The Conditions for Commitments

Scott N. Gerard

IBM, Research Triangle Park, Durham, NC 27709, USA
sgerard@us.ibm.com

Abstract. Previous work describes the use of formal commitments to mediate the communication between autonomous agents through commitment-based protocols. I extend that work to examine the conditions that encourage the success of agents that use commitments. I define a parameterized and iterated *Committer's Dilemma* game that extends the well-known Prisoner's Dilemma game, and then use this game and different agent strategies to examine how the *conditions for commitments* affect game outcomes.

I describe the results of multiple simulations on a multiagent society with various game parameters. Results show that *committing* and *satisfying* agent types dominate over other agent types most frequently (1) when commitments are frequently exchanged, (2) when games tend to end at other than a Nash equilibria, (3) when the cost to create a commitment is low, and (4) when the utility of a given good is about 40% to 60% of the utility of a received good. A classifier, with over 95% accuracy, is trained to predict which Committer's Dilemma game parameters lead to commitment-based agent strategies dominating the agent society.

1 Introduction

My and other's previous work [5, 4, 6, 3, 14, 9] use a specific and formalized type of commitments [13] (Section 2.1). *Commitments* are used to design and analyze inter-agent *commitment protocols* (or processes) between "debtor" and "creditor" agents, typically in a business context. Commitments are considered a good construct for describing and reasoning about agent interactions because commitments depend only on agent's externally observable, social actions, rather than internal and externally invisible constructs, such as intentions or goals.

Agents are assumed to be completely autonomous. Therefore, no external party can unilaterally force an agent to take an action. This recognizes both that agents may not *choose* to keep their commitments, or they may not be *able* to keep their commitments. However, much of my prior analysis tended to be of the form "if all commitments are kept, then ..., else the analysis does not apply". Effectively, the analysis focuses on cooperative agents who keep their commitments.

A premise of much of my previous work is that commitments provide overall value to agent interactions. How can we characterize the value of commitments? I propose to measure the value of commitments by examining those conditions in which commitment-based agents are more *successful*, where success is defined as the agent's ability to survive relative to other agents in agent-to-agent competition.

The goal of this paper is to begin to extend prior work to consider the conditions that must hold for an agent's use of commitments to make the agent more successful compared to other agents. This moves from an essentially cooperative games setting towards a more non-cooperative games setting, where self-interested agents use commitments because they lead to better outcomes.

I examine the conditions for commitments using a game called *Committer's Dilemma*, which is an extension of the well-known Prisoner's Dilemma game. I define a set of different agent types, and classify those types as *committing* and *satisfying* (C&S) or not. Agent types are pitted against each other in an iterated version of Committer's Dilemma. Each game history, consisting of multiple generations, is classified as *successful* if a C&S agent survives to the final generation. I examine the *conditions for commitment* by examining the relationship between game parameters and *successful* game histories.

Section 2 summarizes relevant background on commitments, the Prisoner's Dilemma game, and my use of evolutionary algorithms. Section 3 described the *Committer's Dilemma* game and the tournament structure. Section 4 describes the agent types, the metric of success, and some details about the experimental simulation runs. Section 5 describes my results. Section 6 discusses the implications, related literature and future work.

2 Background

I rely on my previous work in agent commitments, the Prisoner's Dilemma game, and evolutionary algorithms.

2.1 Commitments

I formulate a *commitment* [13, 4] as being from a set of *debtors* to a set of *creditors* that if the *antecedent* begins to hold, the debtors will bring about the *consequent* in the future. In symbols: $\mathsf{C}_{\{\text{debtors}\},\{\text{creditors}\}}(\textit{antecedent}, \textit{consequent})$. Both antecedent and consequent are Boolean expressions over state-space propositions. When the antecedent becomes true, the commitment is *detached*, and the debtors become *unconditionally* committed to the creditors. When the consequent becomes true, the commitment is *discharged*.

Debtors *should* discharge their detached commitments. However, debtors are autonomous and may violate a commitment by canceling it. The only compliance requirement for commitments is: each detached commitment *must* eventually be discharged (satisfied, transferred or released) or canceled. Also, there is no implicit ordering constraint between the antecedent and consequent.

Commitments provide value to the creditor, but they are a liability to the debtor.

2.2 Prisoner's Dilemma Game

The Prisoner's Dilemma game [2, 12] is a well-known game between two players, each of whom can cooperate (COOP) or defect (DEFECT). Let R be the reward for both giving and receiving, S be the sucker's payoff for giving without receiving, T be the cheater's temptation to receive without giving, and P be the penalty payoff when neither player gives. The Prisoner's Dilemma normal-form game matrix is

$$\mathsf{P} = \begin{array}{c|c|c|} & s_C & s_D \\ \hline s_C & R\,/\,R & S\,/\,T \\ \hline s_D & T\,/\,S & P\,/\,P \\ \hline \end{array} \tag{1}$$

Two conditions characterize the Prisoner's Dilemma. Equation 2 ensures temptation and sucker's payoff are extreme. In iterated games, equation 3 ensures agents don't profit by taking turns defecting.

$$T > R > P > S \tag{2}$$

$$R > \tfrac{T+S}{2} \tag{3}$$

2.3 Evolutionary Algorithms

Autonomous agents compete against each other by playing a specific, two-player game (see Section 3.1). Agents play each other in an iterated series of games. Agents have memory only of their counter-player's (opponent's) previous cooperate or defect moves within that series, and can adjust their strategy accordingly. In this paper, all agent's strategies are fixed and determined solely by the agent's type.

I use many existing concepts from evolutionary algorithms [10, 11, 8]. A society of agents play a tournament of iterated games over multiple generations. Agents that accumulate higher payoff utilities are preferentially selected using a *rank selection* algorithm to be part of the agent society for the next generation. The most successful agents survive until the final generation; the least successful agents die out. I postpone to future work, enabling agent strategy's to evolve using cross-over and mutation operators on an agent's strategy genotype as is done in genetic algorithms.

3 Technical Approach

I describe the Committer's Dilemma game and its parameterization and the definition of the tournament structure.

3.1 Committer's Dilemma Game

I define the *Committer's Dilemma* game that extends the well-known *Prisoner's Dilemma* game by enabling agents to make commitments to each other. The game models an economic exchange of goods (transfer of utility) beween two players, Alice and Bob. It models classic economic exchange scenarios where two players value goods differently and trade something they have (lower utility) for something they want (higher utility). For example, it can model a player trading his fruit surplus (lower utility) for another player's meat (higher utility), and it can model the exchange of movies where players trade movies they have watched (lower utility) for movies they haven't watched (higher utility).

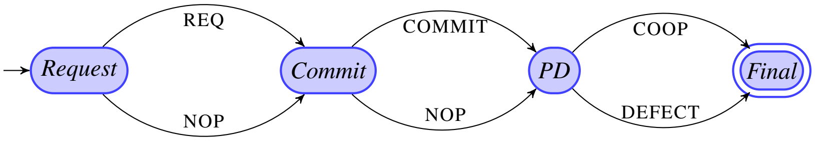

Fig. 1: The four phases and three sets of allowed actions of each player in the Committer's Dilemma. Non-terminal game states have single-line borders, while the terminal game state has a double-line border.

As shown in Figure 1, the *Committer's Dilemma* game has four phases. In each phase, players act simultaneously. In phase n, players have perfect knowledge of their own and their counter-player's previous actions in phases 1 to $n-1$. During phase n, the game infrastructure informs each agent of its counter-player's previous (phase $n-1$) action. Each game consists of three phases where agents respond with an action, plus a fourth phase where agents make no response.

The phases are (1) *Request* phase: each player may request a commitment from their counter-player (action REQ) or not (action NOP). (2) *Commit* phase: if the counter-player requested a commitment in the *Request* phase, player may grant the request (action COMMIT) or reject it (action NOP). Note: there is no incentive for a player to create a commitment without a prior request. This is allowed, but none of the agents in Section 4.1 does so. (3) *PD* phase: the traditional *Prisoner's Dilemma* phase where players cooperate by giving a good (action COOP) or defect by withholding a good (action DEFECT). (4) *Final* phase: players do not choose any action in this phase. This is mearly a technical detail in which players are informed of their counter-player's action during the *PD* phase.

At the end of the game, each player receives a payoff based on (1) whether they created a commitment to their counter-player in the *Commit* phase, and (2) whether they cooperated or defected in the *PD* phase. This creates four essential player responses: (1) response s_C: player makes no commitment and then cooperates, (2) response s_D: makes no commitment then defects, (3) response s_{cC}: makes a commitment then cooperates, and (4) response s_{cD}: makes a commitment then defects. Whether a player requests a commitment or rejects a commitment request, does not affect the payoff utilities, however, it may influence the counter-player's future actions. Making and then violating a commitment does affect payoff utilities.

The four responses for both players generates a symmetric, 4x4 normal form game matrix, G. I construct G from four 2x2 game matrices: G_{PP}, G_{PC}, G_{CP} and G_{CC} (subscripts indicate whether the player is playing a traditional Prisoner's Dilemma game without making a commitment (P) or using commitments (C)). Each of these 2x2 games matrices must satisfy the Prisoner's Dilemma conditions (Equations 2-3).

$$\mathsf{G} = \begin{bmatrix} \mathsf{G}_{PP} & \mathsf{G}_{PC} \\ \mathsf{G}_{CP} & \mathsf{G}_{CC} \end{bmatrix}$$

If neither player makes a commitment (response s_C or s_D), then they are both playing a traditional Prisoner's Dilemma (G_{PP}). Each player performs only action COOP or

DEFECT. Let b be the benefit (utility) a player receives from a good it receives, and let c be the cost (utility) a player gives up for a good it provides. I assume the benefit of a good received must be at least as large as the cost of a good given ($c \leq b$); otherwise the players would not attempt to exchange goods. When a two-way exchange occurs, each player's net change in utility is $b - c$. When an agent receives a good without giving a good, it receives utility b with no cost, and when it gives a good without receiving a good, it loses c with no benefit. When neither agent gives a good, both get zero utility.

$$\mathsf{G}_{PP} = \begin{array}{c|c|c|} & s_C & s_D \\ \hline s_C & b-c\,/\,b-c & -c\,/\,b \\ \hline s_D & b\ \ /\ \ -c & 0\ /\,0 \\ \hline \end{array}$$

Now, consider the case where Alice makes a commitment to cooperate, but Bob does not (matrix G_{CP}). Let i be the additional, up-front investment necessary to make a commitment (suggesting legal fees to create a commitment or contract) which applies to all four cells in G_{CP}. When Alice keeps her commitment (row s_{cC}), she receives no additional benefit; she simply did what she committed to do. But when she violates her commitment (row s_{cD}), she is charged a penalty "tax" of t, and the commitment's debtor (Bob) receives *restitution* r. I assume any restitution is paid out of taxes ($r \leq t$).

I also wish to model the difference between *commitment* semantics where a debtor can unilaterally creates a commitment to a creditor, and (business) *contract* semantics where both agents must accept the contract before commitments are created. Let $w = 0$ represent commitment semantics, $w = 1$ represent contract semantics, and values in-between represent a blend of the two semantics (in-between values allow the analysis to examine intermediate semantics, even though they do not represent real-world possibilities). This value applies only to taxes and restitutions. When $w = 0$, taxes and restitution apply to each agent that has created a commitment. When $w = 1$, taxes and restitution apply to both agents only if *both* agents have created commitments.

The utilities for G_{CP} are the utilities in G_{PP} plus the incremental utilities for Alice making a commitment ($\mathsf{G}_{CP} = \mathsf{G}_{PP} + \mathit{\Delta}\mathsf{G}_{CP}$) where

$$\mathit{\Delta}\mathsf{G}_{CP} = \begin{array}{c|c|c|} & s_C & s_D \\ \hline s_{cC} & -i\ \ /\ 0 & -i\ \ /\ 0 \\ \hline s_{cD} & -wt-i\,/\,wr & -wt-i\,/\,wr \\ \hline \end{array}$$

The case where Bob makes a commitment to cooperate, but Alice does not, is symmetric to the previous case. The game matrix is $\mathsf{G}_{PC} = \mathsf{G}_{PP} + \mathit{\Delta}\mathsf{G}_{PC}$ where

$$\mathit{\Delta}\mathsf{G}_{PC} = \begin{array}{c|c|c|} & s_{cC} & s_{cD} \\ \hline s_C & 0\,/-i & wr\,/-wt-i \\ \hline s_D & 0\,/-i & wr\,/-wt-i \\ \hline \end{array}$$

Finally, the case where both players make commitments to cooperate is $\mathsf{G}_{CC} = \mathsf{G}_{PP} + \mathit{\Delta}\mathsf{G}_{CC}$ where I fix $w = 1$ (contract semantics).

$$\mathit{\Delta}\mathsf{G}_{CC} = \begin{array}{c|c|c|} & s_{cC} & s_{cD} \\ \hline s_{cC} & -i\ \ /\ \ -i & r-i\ \ /\ \ -t-i \\ \hline s_{cD} & -t-i\,/\,r-i & r-t-i\,/\,r-t-i \\ \hline \end{array}$$

Combining and expanding the results above gives the 4x4, normal-form Committer's Dilemma game payoff matrix G parameterized by b, c, i, t, r, and w, shown here as Equation 4.

$$
\begin{array}{c|c|c|c|c|}
 & s_C & s_D & s_{cC} & s_{cD} \\
\hline
s_C & b-c \;/\; b-c & -c \;/\; b & b-c \;/\; b-c-i & -c+wr \;/\; b-wt-i \\
\hline
s_D & b \;/\; -c & 0 \;/\; 0 & b \;/\; -c-i & wr \;/\; -wt-i \\
\hline
s_{cC} & b-c-i \;/\; b-c & -c-i \;/\; b & b-c-i \;/\; b-c-i & -c+r-i \;/\; b-t-i \\
\hline
s_{cD} & b-wt-i \;/\; -c+wr & -wt-i \;/\; wr & b-t-i \;/\; -c+r-i & r-t-i \;/\; r-t-i \\
\hline
\end{array}
\tag{4}
$$

Section 5.1 below describes two specific examples of this game matrix.

3.2 Tournament

Each pair of agents make successive *actions* (or moves) in each *game* as shown in Figure 1. The outcome of each, single game is scored using the players' final states and the Committer's Dilemma game's payoff matrix G in Equation 4.

Each pair of agents play an iterated *series* of multiple (arbitrarily fixed at ten) consecutive games. Agents can remember previous outcomes within a series, but have no memory of competitor actions or game outcomes in any other series. Each agent's series payoff is the sum of its payoffs for each game in the series.

A *generation* is defined by its *society*, which is the set of agents currently "alive". The initial society of agents contains multiple (arbitrarily fixed at three) instances of each of the ten agent types in Section 4.1, giving a society size of 30. In every generation, each agent plays a series against every other agent in that generation's society, including a copy of itself. Each agent's generation payoff is the sum of its series payoffs against all other agents (including itself). At the end of each generation, agents with higher series payoffs are preferentially selected to enter the next generation's society. Agents are ranked from 1 (lowest generation payoff) to 30 (highest generation payoff), and then a stochastic universal selection algorithm, executed on the ranks, randomly selects the surviving agents. Agents that are not selected to enter the next generation die out.

A *history* is a sequence of multiple (arbitrarily fixed at ten) successive generations with a fixed game payoff matrix. A history is defined by fixed parameter values for c, t, r, i, and w.

A *tournament* is a collection of histories, one history for each set of parameter values.

4 Evaluation

I describe the different types of agents used in my experiments, ant then I describe the experiments.

4.1 Agent Types

I define the following types of agents. The first three *basic* agent type versions completely ignore requesting or making commitments. These versions are typical of a traditional Prisoner's Dilemma game.

- `AllD`: Never requests or creates a commitment. Always DEFECT in *PD* phase.
- `AllC`: Never requests or creates a commitment. Always COOP in *PD* phase.
- `TFT` (tit-for-tat): Never requests or creates a commitment. COOP in *PD* phase of the first game of a series. Plays its counter-player's previous move after that.

In the *commitment-based* agent type versions, the agent always requests a commitment from counter-player. Then, the agent creates a commitment if and only if counter-player requested one.

- `cAllD` (commit, then `AllD`): Always DEFECT in *PD* phase.
- `cAllC` (commit, then `AllC`): Always COOP in *PD* phase.
- `cTFT` (commit, then `TFT`): In *PD* phase, if counter-player grants a commitment, then tit-for-tat; if a commitment is not granted, then DEFECT.
- `c2TFT` (commit, then double `TFT`): agent's *PD* phase strategy consists of two, separate, tit-for-tat streams: one stream for *PD* phase actions in this series in which counter-player granted a *Request* phase commitment, and one stream in which counter-player did not grant a commitment.

In the *contract* agent type versions, *both* players must make commitments ("sign a contract") for commitments to be created. Otherwise, the players defect because there is no signed contract. The agent always requests a commitment from counter-player. Then, the agent creates a commitment if and only if counter-player requested one. Without a contract, there is little distinction between COOP and DEFECT.

- `tAllD` (commit, then `AllD`): In *PD* phase, always DEFECT.
- `tAllC` (commit, then `AllC`): In *PD* phase, if both players have COMMITed, then COOP; otherwise there is no contract (DEFECT).
- `tTFT` (commit, then `TFT`): In *PD* phase, if both players have COMMITed, then play tit-for-tat; otherwise there is no contract (DEFECT).

Each agent type is manually classified by two properties: An agent is *committing* if it ever grants commitment requests in any circumstance; non-committing agents never grant a commitment request. An agent is *satisfying* if it ever satisfies one of its own commitments in any circumstance; a non-satisfying agent never satisfies any of its commitments. An agent is *C&S* if it is both *committing* and *satisfying*. Experiments below examine which game parameter values lead to C&S agent types dominating the history.

Classifying agents as either committing or satisfying requires internal knowledge of the agent. As currently defined, it is not always possible to determine these properties solely by examining the agent's action in game play against a single agent. For example, if the counter-party never requests a commitment, it is not possible to determine whether such a request would have been granted or whether the granted commitment would have been satisfied. Future work will attempt to refine these definitions.

The C&S agent types are `cTFT`, `c2TFT`, and `tTFT`. All other agent types are not C&S. For example, agent type `cAllD` is not C&S, even though it always requests commitments and grants all commitment requests, because its ultimate "always DEFECT" strategy is not *satisfying*.

4.2 Success Metric

While my experiments are similar to those used by others [2], I am primarily interested in evaluating game matrices across a wide variety of parameter values. In particular, I want to quantify the value of making commitments: the "conditions for commitments".

I label the agent types that survive to the final generation of a history as *dominant* in that history. Label a history (which has a common set of parameter values) as *successful* if a C&S agent type(s) dominates that history.

Success is measured by comparing the parameter sets for successful histories to the parameter sets for unsuccessful histories. If there is a successful history with $i = 0$, then I label $i = 0$ as "successful". For each value of each parameter value, count the number of successful histories ($SH(p = v)$) and unsuccessful histories ($UH(p = v)$), and compute the ratio of successful histories to the total of all histories, formalized as

$$\mu(p = v) = \frac{SH(p = v)}{SH(p = v) + UH(p = v)}$$

For example, Figure 3 shows $\mu(i = 0) = 0.48$. If $\mu(p = v) = 0$ there are no successful histories that correspond to $p = v$. A parameter value with many successful histories is very successful. A parameter value with an intermediate number of successful histories is partially successful.

In many histories, agent types `TFT` and `AllD` dominate the society. But I am interested in parameter values where commitments show value. Specifically, I am interested in those histories where one or more C&S agent types dominate.

4.3 Experiments

Each society contains three instances of each of the ten agent types in Section 4.1 for a society size of 30 agents. All pairs of agents in the current society play against each other in every generation. The society evolves through ten generations.

The Committer's Dilemma payoff matrices G were generated using Equation 4 from the end of Section 3.1. A simulation was run for each set of parameter values of the game matrix G, with each set of parameters producing one history. If n is the number of distinct values of each parameter, the total number of simulated histories is $n^3 \times \frac{1}{2}n(n+1)$. Parameters c, i, and w each contribute a factor of n, while t and r contribute the final factor due to the constraint $r \leq t$. History simulations were run for each parameter from 0.0 to 1.0 in steps of 0.2, with b always fixed at 1.0. With $n = 6$ distinct values, there are 4536 distinct simulated histories.

In addition to the game parameters, three additional values are derived for each history:

- ccGiven: the average number of commitments an agent gives to a creditor per game. This is 0 for all the basic agent type versions, and is 1 when commitment-based agents play each other.
- ccGotten: the average number of commitments an agent got from a debtor per game. This is 0 for all the basic agent type versions, and is 1 when commitment-based agents play each other.

- endAtNE: the average number of times in which the two agent's final outcomes for a game ended at a Nash equilibria.

The experiments were performed by programs written in Java, running on a 2.3 GHz Intel Core i7 processor, with 8 GB of memory. It took a little over 4 hours to simulate all 4536 game histories for all parameter values. In the current experiments, the agents redundantly play against each other multiple times, inflating the run-time. But I envision future experiments where to accurately simulate complex (stochastic) agent types might require all the processing performed.

4.4 Classifier

Clearly characterizing the conditions where C&S agent types dominate is challenging given the large number of histories. I construct a binary classifier over a set of history features, where the classifier predicts whether C&S agent types will dominate in that history. The classifier was written in the Octave programming language [1] using the fminunc gradient descent algorithm to optimize a regularized cost function over a set of features. Classifier features include the game parameters: c, t, r, i, and w, but not the constant $b = 1$.

Additional classifier features are a constant term and

- pdCheckOrder: 1.0 if G_{PP} and G_{CC} satisfy Equation 2; otherwise 0.0. Note that G_{PC} and G_{CP} are not symmetric and thus can not satisfy this condition.
- pdCheckAverage: 1.0 if G_{PP} and G_{CC} satisfy Equation 3; otherwise 0.0. Again, G_{PC} and G_{CP} can not satisfy this condition.
- NECount: the number of Nash equilibria in the game matrix, which is a value between 1 and 16. The intuition is a large number of Nash equilibria in G will make it easier for agents to find a naturally, good solution.

5 Results

Simulations are run for each set of parameter values, the metric $\mu(p = v)$ is computed for each parameter value, and a classifier is constructed to predict which sets of parameter values predict that C&S agent types will dominate.

5.1 Example Games

The experiments evaluate agents over a 5-dimensional grid of games. To give some intuition, we describe two of those games based on Equation 4 from the end of Section 3.1. In the payoff matrices, the cells for Nash equilibria are shaded.

Example 1 The game defined by parameters $b = 1.0$, $c = 0.6$, $i = 0.0$, $t = 1.0$, $r = 0.6$, and $w = 0.0$ is

	s_C	s_D	s_{cC}	s_{cD}
s_C	0.40 / 0.40	-0.60 / 1.00	0.40 / 0.40	-0.60 / 1.00
s_D	1.00 / -0.60	0.00 / 0.00	1.00 / -0.60	0.00 / 0.00
s_{cC}	0.40 / 0.40	-0.60 / 1.00	0.40 / 0.40	0.00 / 0.00
s_{cD}	1.00 / -0.60	0.00 / 0.00	0.00 / 0.00	-0.40 / -0.40

This game has three Nash equilibria. However, the players would both be better off (0.40 / 0.40) if both cooperate, with or without creating commitments, but there is a large temptation to defect. Agent types `cTFT`, `tAllC` and `tTFT` dominate in this game. This shows that C&S agents can dominate over even `TFT`.

Example 2 The game defined by parameters $b = 1.0$, $c = 0.6$, $i = 0.4$, $t = 1.0$, $r = 0.6$, and $w = 0.0$ is

	s_C	s_D	s_{cC}	s_{cD}
s_C	0.40 / 0.40	-0.60 / 1.00	0.40 / 0.00	-0.60 / 0.60
s_D	1.00 / -0.60	0.00 / 0.00	1.00 / -1.00	0.00 / -0.40
s_{cC}	0.00 / 0.40	-1.00 / 1.00	0.00 / 0.00	-0.40 / -0.40
s_{cD}	0.60 / -0.60	-0.40 / 0.00	-0.40 / -0.40	-0.80 / -0.80

The parameter values are the same as the previous values except for the value of i. This payoffs here are similar to the previous example, but some values are different due to the different value for i. The new payoffs eliminate two of the Nash equilibria, leaving only one. Both players would be better off by cooperating and not committing, but, again, there is a large temptation to defect. Agent type `TFT` solely dominates in this game. This is characteristic of the traditional Prisoner's Dilemma.

5.2 Simulations

Figure 2 shows the number agents, by agent type, alive at the beginning of each generation, summed across all histories. The x-axis is the generation number, and the y-axis is the number of agents. All agent types start with 3 agents per history at generation 0. As histories evolve through subsequent generations, some agent types prosper and some wither. Note that some histories have multiple dominating agent types, so the sum of the counts can be larger than the total number of histories.

`TFT` dominates most frequently, which fits with its the results of [2]. `AllD` dominates the second most frequently because, in many cases, defecting can lead to large payoffs. The commitment-based agent types that most interest us dominate in some histories, but not as frequently. Next, I focus in on those histories.

5.3 Metric Results

Figure 3 graphs $p = v$ vs. $\mu(p = v)$. The y-axis is the success metric which is the fraction of all histories that resulted in a *C&S* agent type (`cTFT`, `c2TFT`, or `tTFT`) dominating the society in the final generation. Each point for a given parameter value sums the number of histories across all other parameter values.

The derived values ccGiven, ccGotten, and endAtNE can have any value between 0 and 1 (whereas the game parameters are constrained to be exactly equal to 0.0, 0.2, 0.4, 0.6, 0.8 or 1.0). In Figure 3, all histories for these derived values in the range $[0.0, 0.2)$ (half-open interval) are grouped and the success metric $\mu(\text{ccGiven} = 0.1)$ for that group of histories is plotted at the range's midpoint 0.1. The other ranges $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$ and $[0.8, 1.0]$ are handled similarly, with the exception that the final range

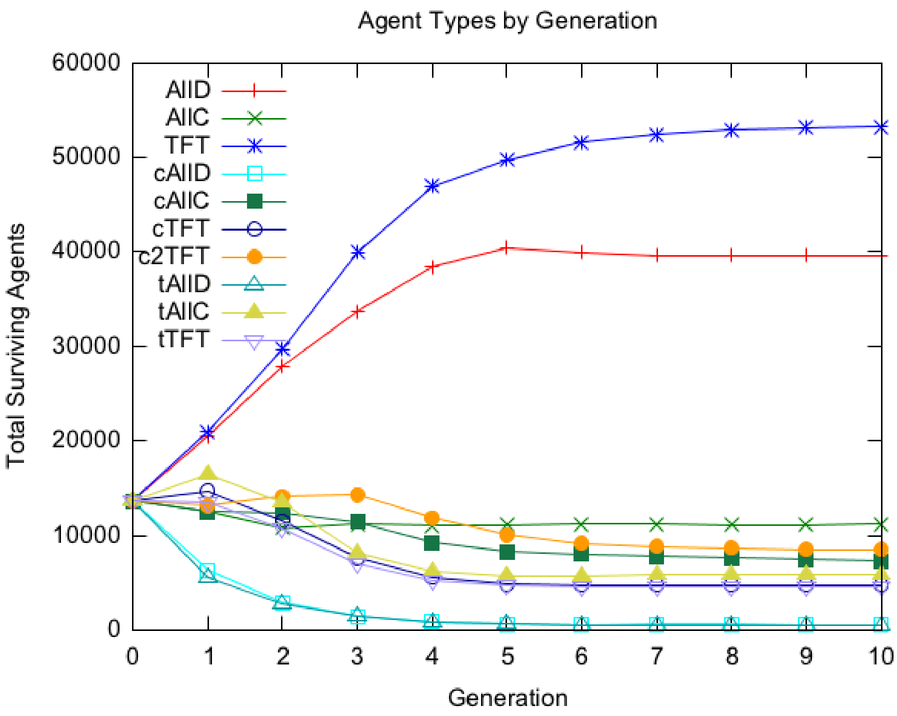

Fig. 2: The numbers of agents, by agent type, alive at the beginning of each generation, summed across all histories.

is a closed interval. ccGiven and ccGotten are not identical for a particular agent, but since the figure plots values for groups of histories, ccGiven and ccGotten are exactly the same values (every commitment given in a history is received in that history). The redundant ccGotten is not included in the figure.

The most influential parameters are ccGiven, endAtNE, investment (i) and cost (c). As the number of commitments exchanged approaches 1, the number of successful histories rises dramatically to 53%. This correlation might be expected as agent types that request and grant commitments are most likely to be *C&S*.

Interestingly, small values of endAtNE are correlated with more successful histories. Small values of endAtNE mean relatively few games ended at a Nash equilibrium. This suggests commitments have positive value in Prisoner's Dilemma-like games where the best outcomes are not at Nash equilibria.

When $i = 0$, C&S agents dominate in 48% of the histories. However, as i increases, C&S agent types dominate less frequently. By $i = 0.4$, C&S agents do not survive at all. This implies that commitments must be inexpensive for debtors to create, otherwise C&S agent types will wither. Therefore, creditors should expect to pay a high price to compensate debtors for any investment costs.

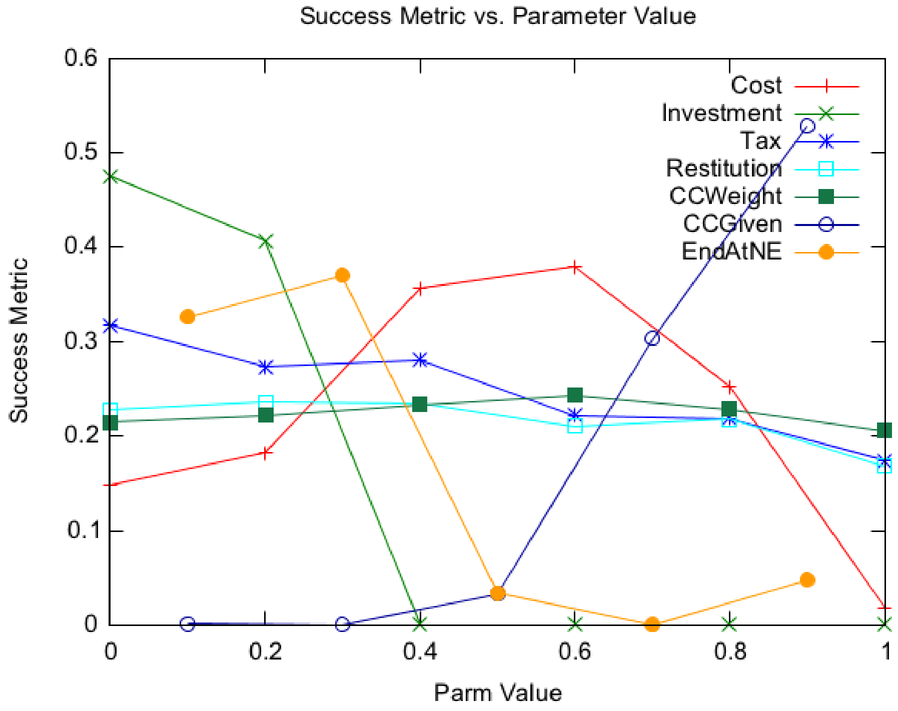

Fig. 3: Each graph shows one parameter (x-axis) vs. the number of histories (y-axis) that resulted in a C&S agent type dominating.

The cost of a given good (c) also strongly influences the conditions in which C&S agents dominate. The optimal utility for a given good should be between 40% and 60% of the utility of the received good. If the cost of the good is too low, there is little benefit to be gained by a creditor from having a commitment from a debtor. And if the cost of the good is too high (nearly equal to the benefit gained) there is great temptation to DEFECT. [Note that a single history can have both $i = 0$ and $c = 0.6$; there is no conflict between these values as might be suggested by the graph.]

Because there is only a small variation in the number of successful histories for different values of tax (t) or restitution (r), these results demonstrate a slight decrease in the success of C&S agent types . The value of commitments to creditors is not as pronounced as expected.

Given the generally flat line for contracts (w or CCWeight), there is no evidence that contracts are significantly more effective over commitments—a possible topic for future work.

5.4 Classifier Results

The factor weights of the classifier are shown in Table 1. The classifier was trained with gradient descent using a regularization parameter $\lambda = 0.3$ which was manually chosen as the largest value that gave the highest observed accuracy of 95.57%. Out of the 4536 total histories, C&S agent types dominated in 1088 histories. The classifier predicts 1119 C&S histories.

Table 1: The factor weights for the classifier that predicts which histories result in *C&S* agent types dominating.

constant	5.3063
cost (c)	-4.9910
tax (t)	0.0534
restitution (r)	1.1126
investment (i)	-21.4269
ccWeight (w)	-0.1065
pdCheckOrder	-0.0415
pdCheckAverage	1.6375
NECount	-0.3165

Investment (i) has the largest weight by absolute value. C&S agent types are more likely to dominate as investment decreases. Cost (c) has the next largest weight by absolute value. C&S agent types are more likely to dominate as cost decreases, and as the benefit/cost gap ($b - c$) increases. pdCheckAverage has the next largest weight. C&S agent types are more likely to dominate, as pdCheckAverage increases. This indicates that the second Prisoner's Dilemma condition Equation 3 is much more predictive of success than Equation 2. C&S agent types seem to favor Prisoner's Dilemma type conditions.

Finally, the weights for the remaining features are small in comparison. Restitution (r) had a larger effect than the insignificant effect of Tax (t). I originally assumed if commitments had a positive effect, then contracts (two reciprocal commitments) would also have a positive effect, but these simulations don't show that, as the weight for contracts (w) is small. And contrary to my intuition, the number of Nash equilibria (NECount) had little effect.

6 Discussion

The primary focus of this work is to characterize the game matrix parameters in which commitments are beneficial. Results show that C&S agent types dominant most frequently (1) when commitments are frequently exchanged, (2) when games tend to end at other than a Nash equilibria, (3) when the cost to create a commitment is low and, (4) when the utility of a given good is about 40% to 60% of the utility of a received good.

I trained a classifier, with high accuracy, to predict when commitment-based (C&S) agent types will dominate a society. Again, results show that the C&S agent types do better when the investment (i) decreases and when the utility of a given good (c) decreases.

As a secondary focus, I consider how an individual agent might use these results. In the beginning of a game series, an agent knows only the game matrix payoffs G, and nothing about its competitors in the initial society. It only discovers opponent information in the course of the game series. My experimental results are strictly valid only for initial societies of the ten, equally represented, agent types in Section 4.1. Societies with different agent types or different agent type mixtures could produce different values of the success metric $\mu(p = v)$. But since my initial society contains a mixture of both competitive and cooperative agent types, these result provide at least an approximation to what an agent might expect to encounter in other situations. They provide some useful information as to whether an agent's strategy should include the use of commitments.

An earlier, experimental version of the Committer's Dilemma game enabled players to cooperate or defect right away, allowing a player to postpone his decision, waiting to see if its counter-player would act quickly. This severely damaged the essence of the Prisoner's Dilemma game where both player must act simultaneously. The current, four phase design eliminates this problem. Future work could consider variations and refinements of the Committer's Dilemma game.

Commitment devices are an existing approach for players to make *credible* commitments to each other. A commitment (or threat) is not credible if the committer can—and likely will—revoke the commitment at a later point. Two kinds of commitment devices are often considered: those where an agent makes a commitment to influence it's own future actions (such as humans trying to diet), and those where an agent tries to *strategically* create commitments to modify the counter-agent's future actions [7]. My approach is related to the latter, but not the former. Commitment devices are another area of interest for future work.

I am interested to see how the results change as new agents types are included in the society. An Axelrod-like [2] contest where (human) participants submit agent types is an interesting possibility.

I have an prototype agent implementation based on a fixed, finte state machine (FSM), where transition probabilities evolve via crossover and mutations. Future work will exploit this capability in experiments that evolve C&S agent types. However, a new mechanistic definition of *C&S* will be needed to classify newly evolved agent types.

References

1. Gnu octave, https://www.gnu.org/software/octave/
2. Axelrod, R.: The Evolution of Cooperation. Basic Books (1984)
3. Baldoni, M., Baroglio, C., Marengo, E.: Behavior-oriented commitment-based protocols. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI). pp. 137–142 (2010)
4. Gerard, S.N., Singh, M.P.: Formalizing and verifying protocol refinements. ACM Transactions on Intelligent Systems and Technology (TIST) (2013)

5. Gerard, S.N., Telang, P.R., Kalia, A., Singh, M.P.: Positron: Composing commitment protocols. In: Proceedings of the First International Workshop on Engineering Multiagent Systems (EMAS). pp. 1–16. St. Paul, MN (2013)
6. Marengo, E.: 2CL Protocols: Interaction Patterns Specification in Commitment Protocols. Ph.D. thesis, Universitá Degli Studi di Torino (Feb 2013)
7. Marks, R.E.: Credible commitments (2000), www.agsm.edu.au/bobm/teaching/SET/week8.pdf
8. Marks, R.E.: Playing games with genetic algorithms. In: Chen, S.H. (ed.) Evolutionary Computation in Economics and Finance. Springer Verlag (2002)
9. Miller, T., McGinnis, J.: Amongst first-class protocols. In: Proceedings of the 8th International Workshop on Engineering Societies in the Agents World (ESAW 2007). LNCS, vol. 4995, pp. 208–223. Springer (2008)
10. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1998)
11. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming. http://lulu.com (2008)
12. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University, 1 edn. (2009)
13. Singh, M.P.: An ontology for commitments in multiagent systems: Toward a unification of normative concepts. Artificial Intelligence and Law 7(1), 97–113 (Mar 1999)
14. Yolum, P., Singh, M.P.: Commitment machines. In: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001). LNAI, vol. 2333, pp. 235–247. Springer, Seattle (2002)