coins as it cycles through a player's deck negates the one-time, initial cost of 6. The exception to this rule is that certain actions allow a player to trash cards from his hand, removing them from his deck permanently. We considered three cards that grant the ability to trash others. The main trashing card is Chapel, which lets a player trash up to 4 cards from his hand. This is certainly useful in combatting Curse cards, but can also be used to weed out relatively weak cards such as Copper and Estates, so that stronger cards turn up more frequently. The other cards in our study that allow a player to trash cards are Mine and Moneylender, both of which can only be used conditionally (see Appendix F for details).

At the start of gameplay, there are multiple copies of each card on the table: for a two-player game, there are 60 Coppers, 40 Silvers, 30 Golds; 8 Estates, 8 Duchies, 8 Provinces; and 8 or 10 of each type of action available, depending on the card (plus, potentially, 10 Curse cards). The game is over either when the Province pile has been exhausted (that is, at the purchase of the 8th Province), or when any other three stacks of cards have been exhausted, whichever comes first. At this point, players calculate the total number of victory points in their decks, and the player with the highest total wins. When we began to code strategies for Dominion, we focused on win percentages as our metric for success. As in Incan Gold, winning an individual game by one point or twenty points is less important than an ability to win consistently.

4.2 General Dominion Strategy Notes

Two major categories of Dominion strategies are often employed. "Big Money" strategies, such as Big Money itself, are strategies that prioritize buying money cards and Province cards, forgoing the action cards for the most part. The Big Money strategy, for example, buys the most expensive card that it can afford of Silver, Gold, and Provinces each turn. Big Money thus never buys and consequently never plays any actions, as any player is entitled to do during any of his turns. These kinds of strategies aim to fill the deck with mainly Gold and Silver for the purpose of drawing hands that have high buying power. They do not often benefit from extra buys or actions, instead relying on being able to buy a powerful card every turn. However, some variations of Big Money strategies throw in two or three copies of one action card, limiting the number of this action that can be bought to maximize its benefits without clogging the deck or risking drawing two of the same action in one hand, thus rendering one of the action cards irrelevant. "Engine" strategies, on the other hand, aim to string together potent series of actions, aiming to draw enough cards and play enough actions to stock up enough money to buy a powerful card. The Village-Smithy strategy is a classic example of an engine. The Village card allows the player to keep playing actions, while the Smithy card allows the player to beef up their hand by drawing more cards from their deck.

Beyond these two general categories, there are, of course, particular strategies that can be tailored to generate significant success. One such strategy that we designed, which proved to be quite successful, was the Gardens-Pawn-Workshop (GPW) strategy. Any strategy that includes Gardens must be engineered to amass a huge deck. Additionally, in order to prevent other players from exhausting the store of Provinces too quickly, this strategy must target three piles to run out sooner. We chose to target the Gardens, Pawns, and Estates to exhaust first, as Gardens cards are the lynchpin of the whole strategy, Pawns are a low-cost card that can give an extra buy, and the Estate pile is relatively small to begin with. As the strategy is written, this strategy prioritizes buying Workshops first, until the deck contains two Workshops, followed by Gardens, Pawns, and Estates. Unlike many other strategies that generally want to avoid clogging the deck with relatively powerless Coppers, this GPW strategy will buy a Copper if left with an extra buy and less than two coins. In the action phase of GPW's turn, Pawns are played first, followed by Workshops. The Pawns are set to always give an extra buy to this player, followed by either an extra action, if GPW has another action in hand, or an extra coin. Should a Workshop be played, GPW will prioritize collecting another Gardens, followed by Workshop, Pawn, and Estate, in that order.

This strategy was inspired by an attempt to create a successful strategy using Gardens, Workshop, and Throne Room. We struggled, however, to develop an engine that would run out three piles quickly enough using those three cards. One significant difference between GPW and Gardens-Throne Room-Workshop is the cost factor in each strategy. In GPW, we aim to run out one pile of cards that cost 4 and two piles that cost 2, whereas in Gardens-Throne Room-Workshop, both Gardens and Throne Room cost 4. Hence, this strategy would be able to buy each card less often and would thus collect more Coppers. Though the deck size would be similar to that of GPW, the game would last longer, thus giving opponents more opportunities to buy victory points. By using Pawns, GPW can more quickly run out three piles and end the game.

Big Money is the most basic strategy possible; despite its simplicity, it often appears unbeatable to new players buying actions at random and in incorrect proportions. It is thus commonly used as a benchmark: a strategy that beats Big Money a majority of the time is at least decent, and one that doesn't is not [Big]. Our first step in analyzing Dominion was to look at Turbo strategies—variations on Big Money that involve a single action card. First, we began with the basic Big Money code:

- `If there are 8 or more coins in hand, buy a Province`
- `If there are 6 or 7 coins in hand, buy a Gold`
- `If there are 3-5 coins in hand, buy a Silver`
- `If there are fewer than 3 coins in hand, do nothing this turn`

Next, we modified the code for Turbo to take on two parameters: which action a player buys, and how many copies of it. Let Action denote the former, and let Limit denote the latter. Since we used actions ranging in cost from 2 to 6 coins, our version of Turbo had to take the cost of the action card into account when deciding to buy the action or more money:

```
• If Action is in the hand, play Action
• As regards the buy, if there are 8 or more coins in hand, buy a Province
• If there are 6 or 7 coins in hand:
    – If Action costs 6 coins and there are fewer than Limit copies of Action in the deck,
      buy whichever there are fewer of in the deck: Gold or Action, breaking ties towards
      Action
    – Otherwise (if there are already Limit copies of Action in the deck, or if Action costs
      less than 6), buy a Gold
• If there are 2-5 coins in hand:
    – If Action costs 4 or 5 coins:
        ∗ If there are fewer than Limit copies of Action in the deck, buy an Action
        ∗ Otherwise, buy a Silver
    – If Action costs 2 or 3 coins:
        ∗ If there are fewer than Limit copies of Action in the deck, buy whichever there
          are fewer of in the deck: Silver or Action, breaking ties towards Action
        ∗ Otherwise, buy a Silver
• If there are fewer than 2 coins in hand, buy nothing this turn
• If there there are multiple buys available this turn, these steps for deciding what to
  buy are repeated for the extra buy(s)
```

We played each action card on our list as Action in the Turbo strategy, using 1, 2, 3, 4 and 5 as the possible corresponding Limit values, in 100 games against Big Money. Any strategies that won less than half the time against Big Money were cropped from our list of contenders, and only the most successful versions of Turbo for given action cards were kept (for example, buying 4 Smithys and buying 1 Smithy both won more than half their games with Big Money, but 1 Smithy won more often than 4 Smithys, so only the former remained on our list).

After selecting the most successful Turbo strategies, we turned our attention to developing powerful Two Card strategies. We recoded Turbo to accept four parameters: a primary action card (Action1), a secondary action card (Action2), and their respective limits (Limit1 and Limit2). This required more complicated code:

- `During the action phase, if Action1 and Action2 are both in the hand, always play Action1 ahead of playing Action2. Otherwise, play any available action, given the chance`
- `As regards the buy, if there are 8 or more coins in hand, buy a Province`
- `If there are 6 or 7 coins in hand:`
  - `If Action1 costs 6 coins and there are fewer than Limit1 copies of Action1 in the deck, buy whichever there are fewer of in the deck: Gold or Action1, breaking ties towards Action1`
  - `If that is not the case (that is, if Action1 does not cost 6 coins and/or there are already Limit1 copies of Action1 in the deck), if Action2 costs 6 coins and there are fewer than Limit2 copies of Action2 in the deck, buy whichever there are fewer of in the deck: Gold or Action2, breaking ties towards Action2`
  - `If neither of these are the case, buy a Gold`
- `If there are fewer than Limit1 copies of Action1 and fewer than Limit2 copies of Action2 in the deck, and there are enough coins in hand to buy both Action1 and Action2:`
  - `If Action2 is less expensive, buy Action1 instead`
  - `Otherwise, buy Action2`
- `If none of the above cases are true, and there are fewer than Limit1 copies of Action1 in the deck and Action1 costs less than 4 coins, buy whichever there are fewer of in the deck: Silver or Action1, breaking ties towards Action1`
- `If none of the above cases are true, and there are fewer than Limit2 copies of Action2 in the deck and Action2 costs less than 4 coins, buy whichever there are fewer of in the deck: Silver or Action2, breaking ties towards Action2`
- `If none of the above cases are true, and there are fewer than Limit1 copies of Action1 in the deck, and there are enough coins in hand to buy Action1, buy Action1`
- `If none of the above cases are true, and there are fewer than Limit2 copies of Action2 in the deck, and there are enough coins in hand to buy Action2, buy Action2`
- `If none of the above cases are true, and there are enough coins in hand to buy a Silver, buy a Silver`
- `Otherwise, buy nothing this turn`
- `If there there are multiple buys available this turn, these steps for deciding what to buy are repeated for the extra buy(s)`

We ran each action card combined with each action card, using Limits of 1-5 (varying independently), against Big Money, giving us an exponentially longer list. As with our Turbo strategies, we removed any combinations that did not win at least half their games against Big Money, as well as any redundant combinations: using Smithy as Action1 with Limit1 = 2 while using Smithy as Action2 with Limit2 = 2 is just Turbo Smith with Limit = 4, so the former was cut from the list. We also kept only the versions of each pair of action cards whose limits enabled that combination to win the most games. Additionally, we removed any pairs that were buying the same cards but in a less successful order. For example, using Fishing Village as the primary action and Laboratory as the secondary action won up to 65.5 games (half a game denoting a tie), with the right limits, but using Laboratory as the primary action and Fishing Village as secondary won up to 71.5, so only the latter remained on our list. Finally, we deleted any strategies that were not as succesful as the Turbo versions of Action1 and Action2: using Council Room as Action1 and Adventurer as Action2 won up to 66 of its games, but Turbo Council Room won up to 79 and Turbo Adventurer won up to 68, so the Two Card version was eliminated.

As we tested different incarnations of strategies, we ran into an interesting hiccup: one set of trials was run with an error in the code that caused these Two Card strategies to alternate between an action and Gold when they had equal costs, but did not buy Gold in any other case. After correcting the error, most strategies improved remarkably, which was expected. The exceptions to the rule were Hoard-Wharf, which won up to 91.5 of its 100 games against Big Money when it forsook Gold, but 90.5 after our correction, and Wharf-Monument, which won up to 97 beforehand but 94.5 after. The difference is not statistically significant, but we decided to keep the supposedly flawed copies of these two strategies that never buy Gold on our master list in light of this surprise.

4.3 The Witch

The card that did the best against Big Money in our preliminary trials was the Witch. The Turbo Witch strategy won up to 97% of its games against Big Money and almost all other strategies involving a Witch did very well as well. This is because of the Witch's ability to Curse. Because Curses give the opposing player -1 victory point, games that would have otherwise been ties end with the Witch strategy winning. Also, Curses clog the opponent's deck, making it less likely that they will be able to buy Provinces. On top of this, the Witch lets the player draw two extra cards into his hand, helping him to buy Provinces and Gold more quickly.

Because of the Witch's success in preliminary trials, and because of its relative rarity (there are other cards with the ability to give Curses, but not many), we set out to find the best course of action to take when the opponent starts buying Witches. First we ran all of the Turbo Witch strategies against each other and found that the optimal parameter was three. This is possibly because it strikes a balance between being able to Curse the other player more frequently and not clogging the deck. Thus, it would win ties against players with fewer witches and be able to buy more expensive cards more frequently than players with more witches.

Next we tried running all the Turbo and Two Card strategies not involving Witches but that still did well against Big Money against Turbo Witch. We did this against Turbo Witch with a parameter of one (Turbo Witch ($N = 1$)) and with a parameter of three (Turbo Witch ($N = 3$)). The logic behind using a parameter of one as well as three was that because the other player is not using Witches, there is no need to Curse them quickly because they will usually end up with all the Curses in their deck by the end of the game anyway. Many of the Turbo and Two Card strategies fared better than Big Money, which was to be expected, but only one won more than 50% of games against Turbo Witch ($N = 1$) (Wharf-Monument won 51%), and none of them won more than 50% of games against Turbo Witch ($N = 3$).

Seeing as these strategies were clearly still being hurt by the Curse cards, we then tried using the defensive cards Moat and Lighthouse. These cards stop the Curse cards from being added to one's deck when an opponent plays a Witch. A few strategies using Lighthouse won more than 50% of their games in our preliminary trials against Big Money, which shows that Lighthouses at least are good for something other than defense. However, we found that when not paired with a Witch of their own, strategies involving these cards fared no better than their non-defensive counterparts. Again, none of them won more than 50% of their games against Turbo Witch ($N = 3$).

Our next contender for beating the Witch was the GPW strategy mentioned earlier. Interestingly, this strategy did the best out of any we had seen so far. It won 66% of its games, approximately how many it won against Big Money. This is because it essentially negates the effect of Curses. The goal of the GPW strategy is to clog the deck as much as possible, and as the player is never going to be able to buy Gold or Provinces anyway, it seldom matters when a Curse is drawn instead of something useful. The negative victory points do not matter either because if the GPW strategy ends up with all the Curses, they add an extra ten cards to its deck, giving it an extra point per Gardens in its deck (see Appendix F for description of Gardens). One extra point each on the eight Gardens cards available in a two-player game reduce the -10 victory points given by the Curses to a mere -2. Additionally, if the Curse pile runs out, the game could end even sooner than it would otherwise, giving the Turbo Witch strategy less time to buy Provinces.

The two other options for beating the Witch were using Chapels and using more Witches. To test the strategies involving Witches we only ran them against Turbo Witch ($N = 3$) because that had already proven to be optimal for playing other Witch strategies. Not surprisingly, there were several strategies that won more than 50% of games when they also used Witches—nine to be precise. This is because they get all the advantages of using Witches and then extra bonuses such as more actions, buys, and money. The best, however, was a Lighthouse-Witch combination, a mixture of offense and defense. This won 67% of its games.

Lastly, we ran strategies involving Chapels against Turbo Witch. We were expecting some of these to do quite well because there were several that performed well against Big Money and because Chapels could be used to trash the Curses the Witch gives. This was not the case. The best strategy that did not involve a Witch was a Wharf-Chapel combination which only won 35% of its games. Even when using a Witch in combination with the Chapels, only the combination of 2 Witches and 2 Chapels won more than 50% of its games. The only explanation we can think of for the failure of these strategies is that when playing Chapel first, the player does not manage to play enough witches and when playing Chapel second, the player hardly ever gets to trash Curses. In this scenario, the Witch-Chapel strategy is very similar to Turbo Witch and is playing against a copy of Turbo Witch that is using the optimal parameter.

4.4 Dominion Tournament

After running all of our tests of strategies against Big Money and the Turbo Witch strategies, a few stood out as quite successful. With these we decided to run a round robin tournament in the style of Axelrod. We simulated 2,000 games between each pair of strategies. In 1,000 of these, one strategy took the first turn and in the other 1,000 the other strategy started. To determine the final results we averaged all of each strategy's win percentages from all the sets of games. The strategies that we selected to compete in the tournament were the following:

- Big Money, Turbo Witch ($N = 1$), Turbo Witch ($N = 3$): These were entered into the tournament as baseline strategies to be used for comparison.

- GPW: This strategy was entered into the tournament because it beat Big Money and had the second highest win percentage against Turbo Witch. Also, it is the only strategy we studied that did not try to win by buying all the Provinces, which made it an interesting contender.

- Wharf-Monument, Nobles-Wharf, Laboratory-Monument: Wharf-Monument uses two Wharfs and one Monument, Nobles-Wharf uses one Nobles and four Wharfs, and Laboratory-Monument uses four Laboratories and one Monument. These strategies were entered because they had the top three win percentages against Big Money. Wharf-Monument, although it performed negligibly worse than the other two against Big Money, also had the highest win percentage against Turbo Witch of all the non-Witch strategies excluding GPW.

- Laboratory-Smithy: Laboratory-Smithy uses two Laboratories and one Smithy. This strategy was chosen because it was the strategy that performed best against Big Money without using a Witch, a Wharf, or a Monument. It was not expected to do especially well in the tournament and was also to be used as a point for comparison.

- Hoard-Wharf 2 and Wharf-Monument 2: Hoard-Wharf 2 uses four Hoards and four Wharfs, while Wharf-Monument 2 uses two Wharfs and one Monument. These strategies were chosen because they were the two best against Big Money when they always purchased their actions before buying any Gold. Interestingly, both did better against Big Money in preliminary trials than their Gold-buying counterparts. This is surprising because we hypothesized that Gold was a stronger card than any of the individual actions we were studying.

- Witch-Wharf, Witch-Market, Witch-Monument: Witch-Wharf uses four Witches and five Wharfs, Witch-Market uses three Witches and three Markets, and Witch-Monument uses five Witches and one Monument. These strategies were selected because they were the non-defensive Witch strategies where Witch was played as the first action with the top three win percentages against Turbo Witch.

- Fishing Village-Witch: This strategy uses five Fishing Villages and four Witches. It was chosen because it is the only strategy where Witch was played as the second action that won more than 50% of its games against Turbo Witch. It won 56% of its games, which is better than any of the strategies (besides Witch-Moat) where Witch was played as the first action.

- Lighthouse-Witch and Witch-Moat: Lighthouse-Witch uses five Lighthouses and two Witches, and Witch-Moat uses three Witches and one Moat. These strategies were entered because both had over 60% win rates against Turbo Witch. In fact, Lighthouse-Witch had the highest overall win percentage against Turbo Witch and Witch-Moat had the third highest.

- Witch-Chapel and Witch-Chapel 2: Witch-Chapel uses two Witches and two Chapels, while Witch-Chapel 2 uses five Witches and one Chapel. Witch-Chapel was chosen because it was the Chapel strategy with the best win percentage against Big Money. Witch-Chapel 2 was chosen because it was the Chapel strategy with the highest win percentage against Turbo Witch.

Before we discuss the results of the tournament it should be noted that almost without exception strategies performed noticeably better when going first. Often times the difference was hundreds of games, meaning that hundreds of games were decided by a single turn. This demonstrates how evenly matched many of these strategies are within an individual game. This also provides a possible explanation for some of the anomalies in the data, discussed later, when we were only running 100 game simulations. See Appendix G for full tournament results.

The winner of the tournament was Lighthouse-Witch, which won more than 50% of its games against every opponent except GPW. In second place was Witch-Monument. This strategy did significantly better against GPW than Lighthouse-Witch did, but still lost more than 50% of its games against GPW. It also lost slightly more than 50% of its games against Witch-Chapel 2 and Witch-Moat. Incidentally, Witch-Moat took third place. Witch-Moat won more than 50% of its games against every strategy except Lighthouse-Witch and GPW, but performed worse against all of the non-Witch strategies than Witch-Monument did. This makes sense because Moats lose most of their usefulness when not playing against attack cards.

Based on these results, it would seem that GPW should actually be winning, seeing as it beat all of the first three finishers. In actuality, it did do the best out of the non-Witch strategies, but it only finished fifth, losing out to Fishing Village-Witch. This is because while it won more than 50% of its games against every Witch strategy, it lost more than 50% of its games against every non-Witch strategy except Wharf-Monument 2 and Big Money. It did especially poorly against Hoard-Wharf 2, only winning 10% of its games. This makes for an interesting dynamic. GPW is clearly the best strategy to use if one's opponent is relying on Witches, but if they are not then GPW fares poorly. On the other hand, if one's opponent is not using Witches, then using Witches against them will allow the player to win more than 50% of the time.

After GPW, in order of place, came Witch-Chapel 2, Turbo Witch ($N = 3$), Witch-Wharf, Witch-Market, Wharf-Monument, Laboratory-Monument, Turbo Witch ($N = 1$), Witch-Chapel, Nobles-Wharf, Hoard-Wharf 2, Wharf-Monument 2, Laboratory-Smithy, and Big Money. For exact percentages and tournament results, see Appendix H There are a few interesting points to note here. Firstly, Turbo Witch ($N = 3$) did fairly well in our tournament. Specifically, it performed better than Witch-Wharf, Witch-Market, and one of the Witch-Chapel strategies, and came very close to the other Witch-Chapel strategy. This, along with the fact that most Two Card Witch strategies did not even appear in our tournament, means that buying non-Witch actions just slows down the Turbo Witch strategy. There are few cards that can improve upon it.

Secondly, although in preliminary runs Witch-Chapel and Wharf-Monument 2 performed better against Big Money than Witch-Chapel 2 and Wharf-Monument, this second set placed much higher in the tournament than their counterparts. In the case of the Wharf-Monument strategies, it turns out that the preliminary trials were an anomaly and Wharf-Monument performed better against Big Money in the tournament than Wharf-Monument 2 did. This shows that Gold is at least as strong a card as even the best of the actions that we studied. In the case of the Witch-Chapel strategies, this is because more than half the strategies in the tournament involved Witches and Witch-Chapel 2 did significantly better against all of those than Witch-Chapel did. This is probably because Witch-Chapel 2 using its five Witches could get more Curses into the opponent's deck more quickly. This, however, is not in accordance with our preliminary trials, suggesting another anomalous data point.

4.5 Further Research

One of the selling points of Dominion is the diversity of play that comes with being able to use different sets of cards for each game. By the same token, it is an extremely complex task to learn the strategy of Dominion.

Our study only looks at 23 kingdom (action) cards. In all the expansions currently available there are a total of 205 kingdom cards. A game can be played with any combination of ten of these cards. Thus there are currently $\binom{205}{10}$ possible games of Dominion. Additionally, we only studied Turbo and Two Card strategies (and the singular example of GPW). In a real game, players can buy any combination of the ten actions available to them. Thus, functionally, there are infinitely many Dominion strategies, so there will always be more research that can be done.

However, there are several logical next steps that could be taken based on our research. Firstly, some expansions use Colonies—victory cards that are worth 10 victory points and cost 11 coins—and Platinums—treasure cards that are worth 5 and cost 9 coins. These cards could potentially raise the question of whether it is better to wait and buy these cards or not, as drawing a hand that could afford either card may take much longer than it would to draw a hand that could afford Gold or even a Province. We could also look at the data we already collected from a more bigger picture view point to see whether in general engine or Big Money style strategies perform better. Alternatively, we could attempt to uncover and analyze other unique strategies, such as GPW, that are not engines yet do not play similarly to Big Money. Lastly, all of our simulations only involved two players and adding more changes several of the dynamics of the game, so research could be done on three and four player games.

5. Conclusion

In games such as IPD, Incan Gold, and Dominion, finding the best strategy is, as we have shown, not an objective task. How well any strategy performs is highly dependent on the strategies that its opponents are utilizing. Thus, our investigation had to incorporate knowledge of the opponents' strategies, which is not necessary in games such as checkers or tic-tac-toe. It is, however, possible to find strategies that perform well most of the time. Our "Everything" strategy in IPD incorporates the ability to detect the opponent's strategy and react accordingly. The variations of Seen Enough Gems that change based on their score relative to their opponents' also have the ability to react to the strategies employed by the other players. As for Dominion, the effects of Witches on various strategies' success shows clearly too that the reaction to the other strategy is crucial in producing success.

Appendices

A. List of 20 strategies

1. Always Cooperate: Always cooperates
2. Always Defect: Always defects
3. Alternating: Starts with a random move, then switches each round
4. Tit for Tat: Cooperates, then does what the opponent did in the last round
5. Suspicious Tit for Tat: Same as Tit for Tat, but starts with defect
6. Tit for Two Tats: Cooperates unless the opponent's last two moves are defects
7. Neg: Cooperates, then does the opposite of what the opponent did last round
8. Spiteful: Cooperates until the opponent defects, then always defects
9. Soft Majority: Starts with cooperate, then does what the opponent has done most often, breaking ties in favor of cooperating
10. Hard Majority: Same as Soft Majority, but breaks ties in favor of defecting
11. Pavlov: Starts with cooperate, then, if the last action had a good outcome (mutual cooperates or defecting while the opponent cooperates), does the same thing. Switches otherwise
12. Hard Pavlov: Same as Pavlov, but starts with defect
13. Gradual: Cooperates until the opponent defects, then defects the total number of times the opponent has defected during the game. Follows up with two cooperates
14. Random(p): Cooperates with probability p and defects with probability $1 - p$
15. Soft Spite(N): Cooperates until an opponent defects, then defects N times
16. Adaptive(N): Starts with $N + 1$ cooperates and N defects, then does the thing that has gotten the higher average score per round so far, with ties broken towards cooperating
17. Hard Adaptive(N): Same as Adaptive, but breaks ties towards defecting
18. Probing Tit for Tat(p): Same as Tit for Tat, but defects when Tit for Tat would cooperate with probability p
19. Peaceful Tit for Tat(p): Same as Tit for Tat, but cooperates when Tit for Tat would defect with probability p
20. Pavlov Random(p): Same as Pavlov, but does makes the opposite decision with probability p

B. Strategy Scores

Strategy	Score of strategy during round robin tournament
Always Cooperate	39,897
Always Defect	38,944
Alternating	39,282
Tit for Tat	51,758
Suspicious Tit for Tat	42,256
Tit for Two Tats	49,770
Neg	34,115
Spiteful	52,108
Soft Majority	49,786
Hard Majority	49,248
Pavlov	48,928
Hard Pavlov	41,314
Gradual	57,455
Random ($p = .5$)	38,538
Soft Spite ($N = 10$)	54,524
Adaptive ($N = 8$)	55,184
Hard Adaptive ($N = 7$)	55,143
Probing Tit for Tat ($p = .2$)	40,896
Peacemaking Tit for Tat ($p = .2$)	53,074
Random Pavlov	36,840

C. How Everything Divides and Conquers

Strategy Everything plays against	How Everything responds	Why this scores the most possible points
Always Cooperate	Defects in each round	Always Cooperate will never retaliate so we can score 5 points in every round
Always Defect	Defects in each round	Always Defect will never cooperate so the best we can do in each round is 1 point by defecting back
Alternating	Defects in each round	In rounds where Alternating cooperates, we score the most points by defecting. In rounds where Alternating defects, we score the most points by still defecting
Tit For Tat	Cooperates in each round	Tit for Tat will respond to a defect in one round by defecting in the next, so the best move is to never defect in the first place
Suspicious Tit For Tat	Cooperates in each round	Suspicious makes the same decisions as Tit For Tat after the first round, so it is still best to always cooperate with it
Tit For Two Tats	Alternates between cooperates and defects in each round	By alternating between the two, Tit For Two Tats will never defect because we will never have two defects in a row following our opening set of moves. This way we can alternate between earning 3 points and 5 points each round
Negative	Defects in each round	If we defect in each round, Negative responds by cooperating in each round, so we score 5 points every time
Spiteful	Defects in each round	Because we must defect during our opening set of moves to determine which strategy we're playing against, Spiteful will defect against us for the rest of the game, playing identically to Always Defect. Since we always defect against Always Defect, we treat the Spiteful the same way

Soft Majority	Alternates between cooperates and defects in each round	In our open set of moves, we cooperate more times than we defect, so Soft Majority cooperates with us. By alternating choices in each round, we keep the balance of cooperating more often than defecting overall, so Soft Majority continues to cooperate with us, while we alternating between scoring 3 and 5 points each round
Hard Majority	Alternates between cooperates and defects in each round	Similar to Soft Majority, alternating keeps the overall balance of cooperating more often than defecting, so Hard Majority will continue to cooperate with us while we alternate between scoring 3 and 5 points each round
Pavlov	Cooperates in each round	After our opening set of moves, Pavlov will continue cooperating for the rest of the run if we do. If we ever defect, Pavlov will switch as well, so we are better off never defecting at all
Hard Pavlov	Defects in each round	After our opening set of moves, Hard Pavlov decides to defect. If we also defect, Hard Pavlov decides this is a negative outcome and switches to cooperate, so the best move for us is to defect again. As long as we continue to defect, Hard Pavlov will continue to switch between defecting and cooperating, so our scores will alternate between 1 and 5 points for an average of 3 over the entire run, the same as the score we get for always cooperating with regular Pavlov
Gradual	Cooperates in each round	After our opening set of moves, Gradual will cooperate with us as long as we cooperate with Gradual, but if we defect, it will defect back an increasing number of times. The best move is then to never defect at all
Random	Cooperates in each round	It is impossible to predict what Random will do in a given round, so by Proposition 2.2.1, it is best to always defect. However, our strategy relies on its ability to recognize strategies during the first twenty moves, but cannot distinguish strategies that involve randomness from each other. Against the other three strategies that involve randomness, the best strategy is to always cooperate (see analysis of these strategies below), so generally speaking, the best strategy is to always cooperate against a strategy we don't recognize, because 75% of the time this will be the correct strategy. Therefore even though always defecting is the best strategy against Random, we maximize our overall score against every strategy by always cooperating

Soft Spite	Cooperates in each round	Soft Spite will cooperate with us as long as we cooperate with it, but will punish a defect on our part with ten defects on its. Therefore the best strategy is to never defect
Adaptive	Alternates between cooperates and defects in each round	After our set of opening moves, Adaptive will have an average score of 2.77 points in the rounds where it cooperates and 1 point in the rounds where it defects, so it will cooperate with us. Adaptive will continue to cooperate with us as we alternate, because on average it will score 1.5 points per round by cooperating (3 in the rounds where we cooperate, 0 in the rounds where we defect), and 1.5 is greater than the 1 point per round we taught it to expect to earn when defecting during our opening set of moves
Hard Adaptive	Defects in each round	After our set of opening moves, Hard Adaptive will have scored, on average, .75 points per round by cooperating and 3.83 points per round by defecting, so it will defect against us. Hard Adaptive will continue to defect against us if we defect or cooperate, because it scores 1 point per round when we defect and 5 points we round when we cooperate, both of which are greater than the .75 points it expects to get by cooperating after our opening set of moves. Since Hard Adaptive now plays as Always Defect, we treat it the same way
Probing Tit for Tat	Cooperates in each round	Probing Tit for Tat generally plays like Tit for Tat, but we cannot know when Probing will make a contrary choice, so the best strategy is to always cooperate. If we defect against it, it will certainly defect against us in the next round, but if we cooperate with it, it will cooperate with us most of the time and we will average more than 1 point per round
Peaceful Tit for Tat	Cooperates in each round	Peaceful Tit for Tat will cooperate with us for as long as we cooperate with it. If we defect against it, it will defect against us in the next round 80% of the time, so it is better to never defect at all, because we cannot know when the other 20% of the time (rounds where we could defect without consequence) will occur. Therefore we should always cooperate with Peaceful Tit for Tat

Random Pavlov	Cooperates in each round	Random Pavlov generally behaves like Pavlov, and accordingly we want to treat it the same way. If we cooperate with it, it will usually cooperate with us. Random Pavlov will occasionally defect even though we've both been cooperating, then consider our cooperate with its defect a positive outcome and continue to defect, but it will also make another contrary decision eventually and switch back to cooperating with us. Though occasionally defecting could break Random Pavlov out of a string of defects, it could also send it into such a string, so it is better for us to always cooperate with it

D. Incan Gold Basic Strategies

1. Leave With Probability: leaves with probability p at any given stage
2. Seen Enough Gems: stays until obtaining a given number of gems
3. Seen Enough Evils: stays until a given number of hazard cards have shown up
4. Table Treasures: stays until there are a certain number of gems left over on the treasure cards that could not be divided equally among the players
5. Nth Treasure: stays until N treasure cards have shown up
6. Chances Of Dying: stays until the probability that the next card turned over will prove deadly is greater than p
7. Leave If Ahead: never leaves unless it has more gems than every other player
8. Last To Leave: never leaves if any other player is still in the game
9. Last To Leave Greedy: leaves only after every other player has left and a subsequent treasure card has been turned up

E. Incan Gold Basic Parameterized Strategies' Scores

Strategy	Parameter	Win Percentage
Leave With Probability	$p = 0.2$	29.9%
Seen Enough Gems	$N = 7$	53.3%
Table Treasures	$N = 3$	38.4%
Seen Enough Evils	$N = 2$	40.2%
Nth Treasure	$N = 3$	53.2%
Chances of Dying	$p = 0.15$	49.2%

F. Dominion Action Cards

Card	Cost	Ability
Adventurer	6	Turns up cards from the deck until two treasure cards appear, at which point those cards are added to the hand and any others turned up are discarded
Chapel	2	Trashes up to 4 cards from the hand. Our copy of chapel prioritizes trashing Curses, then Estates. Once the player has at least two treasure cards in higher denominations than Copper, it will trash all Copper, and once there are no Estate or Copper remaining, it will trash any redundant Chapel cards
Council Room	5	Turns up 4 extra cards and grants one extra buy, but allows every other player to turn up one extra card as well
Explorer	5	If there is a Province card in hand, reveal it to add a Gold to the hand; else, add a Silver to the hand
Festival	5	Allows the player two extra actions, one extra buy, and adds two extra coins to the in-hand total
Fishing Village	3	Allows the player two extra actions and adds one coin to the in-hand total. Also allows the player one extra action and adds one coin to the in-hand total during the turn immediately after Fishing Village is played
Gardens	4	At the conclusion of gameplay, each Gardens card is worth 1 victory point per ten cards in the deck
Hoard	6	Adds two coins to the in-hand total. If a player purchases a victory card with Hoard in hand, he also gains a Gold
Laboratory	5	Turns up two extra cards and allows the player one extra action
Lighthouse	2	Allows the player one extra action and adds one coin to the in-hand total. One extra coin is also added to the in-hand total during the turn immediately after Lighthouse is played. If a player is attacked after playing a Lighthouse on her last turn, she does not suffer the effects of the attack (for our purposes, that means she does not have to add a Curse to her hand when an opponent plays a Witch)
Market	5	Turns up one extra card, allows the player one extra action and one extra buy, and adds one coin to the in-hand total
Merchant Ship	5	Adds two coins to the in-hand total. Also adds two coins to the in-hand total the turn immediately after Merchant Ship is played
Mine	5	Trashes a treasure card from the hand and adds a treasure card costing up to 3 more coins to the hand
Moat	2	Turns up two extra cards. If a player is attacked when he has Moat in hand, he does not suffer the effects of the attack (for our purposes, that means her does not have to add a Curse to his hand when an opponent plays a Witch)
Moneylender	4	Trashes a Copper from the hand in exchange for three coins added to the in-hand total
Monument	4	Adds two coins to the in-hand total, and adds one victory point token. This means that for each time Monument is played during the game, the player gets to add one victory point to her score at the end of the game (the token is merely to keep track of how many times the card is played)

Nobles	6	Worth two victory points. Also allows the player to choose between turning up three extra cards or playing two extra actions. Our copy of Nobles will first determine if the player already has the opportunity to play two or more actions without Nobles' help, in which case it will opt for the cards. If not, it makes sure there are at least two other actions to play before opting for the extra actions. Otherwise, it chooses extra cards
Pawn	2	Allows the player to make two distinct choices from four options: turning up one extra card, playing one extra action, making one extra buy, or adding one coin to the in-hand total
Smithy	4	Turns up three extra cards
Village	3	Turns up one extra card and allows the player two extra actions
Wharf	5	Turns up two extra cards and allows the player one extra buy. Also turns up two extra cards and allows the player one extra buy during the turn immediately after Wharf is played
Witch	5	Turns up two extra cards and adds a Curse card to each opponent's deck
Workshop	3	Allows the player to gain any card costing up to 4 coins

G. Dominion Tournament Results

Place	Strategy	Total Average Win Percentage
1st	Lighthouse-Witch	0.639779412
2nd	Witch-Monument	0.626161765
3rd	Witch-Moat	0.617529412
4th	Fishing Village-Witch	0.606720588
5th	GPW	0.594058824
6th	Witch-Chapel 2	0.589985294
7th	Turbo Witch ($N = 3$)	0.581264706
8th	Witch-Wharf	0.564117647
9th	Witch-Market	0.557779412
10th	Wharf-Monument	0.524955882
11th	Laboratory-Monument	0.501779412
12th	Turbo Witch ($N = 1$)	0.474764706
13th	Witch-Chapel	0.449073529
14th	Nobles-Wharf	0.428117647
15th	Hoard-Wharf 2	0.416529412
16th	Wharf-Monument 2	0.374544118
17th	Laboratory-Smithy	0.315411765
18th	Big Money	0.130926471

H. Tournament Results by Strategy

Table H.1: Fishing Village-Witch Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.5785	.472
Turbo Witch ($N = 1$)	.7345	.648
Big Money	.955	.923
GPW	.304	.1935
Witch-Wharf	.6135	.5045
Witch-Market	.6015	.501
Wharf-Monument	.737	.653
Lighthouse-Witch	.439	.3325
Witch-Moat	.4885	.4145
Laboratory-Monument	.7325	.636
Laboratory-Smithy	.8495	.778
Hoard-Wharf 2	.7835	.666
Witch-Chapel	.587	.516
Witch-Monument	.5675	.4055
Wharf-Monument 2	.7825	.7435
Witch-Chapel 2	.5285	.446
Nobles-Wharf	.797	.716

Table H.2: Nobles-Wharf Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.297	.232
Turbo Witch ($N = 1$)	.3395	.2875
Big Money	.974	.9445
GPW	.783	.637
Witch-Wharf	.3095	.2415
Witch-Market	.336	.25
Wharf-Monument	.2985	.242
Lighthouse-Witch	.3675	.295
Witch-Moat	.345	.2405
Laboratory-Monument	.3725	.2185
Laboratory-Smithy	.8695	.7405
Hoard-Wharf 2	.78	.632
Witch-Chapel	.5105	.4655
Witch-Monument	.324	.201
Wharf-Monument 2	.5215	.3665
Witch-Chapel 2	.3655	.2815
Fishing Village-Witch	.284	.203

Table H.3: Witch-Chapel 2 Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.606	.4985
Turbo Witch ($N = 1$)	.6825	.5605
Big Money	.859	.8245
GPW	.3985	.2895
Witch-Wharf	.6255	.4985
Witch-Market	.6445	.5255
Wharf-Monument	.653	.5315
Lighthouse-Witch	.414	.318
Witch-Moat	.514	.3965
Laboratory-Monument	.755	.638
Laboratory-Smithy	.7675	.661
Hoard-Wharf 2	.719	.5285
Witch-Chapel	.641	.5295
Witch-Monument	.6165	.435
Wharf-Monument 2	.8165	.3665
Witch-Chapel 2	.3655	.6345
Fishing Village-Witch	.554	.4715

Table H.4: Wharf-Monument 2 Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.2575	.2195
Turbo Witch ($N = 1$)	.486	.3785
Big Money	.89	.804
GPW	.4825	.3665
Witch-Wharf	.3105	.24
Witch-Market	.287	.232
Wharf-Monument	.4775	.3495
Lighthouse-Witch	.3565	.2925
Witch-Moat	.3015	.217
Laboratory-Monument	.3	.2565
Laboratory-Smithy	.643	.49
Hoard-Wharf 2	.526	.383
Witch-Chapel	.4665	.402
Witch-Monument	.1645	.1185
Witch-Chapel 2	.2665	.1835
Nobles-Wharf	.6335	.4785
Fishing Village-Witch	.2565	.2175

Table H.5: Witch-Monument Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.636	.514
Turbo Witch ($N = 1$)	.725	.559
Big Money	.945	.8955
GPW	.349	.247
Witch-Wharf	.644	.502
Witch-Market	.676	.518
Wharf-Monument	.7895	.643
Lighthouse-Witch	.49	.3575
Witch-Moat	.5465	.422
Laboratory-Monument	.8305	.7495
Laboratory-Smithy	.8515	.724
Hoard-Wharf 2	.7445	.569
Witch-Chapel	.647	.547
Wharf-Monument 2	.8815	.8355
Witch-Chapel 2	.565	.3835
Nobles-Wharf	.799	.676
Fishing Village-Witch	.5945	.4325

Table H.6: Witch-Chapel Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.502	.419
Turbo Witch ($N = 1$)	.573	.475
Big Money	.6205	.615
GPW	.142	.0985
Witch-Wharf	.5075	.418
Witch-Market	.502	.4425
Wharf-Monument	.4575	.4325
Lighthouse-Witch	.355	.231
Witch-Moat	.4265	.313
Laboratory-Monument	.528	.474
Laboratory-Smithy	.58	.51
Hoard-Wharf 2	.522	.44
Witch-Monument	.453	.353
Wharf-Monument 2	.598	.5335
Witch-Chapel 2	.4665	.359
Nobles-Wharf	.5345	.4895
Fishing Village-Witch	.484	.413

Table H.7: Hoard-Wharf 2 Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.3705	.2185
Turbo Witch ($N = 1$)	.362	.2645
Big Money	.745	.8555
GPW	.923	.878
Witch-Wharf	.3735	.2275
Witch-Market	.3745	.2285
Wharf-Monument	.377	.2245
Lighthouse-Witch	.414	.3055
Witch-Moat	.376	.2915
Laboratory-Monument	.49	.3515
Laboratory-Smithy	.465	.3935
Witch-Chapel	.56	.478
Witch-Monument	.431	.2555
Wharf-Monument 2	.617	.474
Witch-Chapel 2	.4175	.281
Nobles-Wharf	.368	.22
Fishing Village-Witch	.334	.2165

Table H.8: Laboratory-Smithy Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.22	.1685
Turbo Witch ($N = 1$)	.2495	.1405
Big Money	.6905	.8465
GPW	.63	.51
Witch-Wharf	.2415	.175
Witch-Market	.2375	.166
Wharf-Monument	.2535	.139
Lighthouse-Witch	.3305	.215
Witch-Moat	.2745	.187
Laboratory-Monument	.355	.23
Hoard-Wharf 2	.3935	.535
Witch-Chapel	.49	.42
Witch-Monument	.276	.1485
Wharf-Monument 2	.51	.357
Witch-Chapel 2	.339	.2325
Nobles-Wharf	.2595	.1305
Fishing Village-Witch	.222	.1505

Table H.9: Laboratory-Monument Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.3955	.315
Turbo Witch ($N = 1$)	.6085	.496
Big Money	.948	.884
GPW	.658	.5725
Witch-Wharf	.398	.3135
Witch-Market	.398	.312
Wharf-Monument	.645	.536
Lighthouse-Witch	.4315	.359
Witch-Moat	.406	.3015
Laboratory-Smithy	.77	.645
Hoard-Wharf 2	.6485	.51
Witch-Chapel	.526	.472
Witch-Monument	.2505	.1695
Wharf-Monument 2	.7435	.7
Witch-Chapel 2	.362	.245
Nobles-Wharf	.7815	.6275
Fishing Village-Witch	.364	.2675

Table H.10: Witch-Moat Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.645	.514
Turbo Witch ($N = 1$)	.7655	.6725
Big Money	.9385	.911
GPW	.2925	.183
Witch-Wharf	.647	.541
Witch-Market	.678	.591
Wharf-Monument	.6695	.559
Lighthouse-Witch	.4695	.354
Laboratory-Monument	.6985	.594
Laboratory-Smithy	.813	.7255
Hoard-Wharf 2	.7085	.624
Witch-Chapel	.687	.5735
Witch-Monument	.578	.4535
Wharf-Monument 2	.783	.6985
Witch-Chapel 2	.6305	.486
Nobles-Wharf	.7595	.655
Fishing Village-Witch	.5855	.5115

Table H.11: Lighthouse-Witch Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.7235	.629
Turbo Witch ($N = 1$)	.786	.6795
Big Money	.908	.874
GPW	.274	.198
Witch-Wharf	.7145	.626
Witch-Market	.7325	.608
Wharf-Monument	.6115	.5155
Witch-Moat	.646	.5305
Laboratory-Monument	.641	.5685
Laboratory-Smithy	.785	.6695
Hoard-Wharf 2	.6495	.586
Witch-Chapel	.769	.645
Witch-Monument	.6425	.51
Wharf-Monument 2	.7075	.6435
Witch-Chapel 2	.682	.586
Nobles-Wharf	.705	.6325
Fishing Village-Witch	.6675	.561

Table H.12: Wharf-Monument Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.4205	.3235
Turbo Witch ($N = 1$)	.565	.4805
Big Money	.9705	.9285
GPW	.758	.6495
Witch-Wharf	.432	.3425
Witch-Market	.426	.347
Lighthouse-Witch	.4845	.3885
Witch-Moat	.441	.3305
Laboratory-Monument	.464	.355
Laboratory-Smithy	.861	.7465
Hoard-Wharf 2	.7755	.623
Witch-Chapel	.5675	.5425
Witch-Monument	.357	.2105
Wharf-Monument 2	.6505	.5225
Witch-Chapel 2	.4685	.347
Nobles-Wharf	.758	.7015
Fishing Village-Witch	.347	.263

Table H.13: Witch-Market Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.521	.402
Turbo Witch ($N = 1$)	.6795	.572
Big Money	.923	.8915
GPW	.333	.2675
Witch-Wharf	.5465	.4205
Wharf-Monument	.653	.574
Lighthouse-Witch	.392	.2675
Witch-Moat	.409	.327
Laboratory-Monument	.688	.602
Laboratory-Smithy	.834	.7625
Hoard-Wharf 2	.7715	.6255
Witch-Chapel	.5575	.498
Witch-Monument	.482	.342
Wharf-Monument 2	.768	.713
Witch-Chapel 2	.4745	.3555
Nobles-Wharf	.75	.664
Fishing Village-Witch	.499	.3985

Table H.14: Witch-Wharf Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.5235	.4225
Turbo Witch ($N = 1$)	.6915	.5555
Big Money	.9385	.898
GPW	.3215	.243
Witch-Market	.5795	.4535
Wharf-Monument	.6575	.568
Lighthouse-Witch	.374	.2855
Witch-Moat	.459	.353
Laboratory-Monument	.6865	.602
Laboratory-Smithy	.825	.7585
Hoard-Wharf 2	.7725	.6265
Witch-Chapel	.582	.4925
Witch-Monument	.498	.356
Wharf-Monument 2	.76	.6895
Witch-Chapel 2	.5015	.3745
Nobles-Wharf	.7585	.6905
Fishing Village-Witch	.4955	.3865

Table H.15: GPW Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.727	.634
Turbo Witch ($N = 1$)	.747	.6575
Big Money	.762	.653
Witch-Wharf	.757	.6785
Witch-Market	.7325	.667
Wharf-Monument	.3505	.242
Lighthouse-Witch	.802	.726
Witch-Moat	.817	.7075
Laboratory-Monument	.4275	.342
Laboratory-Smithy	.49	.37
Hoard-Wharf 2	.122	.077
Witch-Chapel	.9015	.858
Witch-Monument	.753	.651
Wharf-Monument 2	.6335	.5175
Witch-Chapel 2	.7105	.6015
Nobles-Wharf	.363	.217
Fishing Village-Witch	.8065	.696

Table H.16: Big Money Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.0755	.0575
Turbo Witch ($N = 1$)	.064	.0705
GPW	.347	.238
Witch-Wharf	.102	.0615
Witch-Market	.1085	.077
Wharf-Monument	.0715	.0295
Lighthouse-Witch	.126	.092
Witch-Moat	.089	.0615
Laboratory-Monument	.116	.052
Laboratory-Smithy	.1535	.3095
Hoard-Wharf 2	.1445	.255
Witch-Chapel	.385	.3795
Witch-Monument	.1045	.055
Wharf-Monument 2	.196	.11
Witch-Chapel 2	.1755	.141
Nobles-Wharf	.0555	.026
Fishing Village-Witch	.077	.045

Table H.17: Turbo Witch ($N = 1$) Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 3$)	.3955	.303
Big Money	.9295	.936
GPW	.3425	.253
Witch-Wharf	.4445	.3085
Witch-Market	.428	.3205
Wharf-Monument	.5195	.435
Lighthouse-Witch	.3205	.214
Witch-Moat	.3275	.2345
Laboratory-Monument	.504	.3915
Laboratory-Smithy	.8595	.7505
Hoard-Wharf 2	.7355	.638
Witch-Chapel	.525	.427
Witch-Monument	.441	.275
Wharf-Monument 2	.6215	.514
Witch-Chapel 2	.4395	.3175
Nobles-Wharf	.7125	.6605
Fishing Village-Witch	.352	.2655

Table H.18: Turbo Witch ($N = 3$) Win Percentages

Opponent	Going First	Going Second
Turbo Witch ($N = 1$)	.697	.6045
Big Money	.9425	.9246
GPW	.366	.273
Witch-Wharf	.5775	.4765
Witch-Market	.598	.479
Wharf-Monument	.6765	.5795
Lighthouse-Witch	.371	.2765
Witch-Moat	.486	.355
Laboratory-Monument	.685	.6045
Laboratory-Smithy	.8315	.78
Hoard-Wharf 2	.7815	.6295
Witch-Chapel	.581	.498
Witch-Monument	.486	.364
Wharf-Monument 2	.7805	.7425
Witch-Chapel 2	.5015	.394
Nobles-Wharf	.768	.703
Fishing Village-Witch	.528	.4215

Bibliography

[AD88] Robert Axelrod and Douglas Dion. "The Further Evolution of Cooperation." In: *Science* 242 (Dec. 1988), pp. 1385–1390. URL: `http://sdl.soc.cornell.edu/curricular/axelrod_Evol_Extensions.pdf`.

[Big] *Big Money*. Dominion Strategy. URL: `http://dominionstrategy.com/big-money/`.

[Don86] Christian Donninger. "Is It Always Efficient to Be Nice? A Computer Simulation of Axelrod's Computer Tournament." In: *Paradoxical Effects of Social Behavior*. 1986, pp. 123–134. URL: `http://www.socio.ethz.ch/vlib/pesb/pesb9.pdf`.

[MF06] Alan R. Moon and Bruno Faidutti. *Incan Gold*. Gryphon Games. 2006. Board game.

[PD12] William H. Press and Freeman J. Dyson. "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent." In: *Proceedings of the National Academy of Sciences* (Apr. 2012). URL: `http://www.pnas.org/content/early/2012/05/16/1206569109.full.pdf`.

[Vac08] Donald X. Vaccarino. *Dominion*. Rio Grande Games. 2008. URL: `riograndegames.com/getFile.php?id=348`. Board game.