The Iterated Traveler's Dilemma: Finding Good Strategies in Games with "Bad" Structure
Preliminary Results and Analysis

Philip C. Dasler and Predrag T. Tošić

Department of Computer Science, University of Houston
501 PGH Hall, 4800 Calhoun Road, Houston, Texas 77204-3010, USA
Contact email: *philip.dasler@gmail.com*, *ptosic@uh.edu*

Abstract. We study an interesting 2-player game known as the *Iterated Traveler's Dilemma*. The Traveler's Dilemma (TD) is a non-zero sum game in which each player has a large number of possible actions or moves. In the iterated context, this means many possible actions in each round and therefore an astronomic number of possible strategies overall. What makes Iterated TD particularly interesting is that its structure defies the usual prescriptions of classical game theory insofar as what constitutes "good" strategies. In particular, TD has a *single Nash equilibrium* (NE), yet that NE corresponds to a very low payoff for each individual player and essentially minimizes *social welfare*. We study possible ways of "playing well" from the standpoint of individual players, as well as the strategy pairs that maximize, not minimize, social welfare. We propose a number of possible strategies for ITD, from some trivial and rather "dumb" ones, to generalizations of "Tit-for-Tat," well-known from extensive studies of the (iterated) *Prisoner's Dilemma*, to some relatively sophisticated strategies where an agent tries to non-trivially model the behavior of the other agent in order to respond better in the future rounds. We perform a thorough comparison of 36 different strategies overall via a round-robin, everyone-against-everyone tournament in the spirit of Axelrod's work on the Prisoner's Dilemma [2]. We motivate the choices of strategies that comprise our tournament and then analyze the performance of various strategies. Finally, we draw some tentative conclusions and outline directions for the future work.

Keywords: *algorithmic game theory, two-player non-zero sum games, strategic behavior, Nash equilibria, round-robin tournaments*

1 Introduction

AI and multi-agent system (MAS) research communities have extensively studied interactions among two or more autonomous agents from a game-theoretic standpoint. Game theory is important to AI research because it provides mathematical foundations for modeling interactions among *self-interested* autonomous agents that may need to combine competition and cooperation with each other in non-trivial ways in order to meet their objectives or maximize their individual utilities [13, 15, 22]. A classical example of such interactions is the famous *iterated prisoner's dilemma* [1, 2], a two-person non-zero sum game that has been extensively studied by not just mathematicians and computer scientists, but also psychologists, sociologists, economists and political scientists.

We have been studying an interesting 2-player game known as the *Iterated Traveler's Dilemma* [6, 7, 9, 12]. The *Traveler's Dilemma* (TD) game is a non-zero sum game in which each player has a large number of possible actions or moves. In the iterated context, this means many possible actions in each round and therefore (for games with many rounds) an astronomic number of possible strategies overall. What makes Iterated TD particularly interesting, is that its structure defies the usual prescriptions of classical game theory insofar as what constitutes an "optimal" or even just a "good" strategy.

There are two fundamental general problems to be addressed in this context. One, is finding an optimal, or close to optimal, strategy. This is the central, "default" problem in classical game theory: finding the best "play" for each agent participating in the game. Akin to this is the problem of identifying (or evolving) *pairs of strategies* that would result in an overall desirable behavior, such as maximizing a joint utility function of some sort (e.g. "social welfare"). This second problem is, in essence, the goal of *mechanism design* [22]. We have begun investigating both these problems in the context of Iterated Traveler's Dilemma, a game that has thus far received only modest attention by the AI community. In this paper we shed some light on the first of the above two problems using a round-robin, everyone-against-everyone, many-round tournament. We draw some tentative conclusions and discuss promising ways forward with respect to the Iterated TD and similar games with "bad game-theoretic structures" alluded to earlier.

This paper is organized as follows. We first describe the Traveler's Dilemma (TD) game and briefly discuss why we find it interesting. We also briefly survey the existing prior art. We then describe the Iterated TD round-robin tournament that we have devised, implemented and experimented with; we particularly focus on the actual strategies we chose as the participants in this tournament, and on why these strategies are good examples of the kinds of strategies one would expect to be "reasonable" whether the actual players are humans or artificial software agents. We outline several metrics that we have used as yardsticks of performance of the various strategies involved in our round-robin tournament. Next, we summarize our tournament results with respect to "the bottom line" metric and discuss our findings. Finally, we outline some challenging directions for future research.

2 Traveler's Dilemma (TD)

The Traveler's Dilemma was originally introduced by Basu in 1994 [5]. The motivation behind the game was to show the limitations of classical game theory [11], and in particular the notions of *individual rationality* that stem from game theoretic notions of “solution” or ”optimal play” based on well-known mathematical concepts such as *Nash equilibria* [4, 5, 22]. Basu defines TD with a parable, paraphrased as follows:

“An airline loses two suitcases belonging to two different travelers. Both suitcases happen to be identical and contain identical antiques. An airline manager tasked to settle the claims of both travelers explains that the airline is liable for a maximum of $100 per suitcase. In order to determine an honest appraised value of the antiques, the manager separates both travelers so they can't confer and asks them to write down the amount of their value at no less than $2 and no larger than $100. If both write down the same number, he will treat that number as the true value and reimburse each traveler that amount. However, if one writes down a smaller number than the other, this smaller number will be taken as the true dollar value, and each traveler will receive that amount along with a bonus/malus: $2 extra will be paid to the traveler who wrote down the lower value and a $2 deduction will be taken from the other. The challenge is: what strategy should travelers follow to decide the value they should write down?”

Perhaps the most striking property of TD is that its *unique* Nash equilibrium, the action pair $(p, q) = (\$2, \$2)$, is actually rather bad for both players, assuming the players' *utility* is proportional to the dollar amount they receive. This choice of actions results in:

- very low payoff to each player individually (basically, only slightly above the absolutely worst possible, which is $0); and, moreover,
- it minimizes *social welfare*, if we understand social welfare to simply be the sum of the two players' individual utilities.

Yet, it has been argued [5, 7, 8] that a *perfectly rational* player, according to classical game theory, would “reason through” and converge to choosing the lowest possible value, $2, despite the fact that this results in very low payout and a minimization of social welfare. Given that the TD game is symmetric, each player would supposedly reason along the same line and, once selecting $2, not deviate from it. However, the non-equilibrium pair of actions $(\$100, \$100)$ results in each player earning $100, very near the best possible payoff. Hence, the early studies of TD concluded that this game demonstrates a woeful inadequacy of classical, Nash Equilibrium based game theory. In addition, it has been shown that humans (both game theory experts and laymen) will tend to play far from the equilibrium [6], resulting in better performance than the classical approach.

TD is interesting precisely because it has a unique Nash equilibrium, yet the corresponding strategies result in nearly as low a payoff as one can get. Adopting an alternative notion of game equilibrium does not help, either; for example, it is argued in [9] that the action pair $(\$2, \$2)$ is also the game's only *evolutionary equilibrium.* The situation is further complicated by the fact that the game's only “stable” strategy pair is easily seen to be nowhere close to *Pareto optimal*; there are many obvious ways of making both players much better off than if they play the equilibrium strategies. In particular, while neither stable nor an equilibrium in any sense of those terms, $(\$100, \$100)$ is the unique action pair that maximizes social welfare, and is, in particular, *Pareto optimal.* So, the fundamental question arises, how can agents, artificial or biological, evolve or learn to sufficiently trust each other so that they end up playing this optimal strategy pair in Iterated TD or any similar scenario that can be approximated as ITD? While TD shares some important structural properties with the more famous Prisoner's Dilemma, the two games are also considerably different in a number of respects. First, TD is quite a bit more complex, as each player has significantly more than two actions at its disposal. Because of this, TD lends itself to a different class of real world applications: those that require more than just a binary decision. For example, two companies competing for market share can be modeled by just such a game. By lowering its profit margin, a company can capture a larger market share (and thus greater profit). However, if a bidding war between the two companies gets out of hand, profit margins of both companies can quickly fall to the point where neither company can sustain a viable business. Having a greater understanding of what constitutes a good strategy in TD, and particularly ITD, would lend valuable insight into such problems.

In general, the TD's structure can be "tweaked" in various ways by controlling (i) the set of allowable bids (and, in particular, their "granularity") and (ii) the exact values of bonus/malus. We address the impact of these two parameters on game properties elsewhere, and just point out here that (a) the interaction of the two sets of parameters is nontrivial, and (b) TD can either be very different from or reduced to the classical PD. In the present paper, we confine ourselves to the "default" version of TD as described above, and ask the following question: how (dis)similar is this default version in comparison to PD. In particular, assuming similar metrics, are appropriate modifications of the successful strategies for the Iterated PD (as identified by Axelrod's seminal work) also going to be successful in the context of the Iterated TD? The objective of the research project whose early results are summarized in this paper, therefore, are as follows:

- to determine to what extent lessons learned about the Iterated PD "carry over" to the Iterated TD and, especially, where the prescriptions from the prior art on Iterated PD may be potentially misleading or even outright fail when applied to Iterated TD;
- to identify some successful strategies for Iterated TD, and to attempt to generalize them, thereby obtaining some insights into what approaches are likely to lead to good outcomes (at least, with respect to the selected performance metric(s) and the selected "pool(s)" of opponents);
- based on the above, to begin a quest for an appropriate mathematical notion of *(individual) rationality* that, unlike the pursuit of Nash (or other) equilibria, is actually applicable to a broad variety of two-person games (and not just those that are of a zero-sum, or very close to zero-sum, structure).

3 The Iterated TD Tournament

Our Iterated Traveler's Dilemma tournament is similar to Axelrod's Iterated Prisoner's Dilemma tournament [3]. In particular, it is a round-robin tournament in which each agent plays N matches against each other agent and its own "twin". A match consists of T rounds. In order to have statistically significant results (esp. given that many of our strategies involve randomization in various ways), we have selected $N = 100$ and $T = 1000$.

In each round, both agents must select a valid bid. Thus, the *action space* of an agent participating in the tournament is defined as $A = \{2, 3, \ldots, 100\}$. The method in which an agent chooses its next action for all possible histories of previous rounds is known as a strategy. A *valid strategy* is a function S that maps some set of inputs to an action: $S : \cdot \rightarrow A$. In general, the input may include the entire history of prior play, or, in the case of *bounded rationality* models, an appropriate summary of the past histories.

We next define the participants in the tournament, that is, the set of strategies that play one-against-one matches with each other. Let C be the set of agents competing in the tournament, defined as $C = \{c : (c \in S) \wedge (c \text{ is in the tournament})\}$. From this definition of the set of players or strategies, a pair of competing agents can be captured simply as $(x, y) \in C$. We note that, while we refer to agents as opponents and competitors, this does not imply that they necessarily always choose to act in an adversarial manner; rather, these terms are merely a common and convenient terminology.

We define agents' actions as follows:

$$x_t = \text{the bid traveler } x \text{ makes on round } t.$$
$$x_{nt} = \text{the bid traveler } x \text{ makes on round } t \text{ of match } n.$$

We next define the *reward function* that describes agent payoffs. The reward-per-round (heretofore, simply *reward*), $R : A \times A \rightarrow \mathbb{Z} \in [0, 101]$, for action α against action β, is defined as

$$R(\alpha, \beta) = min(\alpha, \beta) + 2 \cdot sgn(\beta - \alpha)$$

where $\alpha, \beta \in A$ and *sgn* denotes the usual *sign* function. Therefore, the total or cumulative reward $M : S \times S \rightarrow \mathbb{R}$ received by agent x in a match against y is defined as

$$M(x, y) = \sum_{t=1}^{T} R(x_t, y_t)$$

Within a sequence of matches, the reward received by agent x in the n^{th} match against y shall be denoted as $M_n(x, y)$.

4 Strategies in the Tournament

In selecting strategies for the tournament, we considered several types or classes of strategies, and chose a few "typical" representatives from each class, for a total of 36 distinct strategies. These strategies range from rather simplistic and even admittedly "dumb", to relatively complex and sophisticated. What follows is a detailed description of these strategies. For a brief summary and shorthand notation of these strategies, see Appendix A.

Random The first, and simplest, class of strategies are the oblivious agents, who always do the same thing regardless of what the opponent does. Each of these strategies plays a random value, uniformly distributed across a given interval. Ranges are as follows:

1. $\forall \alpha : \alpha \in (\mathbb{Z} \cap [2, 100]) \to p(x_t = \alpha) = \frac{1}{99}$
2. $\forall \alpha : \alpha \in (\mathbb{Z} \cap [99, 100]) \to p(x_t = \alpha) = \frac{1}{2}$

Simpletons The second extremely simple class of strategies are what we have dubbed *simpletons*: very simple deterministic strategies which choose the same action in every round. The values we used in the tournament were $x_t = 2$ (the lowest possible), $x_t = 51$ ("median"), $x_t = 99$ (slightly below the maximum; resulting in maximal individual payoff should the opponent consistently play the highest possible action), and $x_t = 100$ (the highest possible).

Tit-for-Tat-in-spirit The next class of strategies are those that can be viewed as *Tit-for-Tat-in-spirit*, where Tit-for-Tat is the famous name for a very simple, yet very effective, strategy for the iterated prisoner's dilemma [1–3, 14]. The idea behind *Tit-for-Tat* is simple: cooperate, until the first time your opponent defects; from that round on, choose the action that your opponent did on the previous round. Now, in the classical Iterated Prisoner's Dilemma, each player has only two possible actions, hence this simple description of Tit-for-Tat's logic defined a unique strategy. However, in the ITD, each agent has many actions at his disposal on each round (99 of them, to be exact). In general, playing high values can be reasonably considered as an approximate equivalent of "cooperating", whereas playing low values is an analogue of "defecting".

Following this basic intuition, we have defined several Tit-for-Tat-like strategies for ITD roughly grouped into two categories. One is the simplistic adaptation of Axelrod's Tit-for-Tat (TFT) for Iterated PD. Axelrod makes a case for why TFT performs consistently better than the other strategies in his Iterated PD tournament [3]: TFT is simple, robust, and resistent to *invasive strategies*. In particular, he describes it as an evolutionarily stable strategy, a term first defined by Smith in 1974 [17]. Each of our simplistic variants of TFT assumes that the opponent will tend not to deviate in its choice of action between rounds. Based on this extremely simplistic (and likely very naive) assumption, we have decided to implement the following Tit-for-Tat-in-spirit simple strategies: $x_{t+1} = y_t - 1$ and $x_{t+1} = x_t - 2$.

The more complicated TFT-in-spirit strategies we have considered are those that actually try to *predict* the other agent's next action. Of course, there are many ways of trying to predict what the other agent will do next. The particular TFT-predictor strategies that we chosen assume that the opponent is utilizing strategies similar to the Tit-for-Tat simple strategies. Still naive, though less so than the simple TFT strategies. Details can be found in Appendix A.

Mixed Another class of strategies in our tournament are what we have dubbed *mixed strategies*. For each mixed strategy, a strategy σ is selected from the other strategies in the competition (i.e., $\sigma \in C$) each round. Once a strategy has been selected, the value that σ would bid at time step t is bid: $x_t = (\sigma_t)_t$.

We have implemented several instances of mixed strategies. For each strategy, a σ is selected according to a probability mass distribution. The specific mixed strategies have been enumerated in Appendix A.

Buckets - Deterministic Next, we have included several variants of deterministic strategies where the past record is used in a "winner takes all" manner. More specifically, these strategies keep track over time how often each action is taken by the opposing agent in an array of "buckets". The opposing agent's next move is predicted to be the value that it has bid most often. Thus buckets updated each time-step as follows:

$$bucket_{t+1}[y_t] = bucket_t[y_t] + 1$$

The agent then plays "one under" its prediction. We have implemented several different versions of this strategy, differentiated by their tie-breaking methods. We use the following paradigms to deterministically select from multiple buckets that are equally full: the highest valued bucket wins; the lowest valued bucket wins; a random bucket wins; and the newest tied bucket wins.

Buckets - Probability Mass Function The next group of strategies use the same bucket methodology to track past behavior. Instead of winner-take-all, however, the buckets define a probability mass distribution to predict the opposing agent's next move. The values decay each round (in order to emphasize newer data over old) according to a retention rate ($0 \leq \gamma \leq 1$). Then, the bucket representing the opponents current bid is incremented. More formally, the buckets are updated each time-step as follows:

$$\forall i : i \in (\mathbb{Z} \cap [2, 100]) \rightarrow bucket_{t+1}[i] = \begin{cases} \gamma * bucket_t[i] & \text{if } i \neq y_t, \\ 1 + \gamma * bucket_t[i] & \text{if } i = y_t. \end{cases}$$

The opposing agent's next move is predicted by selecting a random value from the resulting probability distribution and the agent then, as before, plays one under its prediction.

We refer to this class of strategies as *probability mass buckets*. The main parameter that can be varied as wished in order to produce different "bucket-like" strategies is the *rate of retention*. We have entered into our tournament several instances of this strategy using the following rate of retention values (γ): 1.0, 0.8, 0.5 and 0.2. We observe that these strategies based on probability mass buckets are quite similar to a learning model by Capra et al. in [7].

Simple Trend Analysis The next class of strategies are based on an agent attempting to predict simple trends based on the previous rounds. We explicitly assume *bounded rationality* here; in particular, an agent will look at the most recent k rewards and, based on those, establish a line of best fit. The slope of this line will determine the strategy taken. If the slope is near zero then the system is relatively stable and the agent will play the "one under" strategy, meaning, $x(t) = y(t-1) - 1$ (where x is the action of this agent, y is the opponent's action, and t indicates the round). If the slope is significantly negative, then the system is trending towards the Nash equilibrium and, thus, smaller rewards. In this case, the agent will attempt to maximize social welfare and play \$100. If the trend is significantly positive, than the agent will continue to place the same bid it has been in order to continue the trend of increasing rewards. We are using an arbitrary slope threshold ϵ to determine a significant slope, with $\epsilon = 0.5$. Thus, our strategy is defined as follows:

$$x_{t+1} = \begin{cases} x_t & \text{if } m > \epsilon \\ y_t - 1 & \text{otherwise} \\ 100 & \text{if } m < -\epsilon \end{cases}$$

We are implementing variants of this strategy with the following values of parameter k: $k \in \{3, 10, 25\}$.

Q-Learning We next briefly describe the participants in our ITD tournament that are capable of *learning* from the past and adjusting their strategy based on what they have learned. In particular, the learners in our tournament are simple implementations of *Q-learning* [20, 21] as a way of predicting the best action at time $(t + 1)$ based on previous action selections and payoffs. This is similar to the Friend-or-Foe Q-learning method [10] without the limitation of having to classify the allegiance of one's opponent.

Q-learning is often implemented as a table of state/action pairs, which, however, scales poorly. TD already contains a large action/state space, but, when coupled with the need for action history, the computational complexity quickly grows to impractical proportions. To address this scaling problem, we break up the state/action spaces into a few clusters. These clusters, meant to capture the intention of actions, are defined as follows:

State:

1. The opponent played higher than our last bid;
2. The opponent played the same bid we did;
3. The opponent played lower than our last bid.

Action:

1. Play one higher than our previous bid;
2. Played the same bid that we played last time;
3. Play one lower than our previous bid.

Recall that *actions* are defined for just a single time-step. The actual implementation treats the state as a collection of moves by the opponent over the last k rounds, with k arbitrarily equal to 5, capturing some history without data sizes getting out of control. All Q-values are initialized to 50.5, the mean of all possible rewards, while the discount rate is set to 0.5, a reasonable balance between current and future rewards.

We have implemented this basic Q-learning algorithm with the learning rates of 0.8, 0.5 and 0.2.

Zeuthen Strategies These strategies, like the Zeuthen Strategy for negotiation, base their bids on the level of one's *risk aversion* [23]. While this isn't a strict negotiation, we are treating individual bids (i.e. x_t and y_t) as proposals. The assumption is made that if $y_t = i$, then the proposal being made by Y is

$(i+1, i)$, as this leads to the best outcome for Y, given its bid. The same applies for X's bid. Thus, at each time step $t+1$ the agent looks at the two bids made at t and determines the amount of risk of each agent. For the Traveler's Dilemma, the conflict deal (the deal made when no acceptable proposal can be found) would be the Nash Equilibrium of $(2, 2)$. As defined in [15] (though originating in [23]) the risk of an agent is:

$$Risk_x(t) = \frac{\text{utility agent } x \text{ loses by conceding and accepting agent } y\text{'s last offer}}{\text{utility agent } x \text{ loses by not conceding and causing a conflict}}$$

Due to space constraints, we leave out mathematical details and briefly summarize the intuition behind Zeuthen strategies. If an agent's risk is higher than its opponent's, it continues to make the same bid. If lower, then the agent makes the *minimal sufficient concession.* The concession is sufficient if it causes the agent's risk to be higher than that of its opponent and is considered minimal by increasing the opponent's utility the least. Due to the peculiar structure of the TD game matrix, it is possible for the "concession" to actually lead to a loss of utility for the opponent. This, however, seems to go against the very notion of what is meant by the term *concession.* Thus, we have implemented the following two strategies:

1. Only positive concessions are allowed. The strategy will find the minimal sufficient concession that does not decrease the opponent's utility.
2. Negative and positive concessions are allowed. The strategy will find the truly minimal "concession" that is enough to reverse the relationship in the agents' risk values.

5 Utility metrics

Our experimentation and subsequent analysis were performed with respect to four distinct utility metrics. The first, U_1, treats the actual dollar amount as the payoff to the agent. This is typical for zero-sum games and, prior literature on Iterated TD generally considers only this metric. In contrast, U_2 is a "pairwise victory" metric: an agent strives to beat its opponent, regardless of the actual dollar amount it receives. Whether an agent earns $1,000 or $5, it counts equally with respect to U_2 if the agent earned more than its opponent.

Finally, we introduce two additional metrics, U_3 and U_3', that attempt to capture both the payoff ($ amount) that an agent has achieved, and the "opportunity lost" due to not playing differently (and in particular, not knowing what the other agent would do). In a sense, both of these metrics attempt to quantify how much an agent wins compared to an omniscient agent, that is, one that always correctly predicts the other agent's bid without directly influencing the opponent's action.

6 Results and Analysis

The Traveler's Dilemma Tournament that we have experimented with involves a total of 36 competitors (i.e., distinct strategies). Each competitor plays each other competitor (including its own "twin") 100 times. Each match is played for 1000 rounds. The following briefly summarizes our main findings.

The top three performers in our tournament, with respect to the earned \$ amount as the "bottom line" (metric U_1), are three dumb strategies that always bid very high; interestingly enough, randomly alternating between the highest possible bid (\$100) and "one under" the highest bid (\$99) slightly outperforms both "always max. possible" and "always one under max. possible" strategies (see Table B.1). We find it somewhat surprising that the performance of Tit-for-Tat-based strategies varies so greatly depending on the details of bid prediction method and metric choice. So, while a relatively complex TFT-based strategy that, in particular, (i) makes a nontrivial model of the other agent's behavior and (ii) "mixes in" some randomization, is among the top performers with respect to metric U_1, other TFT-based strategies have fairly mediocre performance with respect to the same metric, and are, indeed, scattered all over the tournament table. In contrast, if metric U_3 is used, then the simplest, deterministic "one under the opponent's previous bid" TFT strategy, is the top performer among all 36 strategies in the tournament – while more sophisticated TFT strategies, with considerably more complex models of the opponent's behavior and/or randomization involved, show fairly average performance. Moreover, if U_3 is used as the yardstick, then (i) 3 out of the top 4 performers overall are TFT-based strategies, and (ii) all simplistic TFT strategies outperform all more sophisticated ones.

Not very surprisingly, the top (and bottom) performers with respect to metrics U_1 and U_2 turn out to be practically inverted; so, for example, the very best performer with respect to U_2 is the "dumb" strategy (which also happens to be the only non-dominated strategy in the classical game theoretic-sense), namely, "always bid \$2 no matter what the other agent does". On the other hand, the three best performers with respect to U_1 are all among the four bottom performers with respect to U_2, with the only strategy that *may* maximize social welfare (bidding \$100 against a collaborative opponent) falling at the rock bottom of the tournament rankings for U_2. The main conclusion we draw from this performance inversion is that *when a non-zero sum two-player game has a structure that makes it very far from being zero-sum, the traditional precepts from classical game theory on what constitutes good strategies are quite likely to fail.* This does not mean to suggest that classical game theory is useless; rather, we'd argue that the appropriate quantitative, mathematical models of rationality for zero-sum, or nearly zero-sum, encounters aren't necessarily the most appropriate notions for games that are rather far from being zero-sum.

Back to our tournament results, what we have found more surprising than the performance inversion between metric U_2 and the other three metrics is the relative mediocrity of the learning based strategies: Q-learning based strategies did not excel with respect to any of the four metrics we studied. On the other hand, it should be noted that the adaptability of Q-learning based strategies apparently ensures that they do not do too badly overall, regardless of the choice of metric. We also note that, for our game, the choice of the learning rate (we selected three values, one corresponding to forgetting older rounds fairly quickly, one that forgets relatively slowly, and one "in between") seems to make very little difference: for each of the four metrics, all three Q-learning based strategies show similar performance, and hence end up ranked adjacently or almost adjacently (in a tournament where, recalling the "demographics", three learning strategies are "mingled" with 33 non-learning ones). It seems likely that our highly restrictive reduction of the action/state space provides too little information for the Q-learning algorithm to sufficiently model its opponent.

Another somewhat surprising outcome is fairly poor performance of both Zeuthen-based strategies; again, the likely explanation is in the peculiar structure of the Iterated TD game, but admittedly this result warrants further analysis. Interestingly, the more altruistic "positive" Zeuthen strategy always outperforms the negative one – except with respect to metric U_2, for which we have already observed a rather general *performance inversion* with respect to the other metrics.

For space constraints, we just outline some other observations, and leave more detailed analysis for the future work:

- It is far from clear whether more complex models of the other agent really help insofar as bidding better, and hence performing better, in the long run.
- Not all TFT-based strategies in the TD are born equal; in fact, performance of different TFT variants tend to vary broadly with respect to all four of our metrics. This observation opens up interesting questions from meta-learning [18, 19] and meta-reasoning standpoints: how can one design TFT-based strategies that are likely to do well across tournaments (that is, choices of opponents) and across performance metrics.

7 Summary and Future Work

We have studied the *Iterated Traveler's Dilemma* two-player game by designing, implementing and analyzing a round robin tournament with 36 distinct participating strategies. Our detailed study of the performance of various strategies with respect to several different metrics has corroborated that, for a game whose structure is far from zero-sum, the traditional game-theoretic notions of rationality and optimality may turn out to be unsatisfactory if not outright deleterious. Our analysis also raises several interesting questions, among which there are several we are particularly keen to further investigate:

- to what extent can simple models of learning help performance,
- to what extent do complex opponent models help agent in iterated play, and
- what effects would an adjustment in the bonus/malus have on agents?

In our future work, in addition to more detailed analysis of the strategies summarized in this paper and study of some new ones, we also plan to pursue a rigorous mathematical analysis of the Iterated TD game structure similar to the analysis done for the Iterated Prisoner's Dilemma by Smale [16]; such analysis would hopefully provide solid foundations for designing individual strategies that would "push" an adaptable opponent away from the low-paying Nash equilibrium and towards higher bids that are mutually beneficial to both agents.

Bibliography

[1] Axelrod, R.: Effective choice in the prisoner's dilemma. Journal of Conflict Resolution 24(1), 3 –25 (Mar 1980)

[2] Axelrod, R.: The evolution of cooperation. Science 211(4489), 1390–1396 (1981)

[3] Axelrod, R.: The evolution of cooperation. Basic Books (2006)

[4] Basu, K.: The traveler's dilemma. Scientific American Magazine (Jun 2007)

[5] Basu, K.: The traveler's dilemma: Paradoxes of rationality in game theory. The American Economic Review 84(2), 391–395 (May 1994)

[6] Becker, T., Carter, M., Naeve, J.: Experts playing the traveler's dilemma. Tech. rep., Department of Economics, University of Hohenheim, Germany (Jan 2005)

[7] Capra, C.M., Goeree, J.K., Gmez, R., Holt, C.A.: Anomalous behavior in a traveler's dilemma? The American Economic Review 89(3), 678–690 (Jun 1999)

[8] Goeree, J.K., Holt, C.A.: Ten little treasures of game theory and ten intuitive contradictions. The American Economic Review 91(5), 1402–1422 (Dec 2001)

[9] Land, S., van Neerbos, J., Havinga, T.: Analyzing the traveler's dilemma Multi-Agent systems project (2008), http://www.ai.rug.nl/mas/finishedprojects/2008/JoelSanderTim/index.html

[10] Littman, M.L.: Friend-or-Foe q-learning in General-Sum games. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 322–328. Morgan Kaufmann Publishers Inc. (2001)

[11] Neumann, J.V., Morgenstern, O.: Theory of games and economic behavior. Princeton university press (1944)

[12] Pace, M.: How a genetic algorithm learns to play travelers dilemma by choosing dominated strategies to achieve greater payoffs. In: Proceedings of the 5th international conference on Computational Intelligence and Games. p. 194200 (2009)

[13] Parsons, S., Wooldridge, M.: Game theory and decision theory in Multi-Agent systems. AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS 5, 243—254 (2002)

[14] Rapoport, A., Chammah, A.M.: Prisoner's Dilemma. University of Michigan Press (Dec 1965)

[15] Rosenschein, J.S., Zlotkin, G.: Rules of encounter: designing conventions for automated negotiation among computers. MIT Press (1994)

[16] Smale, S.: The prisoner's dilemma and dynamical systems associated to Non-Cooperative games. Econometrica 48(7), 1617–1634 (Nov 1980)

[17] Smith, J.M.: The theory of games and the evolution of animal conflicts. Journal of Theoretical Biology 47(1), 209–221 (Sep 1974)

[18] Sun, R.: Meta-Learning processes in Multi-Agent systems. IN PROCEEDINGS OF INTELLIGENT AGENT TECHNOLOGY pp. 210—219 (2001)

[19] Tosic, P.T., Vilalta, R.: A unified framework for reinforcement learning, co-learning and meta-learning how to coordinate in collaborative multi-agent systems. Procedia Computer Science 1(1), 2211–2220 (May 2010)

[20] Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992)

[21] Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, University of London, King's College (United Kingdom), England (1989), dr.

[22] Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley and Sons (2009)

[23] Zeuthen, F.F.: Problems of monopoly and economic warfare / by F. Zeuthen ; with a preface by Joseph A. Schumpeter. Routledge and K. Paul, London : (1967), first published 1930 by George Routledge & Sons Ltd.

A Notations, Abbreviations, and Enumerations

Randoms The Random strategies bid a value from a uniformly distributed set between α and β, inclusive. The notation is as follows: Random [α, β]

Simpletons These functions will bid the same value every round, referred to as α. The notation is: Always α.

Tit-for-Tat, simple The simple Tit-for-Tat strategies bid some value ϵ below the bid made by competitor c in the last round, where $c \in [x, y]$. Notation: TFT - Simple $(c - \epsilon)$.

Tit-for-Tat, predictors This strategy first compares whether its bid was lower than, equal to, or higher than that of its opponent. Then the strategy will make a bid similar to the simple TFT strategy, i.e. some value ϵ below the bid made by competitor c in the last round, where $c \in [x, y]$. Notation: TFT - low$(c_0 - \epsilon_0)$ equal$(c_1 - \epsilon_1)$ high$(c_2 - \epsilon_2)$.Based on these assumptions, we have decided to implement the following Tit-for-Tat-predictor strategies:

1. $x_{t+1} = \begin{cases} x_t & \text{if } x_t < y_t, \\ x_t & \text{if } x_t = y_t, \\ y_t - 1 & \text{if } x_t > y_t. \end{cases}$

2. $x_{t+1} = \begin{cases} x_t & \text{if } x_t < y_t, \\ x_t - 2 & \text{if } x_t = y_t, \\ y_t - 1 & \text{if } x_t > y_t. \end{cases}$

3. $x_{t+1} = \begin{cases} x_t - 2 & \text{if } x_t < y_t, \\ x_t & \text{if } x_t = y_t, \\ y_t - 1 & \text{if } x_t > y_t. \end{cases}$

4. $x_{t+1} = \begin{cases} x_t - 2 & \text{if } x_t < y_t, \\ x_t - 2 & \text{if } x_t = y_t, \\ y_t - 1 & \text{if } x_t > y_t. \end{cases}$

5. $x_{t+1} = \begin{cases} y_t - 1 & \text{if } x_t < y_t, \\ x_t - 1 & \text{if } x_t = y_t, \\ x_t - 1 & \text{if } x_t > y_t. \end{cases}$

Mixed The mixed strategies combine up to three strategies based on a probability mass. The notation is as follows, where each strategy being used is followed by its probability of use on any given round: Mixed - [($Strategy_0$, $Probability_0$); ($Strategy_1$, $Probability_1$); ($Strategy_2$, $Probability_2$)]For each strategy, a σ is selected according to the following probability mass functions:

1. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 1}, \\ 0.2 \text{ if } \sigma = \text{Always 100}, \\ 0 \quad \text{otherwise.} \end{cases}$

2. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 1}, \\ 0.2 \text{ if } \sigma = \text{Always 2}, \\ 0 \quad \text{otherwise.} \end{cases}$

3. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 1}, \\ 0.1 \text{ if } \sigma = \text{Always 100}, \\ 0.1 \text{ if } \sigma = \text{Always 2}, \\ 0 \quad \text{otherwise.} \end{cases}$

4. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 5}, \\ 0.2 \text{ if } \sigma = \text{Always 100}, \\ 0 \quad \text{otherwise.} \end{cases}$

5. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 5}, \\ 0.2 \text{ if } \sigma = \text{Always 2}, \\ 0 \quad \text{otherwise.} \end{cases}$

6. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Predictor 5}, \\ 0.1 \text{ if } \sigma = \text{Always 100}, \\ 0.1 \text{ if } \sigma = \text{Always 2}, \\ 0 \quad \text{otherwise.} \end{cases}$

7. $p(\sigma) = \begin{cases} 0.8 \text{ if } \sigma = \text{TFT Simple 1}, \\ 0.2 \text{ if } \sigma = \text{Rand } [99, 100], \\ 0 \quad \text{otherwise.} \end{cases}$

Buckets - Deterministic These strategies keep a count of each bid by the opponent in an array of "buckets". The bucket that is most full (i.e., the value bid most often) is used as the predicted value, with ties being broken by the specified method. The notation is as follows: Buckets - Fullest Wins, $TieBreakingMethod$

Buckets - Probability Mass Function As above, this strategy counts instances of the opponent's bids and uses them to predict its next bid. Rather than picking the most bid value, the buckets are used to define a probability mass function from which a prediction is randomly selected. Notation: Buckets - PMF, Retention Rate = γ

Simple Trend This strategy looks at the previous k time steps, creates a line of best fit on the rewards, and compares it to a threshold ϵ. Notation: Simple Trend - K = k, Eps = ϵ.

Q-learning This strategy uses a learning rate α to emphasize new information and a discount rate γ to emphasize future gains. Notation: Q Learning - Learn Rate = α, Discount = γ

Zeuthen Strategies A Zeuthen Strategy calculates the level of risk of each agent, and makes *concessions* accordingly. This "concession" may be negative, or may be restricted to having only a positive effect on the opponents utility. Notation: Zeuthen Strategy - $AllowableConcession$.

B Tournament Rankings

For definitions of the shorthand notation used here, see appendix A

B.1 Ranking Based on U1

0.730402	Random [99, 100]
0.729153	Always 100
0.729039	Always 99
0.716543	Mixed - TFT - low(y-1) equal(x-1) high(x-1), 80%); (Always 100, 20%)
0.715729	Buckets - PMF, Retention Rate = 0.2
0.714632	Buckets - PMF, Retention Rate = 0.5
0.714508	Buckets - PMF, Retention Rate = 0.8
0.661574	Simple Trend - K = 3, Eps = 0.5
0.656294	Mixed - TFT - Simple (y-1), 80%); (Random [99, 100], 20%)
0.632261	Simple Trend - K = 10, Eps = 0.5
0.627474	Mixed - TFT - low(x) equal(x) high(y-1), 80%); (Always 100, 20%)
0.604617	Simple Trend - K = 25, Eps = 0.5
0.537722	Zeuthen Strategy - Positive
0.511811	Q Learning - Learn Rate = 0.8, Discount = 0.5
0.510264	Q Learning - Learn Rate = 0.5, Discount = 0.5
0.509670	TFT - low(y-1) equal(x-1) high(x-1)
0.508661	Q Learning - Learn Rate = 0.2, Discount = 0.5
0.499094	Mixed - TFT - low(y-1) equal(x-1) high(x-1), 80%); (Always 100, 10%); (Always 2, 10%)
0.488350	Buckets - Fullest Wins, Highest Breaks Ties
0.438149	Buckets - Fullest Wins, Random Breaks Ties
0.436808	TFT - Simple (y-1)
0.420152	Buckets - Fullest Wins, Newest Breaks Ties
0.396397	Always 51
0.396228	TFT - Simple (y-2)
0.385955	Buckets - Fullest Wins, Lowest Breaks Ties
0.305122	Random [2, 100]
0.285491	Mixed - TFT - low(y-1) equal(x-1) high(x-1), 80%); (Always 2, 20%)
0.189364	Mixed - TFT - low(x) equal(x) high(y-1), 80%); (Always 100, 10%); (Always 2, 10%)
0.146467	TFT - low(x) equal(x-2) high(y-1)
0.142023	TFT - low(x) equal(x) high(y-1)
0.105737	Zeuthen Strategy - Negative
0.031648	TFT - low(x-2) equal(x) high(y-1)
0.031134	TFT - low(x-2) equal(x-2) high(y-1)
0.028321	Mixed - TFT - low(x) equal(x) high(y-1), 80%); (Always 2, 20%)
0.026786	Buckets - PMF, Retention Rate = 1.0
0.026459	Always 2

Table 1. for metric description see Section (5)