Discrete Gift Exchange Game: Effects of Limited Choices on Opponent Interaction

Steven Damer
University of Minnesota
damer@cs.umn.edu

Maria Gini
University of Minnesota
gini@cs.umn.edu

Jeffrey S. Rosenschein
The Hebrew University of Jerusalem
jeff@cs.huji.ac.il

ABSTRACT

When an agent interacts with an opponent there are many factors that make it difficult to determine the appropriate action to take. The outcomes of the possible actions may be uncertain because of hidden information or simultaneous actions by the opponent. The value of a game state may be uncertain because of the complexity of analyzing the game or because of uncertainty about how the opponent will act in the future. Finally, in general-sum games, the agent is uncertain about how its own actions will affect the opponent's future play. The Gift Exchange game [10] has been developed to study the problem of how an agent's actions can affect the opponent's future actions. In the Gift Exchange game the action space is continuous; an opponent's intentions can be observed from the action they choose. In this paper we will explore the effect of discretizing the action space of the Gift Exchange game. We will describe the discretized Gift Exchange game and discuss how to learn an optimal response to simple opponents. Then we will show how the proper selection of a simple strategy can take advantage of a learning opponent. We will present experimental results showing the effect of various parameters on the optimal strategy.

KEYWORDS

Repeated sequential games; cooperation; non-stationary opponents

1 INTRODUCTION

Interacting with other self-interested agents is an important focus of multiagent systems research [4, 6, 8, 19, 22, 23]. When deciding how to act, it is necessary to calculate the immediate effects of each action, the effect of each action on the game state, and the effect each action will have on future behavior of the opponent. One approach is to treat each interaction with the opponent independently; no-regret learning does this, and is able to achieve a payoff at least as good as the performance of the optimal constant strategy against the observed sequence of opponent moves [7]. However, this approach does not take into account the possibility that the agent's moves may affect future choices of the opponent. For example, in the repeated Prisoner's Dilemma the no-regret strategy is to constantly defect [18], which performs poorly against strategies like Tit-for-Tat.

On the other hand, if the agent's actions may affect future opponent actions, it becomes considerably more difficult to determine how to play. One approach would be to form a prior distribution over some set of possible opponents, and play optimally against that prior distribution; if the possible opponents are described as finite state machines, this is equivalent to solving a Partially Observable Markov Decision Process (POMDP) [20]. However, specifying that prior distribution and solving the problem itself are computationally prohibitive for reasonably complex opponents.

Another way to play is to calculate a Nash equilibrium of the game. If we assume a fixed number of repetitions, induction will often lead to an equilibrium without cooperation, but if we assume an indefinite number of repetitions the Folk Theorem tells us that any individually rational outcome can be supported as an equilibrium. These problems can be solved by augmenting the game definition with discount factors and priors for the players, but again this would be computationally expensive and difficult to specify.

In this paper we will look at strategies that take advantage of the performance guarantees of the opponent's strategy instead of attempting to find an equilibrium or explicitly model the opponent. We develop strategies for a game which is explicitly constructed to trivialize the problems of determining the outcome of the agent's actions or the intended outcome of the opponent's actions.

Our environment is a *Gift Exchange game* [10] in which two players take turns selecting outcomes, which provide payoffs (which may be negative) to each player. There are several important properties of this game. First, interactions are isolated; a player's choice of action completely determines the immediate payoffs to the player and its opponent, but has no further effect on the state of the game. This means that agents do not need to track a game state; prior interactions are only relevant in terms of how they affected the agent's model of the opponent, and the agent's only concern when selecting an action is the immediate payoff of the action and the effect of the action on the opponent's state. Second, players take turns acting, instead of acting simultaneously. This simplifies the analysis because it means that players have complete information about the state of the game when they act; there's no need to model what the opponent is currently doing. Finally, payoffs are observable—there is no secret information for either player.

The main contribution of this paper is an extension of the Gift Exchange game to discretize its action space and an analysis of strategies for playing the discrete version of the game against different types of opponents.

2 RELATED WORK

There is a large amount of work in multiagent systems on cooperation in general, and specifically on choosing one's actions to affect an opponent's future actions. In this section we provide a brief survey of that work.

The Gift Exchange game [10] is most similar to the Dictator [15] game. Both games involve the active player choosing an outcome

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

that cannot be affected by the opponent; however, the Gift Exchange Game involves an element of social dilemma that is not usually present in the Dictator game, and the Dictator game is not usually a repeated game. In the Dictator game, one player is given an initial endowment, which they may then choose to divide with the other player in any proportion they choose (including the option of keeping the entire endowment for themselves). The game-theoretic analysis of the Dictator game is trivial—the dictator keeps the entire endowment. People playing the dictator game do not generally play the Nash equilibrium. The Dictator game has been extensively studied to explore the factors that influence how people play it. The Dictator game is generally played as a single shot game, but one study has explored the effects on the second player when two games are played in a row with players swapping roles [11]. In that situation the second player is more likely to return the gift received from the first player. Other factors that have been studied include demographics, cultural factors, the value of the endowment, social distance between the players, and how deserving the recipient is. Attention has been focused on measuring how these factors affect human play [13]. In contrast, our focus in the Gift Exchange Game is on exploring reciprocation as a basis for non-equilibrium play.

The Automated Negotiating Agents Competition [3] pits negotiation agents against one another. In each match two or more agents take turns proposing outcomes from a space of possible outcomes, which is selected on a per-match basis. If agents do not agree on an outcome before the deadline is reached, each agent receives its reserve value. In addition, in some matches there is a discount factor to encourage agents to reach an agreement more rapidly. Unlike in our environment, agents do not know their opponent's utility function—to play the game, they must simultaneously attempt to estimate their opponent's utility for each of the potential outcomes and their opponent's willingness to cooperate or make concessions. Most agents designed for this competition focus on opponent modeling; they attempt to predict which offers their opponent will accept. In competition it has been found that the most successful agents are generally tougher negotiators. This is a consequence of the fact that agents generally reach some agreement (which suggests that most agents are not too rigid). If the community of agents frequently failed to reach an agreement, then more generous agents would be favored. This game is more suited to negotiation than the Gift Exchange Game because only the final offer has an effect on the payoffs received by the agents. In the Gift Exchange Game, being a tough negotiator imposes an immediate opportunity cost, as the agent forgoes the chance to cooperate in that round, and may need to pay a cost to punish the opponent for rejecting the agent's desired outcome.

Randomization can be used to generate strategies for the Repeated Prisoner's Dilemma [16] which confine the outcome of the game to a bounded region. The authors show how this approach can be successful in a tournament, and also show good performance of this strategy against a reinforcement learner. We use a similar approach to achieve good performance against a learning opponent.

Instead of treating the problem of choosing a strategy in repeated normal form games as a sequence of decisions, in [27] it is framed as one player selecting a finite automaton to play for them, and the other player selecting an automaton in response. The size of the automata can be used to represent the bounded rationality of the players. The paper describes how to compute an optimal automaton for the first player to commit to. This is similar (but more complex) to the strategies we have developed to play our Gift Exchange game. We handle the problem of selecting an optimal strategy by introducing a discount factor, while they handle it by introducing limits on the complexity of the automata.

If the opponent is playing a finite automaton in a repeated normal form game, the agent generally does not know the characteristics of the automaton. It is possible to learn a stationary strategy for the opponent using reinforcement learning, however the opponent may not stick to a stationary strategy. This problem can be handled using the R-MAX# algorithm [17] which can detect when the opponent has deviated from the learned strategy and learn to best respond to the new strategy. The strategies we have developed for the Gift Exchange game are intended to take advantage of strategies like R-MAX# [5]; it is easy to find the best response, but the best response to the strategy is beneficial for the agent.

A modification of reinforcement learning is described in [14]; the agent updates its policy to optimize performance against a learning opponent instead of a static opponent. The authors show that agents using this update rule are capable of learning to cooperate in the Repeated Prisoner's Dilemma. Furthermore, they show that when the technique is applied again (i.e., the agent optimizes its policy under the assumption that the opponent is updating *its* policy under the assumption that the agent is a naive reinforcement learner), it does not result in additional gains. In our work we do not explicitly model the learning that agents do as they converge to a cooperative outcome, but [14] suggests that it could be a successful approach.

An algorithm for repeated stochastic games, presented in [12], uses lossy game abstraction [24] to reduce the state space of the game and facilitate learning and adapting rapidly to a non-stationary opponent. The algorithm in [9] reduces the problem to a multi-armed bandit problem by generating a handful of expert strategies to use in the repeated stochastic game. This approach simplifies the underlying game to make it a matter of selecting the appropriate expert strategy. The agent selects a strategy with the intent that the opponent will play its part in the selected strategy, and enforce compliance by punishing the opponent when it fails to comply. The work on repeated stochastic games focuses more on the problem of how to cooperate, rather than whether to cooperate.

To study cooperation in Markov games, in [21] a pair of games are presented that have opportunities for cooperation, and a Deep Q-Network is used to learn strategies for those games. By varying parameters of the game, different strategies are learned, which can be designated *cooperate* or *defect* according to the performance of the strategy in self-play. The strategies developed implement cooperation or competition in the underlying Markov game, but they do not attempt to reciprocate. This work approaches the problem of cooperation from the other end—instead of starting from a simple environment and looking at how to decide when to cooperate, they start from a complex environment and look at learning how to cooperate in the first place.

We have focused on agents interacting with opponents whose goals are orthogonal to the agent's goals. It is also interesting to consider the problem of how to interact with an opponent that shares the goals of the agent, but has not been designed to coordinate with the agent [25]. If the opponent is able to learn by observing the

agent, the agent must decide how to balance acting to further its goals versus acting to teach the opponent a better strategy. This problem is considered in two different contexts : a repeated normal-form game (with both players sharing the same utility function) in which the agent must find the most effective teaching sequence to optimize performance against a bounded memory best-responding opponent, and a shared multi-armed bandit problem where the agent must decide when to pay a cost to demonstrate the optimal choice to the opponent. The problems are similar to the problem faced by an agent in the Gift Exchange game that is attempting to manipulate its opponent—interacting with the opponent is simpler, because the agent and the opponent have a shared utility function, but the environment is more complex than the Gift Exchange game.

A recent survey [1] covers work done across a wide spectrum of multi-agent systems in modelling opponents. We refer the reader to it for approaches we have not covered here.

3 DISCRETIZING THE GIFT EXCHANGE GAME

In the Gift Exchange game, agents take turns choosing actions, where each action consists of a choice of outcome from a set of potential outcomes. Each outcome is an assignment of (potentially negative) payoffs to the agent and its opponent. In previous work [10] the set of potential outcomes is the unit circle, where one player receives the x-coordinate of the chosen point and the other player receives the y-coordinate of the chosen point. This set of choices allows for costly cooperation and places no limitations on the ratios between payoffs that can be achieved in a single choice. However, in many environments this level of precision is not available. The Gift Exchange game can be easily modified to present agents with a discrete set of choices, which changes the dynamics of agents attempting to learn opponent models or take advantage of a learning opponent.

Figure 1 shows a sample choice set for a discrete Gift Exchange game. The options available for players are:

- A The most beneficial option for the agent
- B The most beneficial option for the opponent
- C A cooperative outcome
- D The most punishing option for the agent
- E The most punishing option for the opponent
- F A strictly competitive option for the agent
- G A strictly competitive option for the opponent

The space of achievable outcomes is shown by the gray lines. The space of pareto-optimal outcomes is shown by the green lines. Note that there are no options between D and E because neither player would choose an action that is better for their opponent and worse for them than the maximally punishing option for their opponent.

Figure 2 shows a discrete Gift Exchange game with a much smaller choice set. Each player has a preferred choice, and there is a mutually punishing option as well. Unlike the first sample game in Figure 1 there are no benefits to mutual cooperation, but the punishing option creates the possibility for players to threaten their opponent to get them to play their preferred option.

Any combination of points can form a set of possible payoffs for a Gift Exchange game as long as they form a convex hull. A point in the interior of the convex hull of possible payoffs would not be a rational choice for either player because regardless of that players intent, they could fulfill it more effectively by choosing from points on the convex hull.

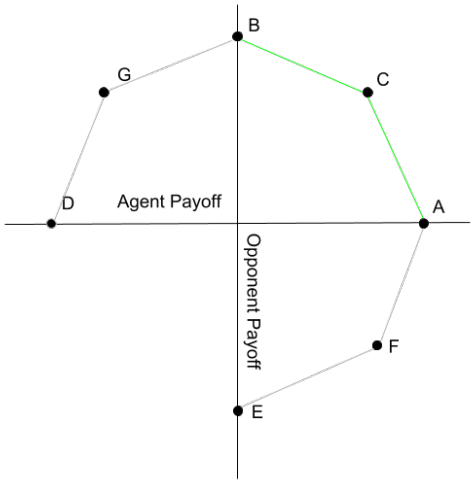

Figure 1: A sample choice set for a discrete Gift Exchange game. The x-axis is the agent's payoff and the y-axis is the payoff of its opponent.

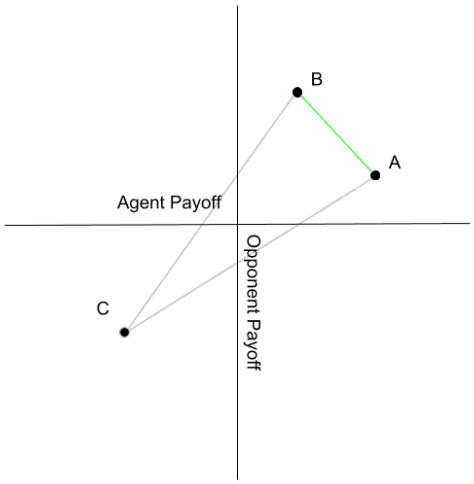

Figure 2: A choice set for a discrete Gift Exchange game with more limited options. The x-axis is the agent's payoff and the y-axis is the payoff of its opponent.

3.1 Formal Definitions

A **discrete Gift Exchange game** is described by a set of two players $P = \{A, B\}$ and a choice set $U = \{(u_A, u_B) \in \mathbb{R}^2\}$ where the points in U form a convex hull.

In round i the current player chooses an outcome $c_i = (c_{i,A}, c_{i,B}) \in U$ with player A choosing in odd numbered rounds and player B choosing in even numbered rounds. After each choice $(c_{i,A}, c_{i,B})$

player A receives payoff $c_{i;A}$ and player B receives payoff $c_{i;B}$. Note that these payoffs may be negative, in which case they represent a loss to that player. The game may be played for a fixed or an indeterminate number of rounds. By convention we will assume the agent is player A and the opponent is player B.

We will occasionally use the discretized perimeter choice set U for examples. U is a discretization of points on the perimeter of the unit circle. For a given n divisible by 4, U_n contains all points of the form $\langle \sin 2\pi \frac{i}{n}; \cos 2\pi \frac{i}{n} \rangle$ where i is any integer from 1 to n. For any U_n, the maximally rewarding strategies for A and B will be $\langle 1; 0 \rangle$ and $\langle 0; 1 \rangle$ respectively, and the maximally punishing strategies will be $\langle -1; 0 \rangle$ and $\langle 0; -1 \rangle$ respectively.

We define a history as a sequence of outcomes, $h = \{h_1 : : :\}$ with $h_i \in U$, where h_i is the choice made in round i, $h_{i;A}$ is the payoff assigned to player A and $h_{i;B}$ the payoff assigned to player B. We define $h_{:i}$ as the subsequence of choices in h up to and including round i and h_{-1} as the final choice in h for histories with finite length. We define H as the set of all possible histories, with $H_A \subseteq H$ as the set of all possible histories of even length (including the empty history), and $H_B \subseteq H$ as the set of all possible histories of odd length. A will be the next player to choose an outcome after histories in H_A and B will be the next player to choose an outcome after histories in H_B.

A strategy s is a function that maps histories to outcomes; $s\langle h \rangle \in U$ is the outcome chosen by a player following strategy s after observing history h. The set of all strategies for player A is $S_A : H_A \to U$, the set of functions mapping H_A to U. Similarly, S_B is the set of all strategies for player B. The combination of strategy $s_A \in S_A$ for player A,and strategy $s_B \in S_B$ will produce a specific history $Outcome\langle s_A; s_B \rangle \in H$ with the property that $h_i = s_A\langle h_{:i-1} \rangle$ when i is odd and $h_i = s_B\langle h_{:i-1} \rangle$ when i is even. We refer to generic strategies $s \in S_A \cup S_B = S$ when we do not wish to specify to which player we are referring. Strategies that include randomization can be represented as a probability distribution over S_A or S_B. The outcome of two randomizing strategies is a distribution over H.

A strategy s is a *constant* strategy if $s\langle h \rangle = c \in U$ for some constant c. A strategy is *non-reactive* if the choices it makes do not depend on the past actions of the opponent; a strategy s is non-reactive if $len\partial th\langle h \rangle = len\partial th\langle h' \rangle = \rangle \; s\langle h \rangle = s\langle h' \rangle$. Note that a non-reactive strategy is not necessarily stationary—it can change, just not in response to opponent choices. A *reactive* strategy is one that conditions its choices on the choices made by the opponent in the past. Reactive strategies can be divided into those that depend only on the most recent action taken by the opponent and those that depend on the entire history of play. A strategy that only depends on the most recent action is *immediately reactive*; s is immediately reactive if $s\langle h \rangle = s\langle h' \rangle$ whenever h and h' have the same length $len\partial th\langle h \rangle = len\partial th\langle h' \rangle$ and the same last move $h_{-1} = h'_{-1}$. A *randomizing immediately reactive strategy* makes its choice at random according to a distribution determined by the opponent's last choice. A *fixed immediately reactive strategy* does not use randomization.

The payoff of a history through time t can be described as the sum of payoffs of the moves $Payoff_t\langle h \rangle = \langle \sum_{i=1}^{t} h_{i;A}; \sum_{i=1}^{t} h_{i;B} \rangle$. The average payoff is $\overline{Payoff}_t\langle h \rangle = \langle \frac{\sum_{i=1}^{t} h_{i;A}}{t}; \frac{\sum_{i=1}^{t} h_{i;B}}{t} \rangle$. In games played for an indefinite period, where the stopping point is unknown or there is no stopping point, it is trickier to evaluate performance. Clearly the sum of the payoffs can diverge, so we will generally use the limit of the average payoff or the average discounted payoff. The limit of the average payoff can be easily calculated as $\overline{Payoff}_{-1}\langle h \rangle = \lim_{t \to \infty} \overline{Payoff}_t\langle h \rangle$, but note that there are histories for which that limit does not converge. An example of a non-converging history is one that switches between playing $\langle -1; 1 \rangle$ and $\langle 1; 0 \rangle$ depending on whether $\lfloor \log_{10} i \rfloor$ is even or odd where i is the round. For this history average payoffs will oscillate between $\langle -:9; :1 \rangle$ and $\langle -:1; :9 \rangle$ and never converge. Combinations of strategies that produce histories for which the limit does not converge are not generally well-justified, as they involve cycling between different choices infinitely often which is not Pareto-optimal. The average discounted payoff is $\overline{Payoff}^{\delta} = \lim_{t \to \infty} \langle (1-\delta) \sum_{i=1}^{t} \delta^{i} h_{i;A}; (1-\delta) \sum_{i=1}^{t} \delta^{i} h_{i;B} \rangle$. The discount factor, δ, describes the degree to which the agent discounts future payoffs. It is often used to describe a situation in which there is a fixed probability, δ of the game ending after the current round.

4 PLAYING AGAINST IMMEDIATELY REACTIVE OPPONENTS

One advantage of the discrete version of the Gift Exchange game is that it is easier to define specific classes of opponents and the best responses to them. In this section we will discuss best responses to opponents that only consider the previous choice of the agent when making their choice under a variety of circumstances.

Opponents that are non-reactive are trivial to best-respond to. The best response to a non-reactive opponent is to always select the choice that maximizes the agent's payoff.

Theorem 4.1. *Given a non-reactive strategy $s_b \in S_B$, the strategy $s_a\langle h \rangle = \arg\max_{u \in U} u_A$ maximizes the payoff of player A.*

Proof. Let $h = Outcome\langle s_a; s_b \rangle$. The payoff of player A through round t is $Payoff_{t;A}\langle h \rangle = \sum_{i=1}^{t} h_{i;A}$. This can be split into the rounds where player A chose and the rounds where player B chose $\sum_{i=1}^{t \cdot 2} h_{2i-1;A} + \sum_{i=1}^{t \cdot 2} h_{2i;A}$. which is equal to $\sum_{i=1}^{t \cdot 2} s\langle h_{2i-2} \rangle_A + \sum_{i=1}^{t \cdot 2} s\langle h_{2i-1} \rangle_A$. The first sum is maximal by the definition of s_a and the second sum is constant because s_b is a non-reactive strategy.

If the opponent is playing a fixed immediately reactive strategy, we consider these different goals:

Maximize average payoff with no time horizon. If the goal is to maximize the average payoff with no time horizon ($\overline{Payoff}_{-1}\langle h \rangle_A$), the agent should make every choice once to learn the opponent strategy s^B, and then play the constant strategy that gives the best payoff $s\langle h \rangle = \arg\max_{u \in U} u_A + s^B\langle u \rangle_A$. The order in which options are checked does not matter because the goal is to maximize average payoff over an infinite time horizon.

Maximize total payoff over a fixed horizon. If the opponent is immediately reactive, and the goal is to maximize the total payoff over some fixed time horizon that is much longer than the total number of options available ($t >> |U|$), then the optimal strategy is similar, except that options should be checked in descending order of u_A, and no further options

should be checked when the payoff of choosing the current best option for the rest of the game is greater than the maximum possible payoff of the remaining unchecked options. Let $s_i : U_i^0 \to U$ be the incomplete function learned from observing the opponent's behavior up to round i where U_i^0 is the set of choices for which responses have been observed up to round i. Let u_i^{max} be the best observed option in round i, $u_i^{max} = \arg\max_{u \in U_i^0} u_A + s_i(u)_A$

Theorem 4.2. *The optimal sequence of choices* $\{u_1, u_2, \ldots, u_n\}$ *to maximize the total payoff assuming a uniform distribution over immediately reactive opponents is in descending order of preference* $u_{i,A} \geq u_{i+1,A}$.

Proof. Sketch. Consider any two sequences that are identical except for two adjacent choices, $C = \{u_1, \ldots, u, u', \ldots u_n\}$ and $C' = \{u_1, \ldots, u', u, \ldots u_n\}$. Assume without loss of generality $u_A > u'_A$. When neither u nor u' is the maximizing choice, the expected payoff of ordering guesses according to either sequence is identical. When one of u and u' is the payoff maximizing choice, the expected payoff of guessing u first is higher because when the agent guesses u first the immediate payoff is higher, and the agent is able to rule out more unchecked options. Therefore, the only sequence for which the expected payoff cannot be improved is one in which options are checked in descending order of preference.

Maximize discounted average payoff. If the opponent is immediately reactive and the goal is to maximize the discounted average payoff, then the agent should guess in descending order of expected value $E[u_A + s(u)_A]$, but stop guessing when the expected payoff of guessing the next value is lower than the value of choosing the best option found so far for the rest of the game $\frac{u_{i,A}^{max} + s(u_i^{max})_A}{1-\delta} > u_{i+1,A} + \delta E[s(u_{i+1,A})] + \frac{\delta^2}{1-\delta} E[max(u_{i,A}^{max} + s(u_i^{max})_A, u_{i+1,A} + s(u_{i+1})_A)]$ where δ is the discount factor of the agent. Note that the expectation over opponent strategies can be uniform, but is not required to be.

All the strategies described up to now have assumed that the opponent is playing a fixed immediately reactive strategy. However the opponent may be using randomization. A randomizing immediately reactive opponent will respond to an agent choice u by drawing from a random distribution over immediately reactive strategies, which is functionally identical to a multi-armed bandit. Each choice of outcome u represents a separate arm, with a payoff of $u_A + E[s(u)]$ where s is drawn from the opponent's distribution over strategies. A well-known algorithm for multi-armed bandits is UCB [2], which we will use in this paper to play against randomizing immediately reactive strategies.

5 TAKING ADVANTAGE OF LEARNING OPPONENTS

There are a number of ways to take advantage of the strategies described in the previous section that find a best-response to an immediately reactive opponent. One method is to violate the assumptions of the learning strategies which assume a fixed immediately reactive opponent. An agent can take advantage of those strategies by playing its initial responses to deceive the opponent and then, once the opponent has settled on what it thinks is the best response, switching to playing the choice that maximizes its own payoff. A more interesting way to take advantage of learning opponents is to play a strategy which follows the assumption of the learning strategy, but is structured such that the best response to that strategy is beneficial to the agent. Essentially, such a strategy is using the constraints on the agent's behavior to gain an advantage over the opponent in a manner similar to a Stackelberg [26] leader.

If the opponent assumes the agent is playing an immediately reactive strategy, we can consider these different cases:

Maximize average payoff vs. fixed strategy: If the opponent is attempting to maximize its average payoff, then the best immediately reactive strategy for the agent is the one that maximizes the agent's payoff subject to the constraint that the opponent's payoff is greater than the maximum amount it can guarantee itself. Let $u_B^{max} \in U$ be the choice that maximizes the opponent's payoff, and $u_B^{min} \in U$ be the choice that minimizes the opponent's payoff. Let $u, u' \in U$ be the choices that maximize $u_A + u'_A$ subject to the constraint that $u_B + u'_B > u_B^{max} + u_B^{min}$. Depending on the choices in U it is possible that $u = u'$; for example, if $u_B^{max} + u_B^{min} = 0$ and $U = \{(1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-.7, .7), (.7, -.7), (.7, .7)\}$ then u and u' will be $(1, 0)$ and $(.7, .7)$, but if we change $(1, 0)$ to $(1, .1)$ then $u = u' = (1, .1)$. u is the choice that the agent would prefer; u' is the offer that the agent needs to make to ensure that the opponent is better off accepting the deal. Define the agent's optimal strategy s as $s(u) = u'$ and $s(u') = u$; for any other choice $v \in U$ let $s(v) = \arg\max_{v' \in U} v'_A$ subject to the constraint that $v_B + v'_B < u_B + u'_B$. s is a best response to any opponent learning strategy that assumes the agent is playing a fixed immediately reactive strategy.
As an example, consider the optimal fixed immediately reactive strategy for an agent playing a Gift Exchange game with discretized perimeter choice set U_n against a learning opponent. The optimal strategy for the agent is:

$$s_n(u) = \begin{cases} (\cos\frac{\pi}{2n}, \sin\frac{\pi}{2n}) & if\ u = (1, 0) \\ (1, 0) & if\ u_B < 0\ or\ u = (\cos\frac{\pi}{2n}, \sin\frac{\pi}{2n}) \\ (u_B, u_A) & if\ u_A < 0 \\ (u_A, u_B) & otherwise \end{cases}$$

This strategy limits the opponent to receiving a payoff of 0 unless it chooses a preferred outcome for the agent in which case it will receive an average payoff of $\frac{1}{2}\sin\frac{\pi}{2n}$.

Maximize discounted average payoff vs. fixed strategy: the agent's best strategy is constructed in a similar manner, except that the preferred choice must be one of the choices the opponent will check.

Maximize average payoff vs. randomizing strategy: If the opponent assumes that the agent is playing a randomizing immediately reactive strategy and is attempting to maximize its average payoff, this allows the agent to improve its performance by using a randomizing strategy. This is because the ability to randomize allows the agent to make choices with an expected payoff on the convex hull of the set of choices,

which allows it to give the opponent a choice even more beneficial to it. Given a choice set U we can construct a randomizing immediately reactive strategy to take advantage of a learning opponent as follows. Let $u_B^{max} + u_B^{min}$ be the amount the opponent can guarantee itself by always playing u_B^{max}. Let u be the choice that maximizes u_A subject to the constraint that $u_B > \frac{u_B^{max}+u_B^{min}}{2}$. Let u' be the choice adjacent to u on the convex hull that maximizes u_A; if both neighbors of u have a lower payoff for the agent, then $u' = u$. Pick a probability p such that $p\ u_B + (1-p)\ u_B' > u_B^{max} + u_B^{min}$ Then play the following strategy: if the opponent plays u play u with probability p otherwise play u', and if the opponent does not play u play $\arg\max_{u' \in U} u_A'$ subject to the constraint $u_B + u_B' < u_B(1-p) + u_B'(1-p)$. This strategy will ensure that the opponent's best response is to play u while maximizing the agent's payoff.

Algorithm 1 Simulated Annealing Algorithm

```
generate a population of 10 candidates
σ ← 0.5                                   . Level of noise to add
initialize n                              . Number of choices
while σ > 0.05 do
    for each candidate do
        generate 8 perturbed candidates
    end for
    evaluate the perturbed candidates
    keep the top 10
    σ ← σ × 0.99
end while
return current population
function GENERATECANDIDATE(n)
    Strategies are represented as an array of values
    the integer part indicates the base option to return
    the fractional part indicates the probability of switching
    to the next option
    return an array of random values from 1 to n
end function
function PERTURBCANDIDATE(candidate,σ)
    Add gaussian noise with standard deviation σ to candidate
end function
function EVALUATECANDIDATE
    return Payoff_10000(Outcome(candidate, opponent))
end function
```

When the opponent is maximizing the average payoff against a randomizing strategy, the optimal strategy for the agent gives the opponent a payoff very slightly greater than the amount the opponent can guarantee for itself. The closer the opponent's payoff is to the amount it can guarantee for itself, the better the agent's payoff is. However, the closer the opponent's payoff is to the amount it can guarantee itself, the longer it will take the opponent to learn the optimal response. In infinitely repeated games this is irrelevant because the agent's average payoff will be $\frac{u_A(1-p)+u_A'(1-p)}{2}$, but in finitely repeated games or games with discounting it is significant.

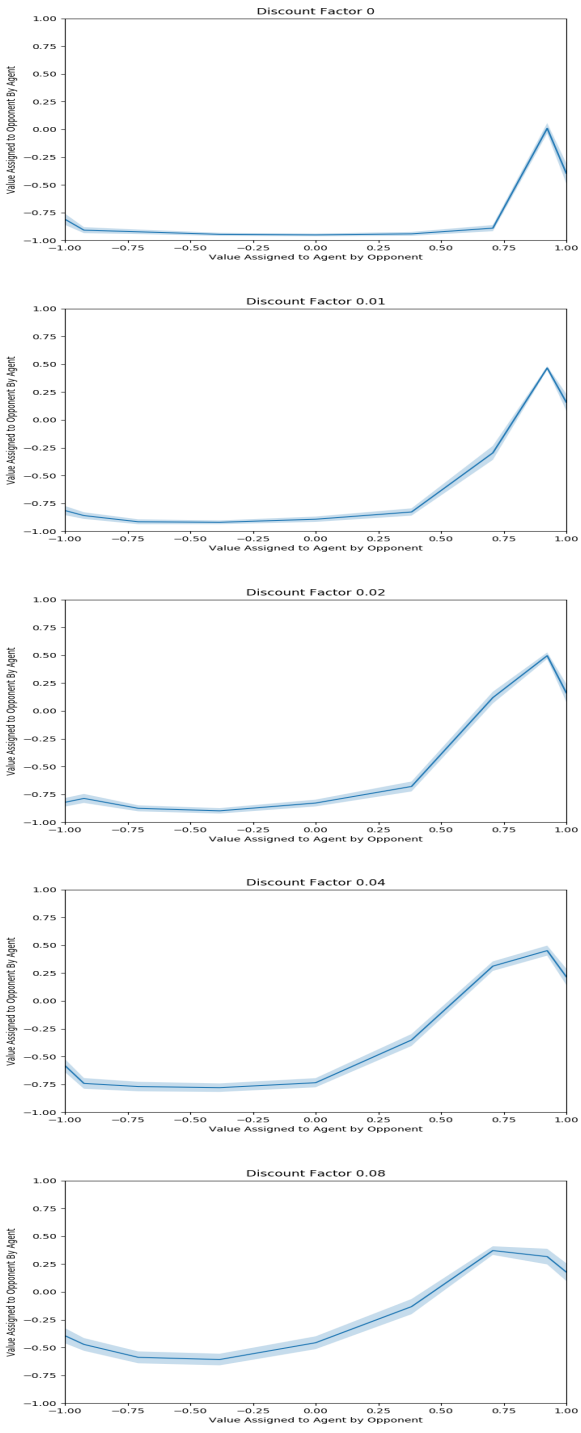

Figure 3: Strategies found using simulated annealing in U_{16} against a UCB opponent. The discount factors are 0.0, 0.01, 0.02, 0.04, and 0.08, from top to bottom. The shaded region shows the 95% confidence interval.

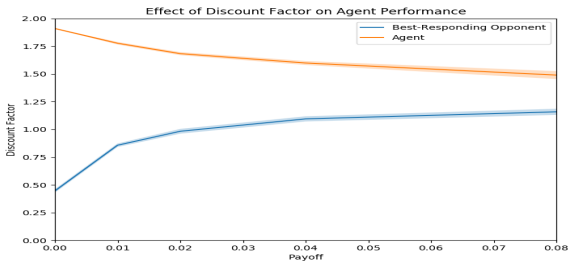

Figure 4: Performance of strategies found using simulated annealing in U_{16} against a UCB opponent as the discount factor varies. The shaded region shows the 95% confidence interval.

In a game that is not infinitely repeated we can use simulated annealing (Algorithm 1) to find the optimal randomizing immediately reactive strategy for a given opponent. A strategy is represented by an array that describes the response to each opponent action. The response of a strategy to a value is given as an index of the value to return plus a probability of giving the opponent the next higher amount. This can represent any randomizing immediately reactive strategy that only randomizes between adjacent options. This limitation is reasonable because randomizing between non-adjacent options is inefficient—it will select points from the interior of the convex hull of possible payoffs. Strategies are also limited to choosing rational outcomes in U_n; they will only assign themselves a non-negative value. We do not restrict strategies more than this because it is not clear that the form of an optimal strategy follows a more strictly defined structure (and results of simulated annealing do not suggest this).

We have used simulated annealing to find effective randomizing immediately reactive strategies for the choice set U . We look at the effects of discount factor, number of choices in the choice set, and the exploration factor of the opponent.

Figure 3 shows how the best-performing strategies vary as they are optimized for different discount factors. When the discount factor is 0 the strategy is extremely punitive and the opponent's best response is to give the agent :923; the agent will occasionally reciprocate with a gift of 0:382 to the opponent. When the discount factor increases, the agent will accept lower amounts from the opponent, and punish non-compliant choices less severely. This occurs because punishment is costly and with a high discount factor the agent is unwilling to incur those upfront costs to coerce the opponent into a better long-term strategy. Figure 4 shows how the performance of the agent and a best-responding opponent is affected by the discount factor. Note that as the discount factor increases, the agent is offering the opponent a nearly equal split of the potential payoff.

Figure 5 shows how the best-performing strategies vary as the number of choices varies. Choices are evenly distributed around the unit circle with 8, 16, or 32 choices. In this case, increasing the number of choices allows the agent to be more precise in its demands. All of the strategies punish at approximately the same level. Strategies with access to more choices are able to be more

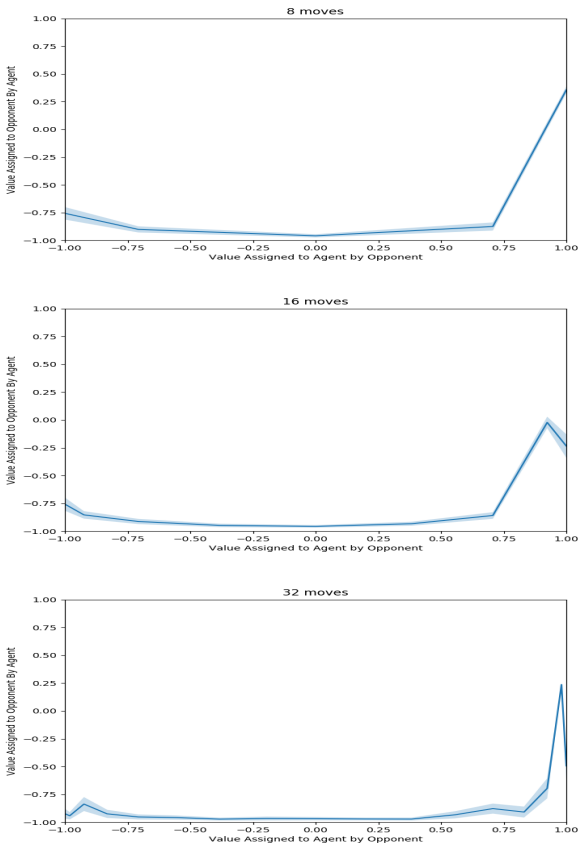

Figure 5: Strategies found using simulated annealing in U_8, U_{16}, and U_{32} (from top to bottom) against a UCB opponent. The shaded region shows the 95% confidence interval.

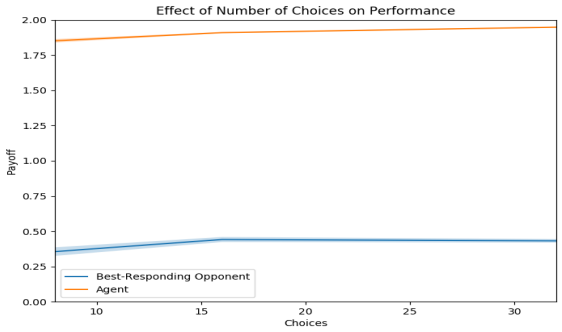

Figure 6: Performance of strategies found using simulated annealing in in U_8, U_{16}, and U_{32} against a UCB opponent. The shaded region shows the 95% confidence interval.

precise in the amount they demand from the opponent, but they all demand about the same amount.

Figure 6 shows how the performance of the agent is affected by the number of moves available. Since the discount factor is 0,

the agent adopts a very greedy strategy. Increasing the number of moves causes the agent's payoff to go up slightly, but the effect is not as large as that of the discount factor. Unlike the other parameters we have looked at, both the agent's and the opponent's payoff increase as the number of moves rises; this implies that the performance increase is due to more efficient cooperation.

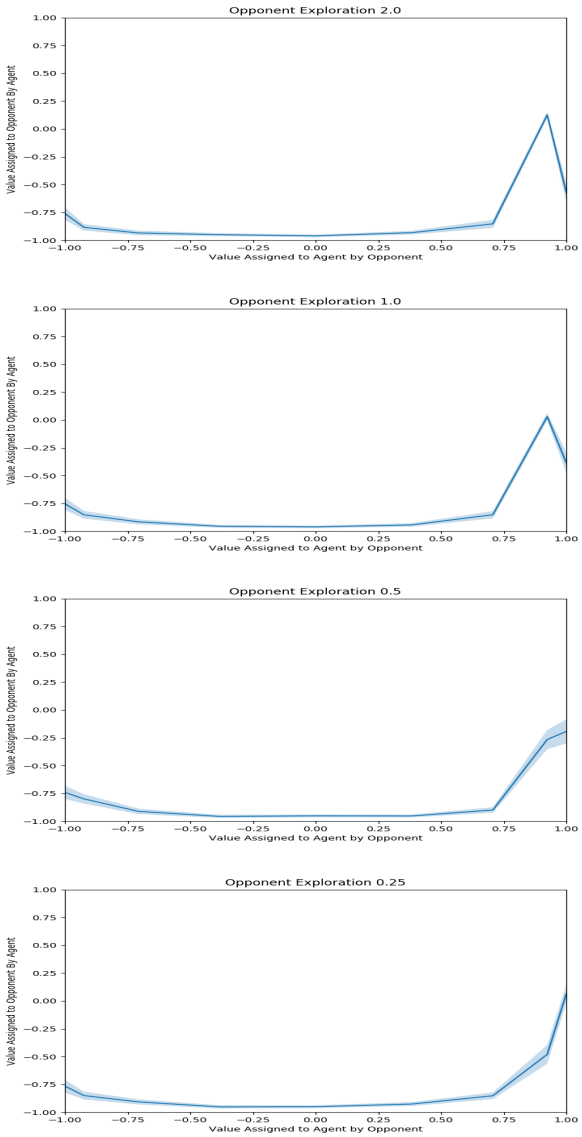

Figure 7: Strategies found using simulated annealing in U_{16} against a UCB opponent with exploration factors 2:0, 1:0, 0:5, and 0:25 (from top to bottom). The shaded region shows the 95% confidence interval.

Figure 7 shows how the best-performing strategies vary as the opponent explores less. The level of exploration has the smallest effect of all the parameters we have looked at. With higher levels, the agent will select costly punitive choices more frequently, so

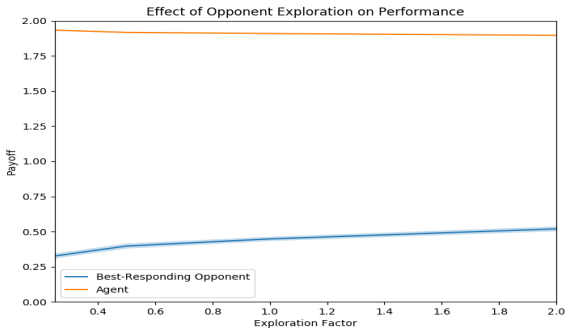

Figure 8: Performance of strategies found using simulated annealing in U_{16} against a UCB opponent as the exploration factor varies. The shaded region shows the 95% confidence interval.

the optimal strategy is slightly more moderate in its demands of the opponent. However, the effect is far smaller than the effect of varying the discount factor of the agent. Figure 8 shows that as the exploration of the opponent increases the performance of the agent decreases while the performance of the opponent increases.

6 CONCLUSIONS

The discretized version of the Gift Exchange game provides a useful structure to explore the problem of encouraging a self-interested opponent to adopt preferred strategies. Although it is more complicated to generate reciprocating strategies than in the continuous version of the Gift Exchange game it is still possible, and the results shown in Figure 6 suggest that a relatively low number of choices is required for the game to provide opportunities to fully explore the possibilities of a reciprocating strategy. Another advantage of the discretized version over the continuous version is that it is easier to define strategies that attempt to learn a best response to the opponent. If the opponent is assumed to be playing a fixed strategy, the best response can be learned after $|U|$ observations, and if the opponent is randomizing the UCB algorithm can be used to play against immediately reactive strategies without further modification. In this paper we showed how to create immediately reactive strategies that are best responses to learning strategies. By varying parameters of the game and the process of evaluation we looked at how they affect the optimal immediately reactive strategy. Increasing the number of choices increases the payoff of both players, suggesting that it allows for more efficient cooperation. Increasing the exploration factor of the learning opponent resulted in strategies that are slightly less greedy, as the cost of punishing the opponent becomes more significant. Increasing the discount factor of the agent had the largest effect, with the opponent receiving almost as much as the agent at high discount factors.

Acknowledgments: This work was supported in part by Israel Science Foundation grant #1340/18.

REFERENCES

[1] Stefano V. Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66 – 95.
[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.
[3] Tim Baarslag, Mark JC Hendrikx, Koen V Hindriks, and Catholijn M Jonker. 2016. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Journal of Autonomous Agents and Multi-agent Systems* 30, 5 (2016), 849–898.
[4] Michael Bowling and Manuela Veloso. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136 (2002), 215–250.
[5] R. I. Brafman and M. Tennenholtz. 2003. R-MAX a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research* 3 (2003), 213–231.
[6] Colin F Camerer, Teck-Hua Ho, and Juin-Kuan Chong. 2004. A Cognitive Hierarchy Model of Games. *The Quarterly Journal of Economics* 119, 3 (2004), 861–898.
[7] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge University Press.
[8] Vincent Conitzer and Tuomas Sandholm. 2007. AWESOME: A General Multiagent Learning Algorithm that Converges in Self-Play and Learns a Best Response Against Stationary Opponents. *Machine Learning* 67, 1–2 (2007), 23–43.
[9] Jacob W Crandall. 2015. Robust Learning for Repeated Stochastic Games via Meta-Gaming.. In *Proc. Int'l Joint Conf. on Artificial intelligence*. 3416–3422.
[10] Steven Damer, Jeff Rosenschein, and Maria Gini. 2019. A Game for Cooperation with Strangers (Extended Abstract). In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*.
[11] Andreas Diekmann. 2004. The power of reciprocity: Fairness, reciprocity, and stakes in variants of the dictator game. *Journal of conflict resolution* 48, 4 (2004), 487–505.
[12] Mohamed Elidrisi, Nicholas Johnson, Maria Gini, and Jacob Crandall. 2014. Fast Adaptive Learning in Repeated Stochastic Games by Game Abstraction. In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*. 1141–1148.
[13] Christoph Engel. 2011. Dictator games: A meta study. *Experimental Economics* 14, 4 (2011), 583–610.
[14] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with opponent-learning awareness. In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 122–130.
[15] Werner Güth, Rolf Schmittberger, and Bernd Schwarze. 1982. An experimental analysis of ultimatum bargaining. *Journal of economic behavior & organization* 3, 4 (1982), 367–388.
[16] Dong Hao, Kai Li, and Tao Zhou. 2018. Payoff Control in the Iterated Prisoner's Dilemma. In *Proc. Int'l Joint Conf. on Artificial intelligence*. 296–302.
[17] Pablo Hernandez-Leal, Yusen Zhan, Matthew E. Taylor, L. Enrique Sucar, and Enrique Munoz de Cote. 2017. An exploration strategy for non-stationary opponents. *Journal of Autonomous Agents and Multi-agent Systems* 31, 5 (2017), 971–1002.
[18] Amir Jafari, Amy Greenwald, David Gondek, and Gunes Ercal. 2001. On No-Regret Learning, Fictitious Play, and Nash Equilibrium. In *Proc. of the Int'l Conf. on Machine Learning*. 226–233.
[19] Michael Johanson, Martin Zinkevich, and Michael Bowling. 2007. Computing robust counter-strategies. In *Advances in Neural Information Processing Systems (NIPS)*. 721–728.
[20] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101, 1 (1998), 99–134.
[21] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proc. Int'l Conf. on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 464–473.
[22] Michael L. Littman. 2001. Friend-or-Foe Q-learning in General-Sum Games. In *Proc. of the Int'l Conf. on Machine Learning*. 322–328.
[23] R. Powers, Y. Shoham, and T. Vu. 2007. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning* 67, 1–2 (2007), 45–76.
[24] T. Sandholm and S. Singh. 2012. Lossy stochastic game abstraction with bounds. In *Prod. of the 13th ACM Conf. on Electronic Commerce*. 880–897.
[25] Peter Stone, Gal A. Kaminka, Sarit Kraus, Jeffrey R. Rosenschein, and Noa Agmon. 2013. Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. *Artificial Intelligence* 203 (2013), 35–65.
[26] H. von Stackelberg. 2011. *Market Structure and Equilibrium*. Springer. Translation to English.
[27] Song Zuo and Pingzhong Tang. 2015. Optimal Machine Strategies to Commit to in Two-Person Repeated Games.. In *Proc. of the Nat'l Conf. on Artificial Intelligence*. 1071–1078.