Reinforcement Learning Produces Dominant Strategies for the Iterated Prisoner's Dilemma

Marc Harper Vincent Knight

Abstract

We present tournament results and several powerful strategies for the Iterated Prisoner's Dilemma created using reinforcement learning techniques (evolutionary and particle swarm algorithms). These strategies are trained to perform well against a corpus of over 100 distinct opponents, including many well-known strategies from the literature, and all the trained strategies win standard tournaments against the total collection of other opponents. We also trained variants to win noisy tournaments.

1 Introduction

* Update on Axelrod Library

The Axelrod library [**knight2016open**] now contains over 200 strategies, many from the scientific literature, including classic strategies like Win Stay Lose Shift [**nowak1993strategy**] and previous tournament winners such as OmegaTFT [**slany2007some**], Adaptive Pavlov [**li2007design**], and ZDGTFT2 [**stewart2012extortion**].

There are several previous publications that use evolutionary algorithms to evolve IPD strategies in various circumstances [**marks1989niche**] [**fogel1993evolving**] [**ashlock2006training**] [**ashlock2006changes**] [**vassiliades2010multiagent**] [**ashlock2014shaped**] [**ashlock2014evolution**] [**ashlock2015multiple**] [**barlow2015varying**] [**sudo2015effects**]. See also [**gaudesi2016exploiting**] for a strategy trained to win against a collection of well-known IPD opponents and see [**franken2005particle**] for a prior use of particle swarm algorithms. Our results are unique in that we are able to train against a large collection of well-known strategies available in the scientific literature.

2 The Strategy Archetypes

The Axelrod library now contains many strategies based on machine learning methods. Most are deterministic, use many rounds of memory, and perform extremely well in tournaments.

2.1 LookerUp

The first strategy trained with reinforcement learning in the library is based on lookup tables. The strategy encodes a set of deterministic responses based on the opponent's first n_1 moves, the opponent's last m_1 moves, and the players last m_2 moves. If $n_1 > 0$ then the player has infinite memory depth, otherwise it has depth $\max m_1, m_2$. Although we tried various combinations of n_1, m_1, and m_2, the best performance at the time of training was obtained for $n_1 = m_1 = m_2 = 2$ and generally for $n_1 > 0$. First impressions matter in the IPD. The library includes a strategy called EvolvedLookerUp2_2_2 which is among the top strategies in the library.

This archetype can be used to train deterministic memory-n strategies with the parameters $n_1 = 0$ and $m_1 = m_2 = n$. For $n = 1$, the resulting strategy cooperates if the last round was mutual cooperation and defects otherwise.

Two strategies in the library, Winner12 and Winner21, from [**Mathieu2015**], are based on lookup tables for $n_1 = 0$, $m_1 = 1$, and $m_2 = 2$. The strategy Winner12 emerged in less than 10 generations of training in our framework using a score maximizing objective. Strategies nearly identical to Winner21 arise from training with a Moran process objective.

2.2 PSO Gambler

PSO Gambler is a stochastic variant of LookerUp. Instead of deterministically encoded moves the lookup table emits probabilities which are used to choose cooperation or defection. The library includes a player trained with $n_1 = m_1 = m_2 = 2$ that is *mostly deterministic*, with most of the probabilities being 0 or 1. At one time this strategy outperformed EvolvedLookerUp2_2_2.

This strategy type can be used to train arbitrary memory-n strategies and a memory one strategy was trained and is called PSO Gambler Mem 1, with probabilities $(\Pr(\text{C—CC}), \Pr(\text{C—CD}), \Pr(\text{C—DC}), \Pr(\text{C—DD})) = (1, 0.5217, 0, 0.121)$. Though it performs well in deterministic tournaments it is not as good as the longer memory strategies, and is bested by a similar strategy that also uses the first round of play.

These strategies are trained with a particle swarm algorithm rather than an evolutionary algorithm (though the former would suffice). Particle swarm algorithms have been used to trained IPD strategies previously [**franken2005particle**].

2.3 ANN: Single Layer Artificial Neural Network

Strategies based on artificial neural networks can also be trained with an evolutionary algorithm. A variety of features are computed from the history of play such as the opponents trailing moves, the total number of cooperations of the player and the opponent, and several others, which are then input into a feed forward neural network with one layer and user-supplied width. An inner layer with just five nodes performs quite well in both deterministic and noisy tournaments. The output of the ANN used in this work is deterministic and a stochastic variant that outputs probabilities rather than exact moves could be easily created.

2.4 Finite State Machines

We used strategies based on finite state machines to create a number of strategies. These strategies are deterministic and are efficient computational. At each state the machine transitions to a new state and plays a specified move depending on the opponent's last action. These strategies can encode a variety of other strategies, including classic strategies like TitForTat, encode handshakes, and grudging strategies that always defect after an opponent defection.

2.5 Hidden Markov Models

We also trained stochastic versions of finite state machine players called hidden Markov model players or HMMs. These strategies also encode an internal state with probabilistic transitions to other states and cooperate or defect with various probabilities at each state. These are the best performing stochastic strategies in the library but take longer to train due to their stochasticity.

2.6 Meta Strategies

Last but not least there are several strategies based on ensemble methods that are common in machine learning called Meta strategies. These strategies are composed of a team of other strategies and each is polled for its desired next move. The ensemble then selects the next move based on some rule, such as the consensus vote in the case of MetaMajority or the best individual performance in the case of MetaWinner. These strategies were among the best in the library before the inclusion of those trained by reinforcement learning.

Because these strategies inherit many of the properties of the strategies on which they are based, including using the match length to defect on the last rounds of play, these strategies were omitted from the tournament results.

3 Results

3.1 Standard Tournament

We conducted a tournament with all strategies in the Axelrod library. The top 11 performing strategies by mean payoff are all strategies trained to maximize total payoff against a subset of the strategies (Table 1). The next strategy is Desired Belief Strategy (DBS) which actively analyzes the opponent and responds accordingly. The next two strategies are Winner12, based on a lookup table, Omega TFT, and Fool Me Once, a grudging strategy that defects indefinitely on the second defection. All strategies in the tournament follow a simple set of rules in accordance with earlier tournaments:

- Players carry no acquired state between matches
- Players cannot observe the outcome of other matches
- Players cannot identify their opponent by any label or identifier
- Players cannot manipulate or inspect their opponents in any way

Pairwise payoff results are given as a heatmap (Figure) which shows that many strategies achieve mutual cooperation with each other and that the top performing strategies never defect first but are able to exploit weaker strategies that attempt to defect.

The strategies that win the most matches are Defector, Aggravater, followed by handshaking and zero determinant strategies. This includes two handshaking strategies that were the result of training to maximize Moran process fixation. No strategies were trained specifically to win matches. None of the top scoring strategies appear in the top 20 list of strategies ranked by match wins.

Rank	Player	Median Score
1	EvolvedLookerUp2_2_2	2.9584
2	Evolved HMM 5	2.9545
3	Evolved FSM 16	2.9535
4	PSO Gambler 2_2_2	2.9449
5	Evolved ANN 5	2.9198
6	Evolved FSM 16 Noise 05	2.9197
7	PSO Gambler 1_1_1	2.9168
8	Evolved ANN	2.9151
9	Evolved FSM 4	2.9102
10	PSO Gambler Mem1	2.9088
11	Evolved ANN 5 Noise 05	2.8705
12	DBS: 0.75, 3, 4, 3, 5	2.8575
13	Winner12	2.8511
14	Omega TFT: 3, 8	2.8466
15	Fool Me Once	2.8450

Table 1: Standard Tournament: Top Ranking Strategies by Median Score in 15,000 Tournaments

Standard - Payoff

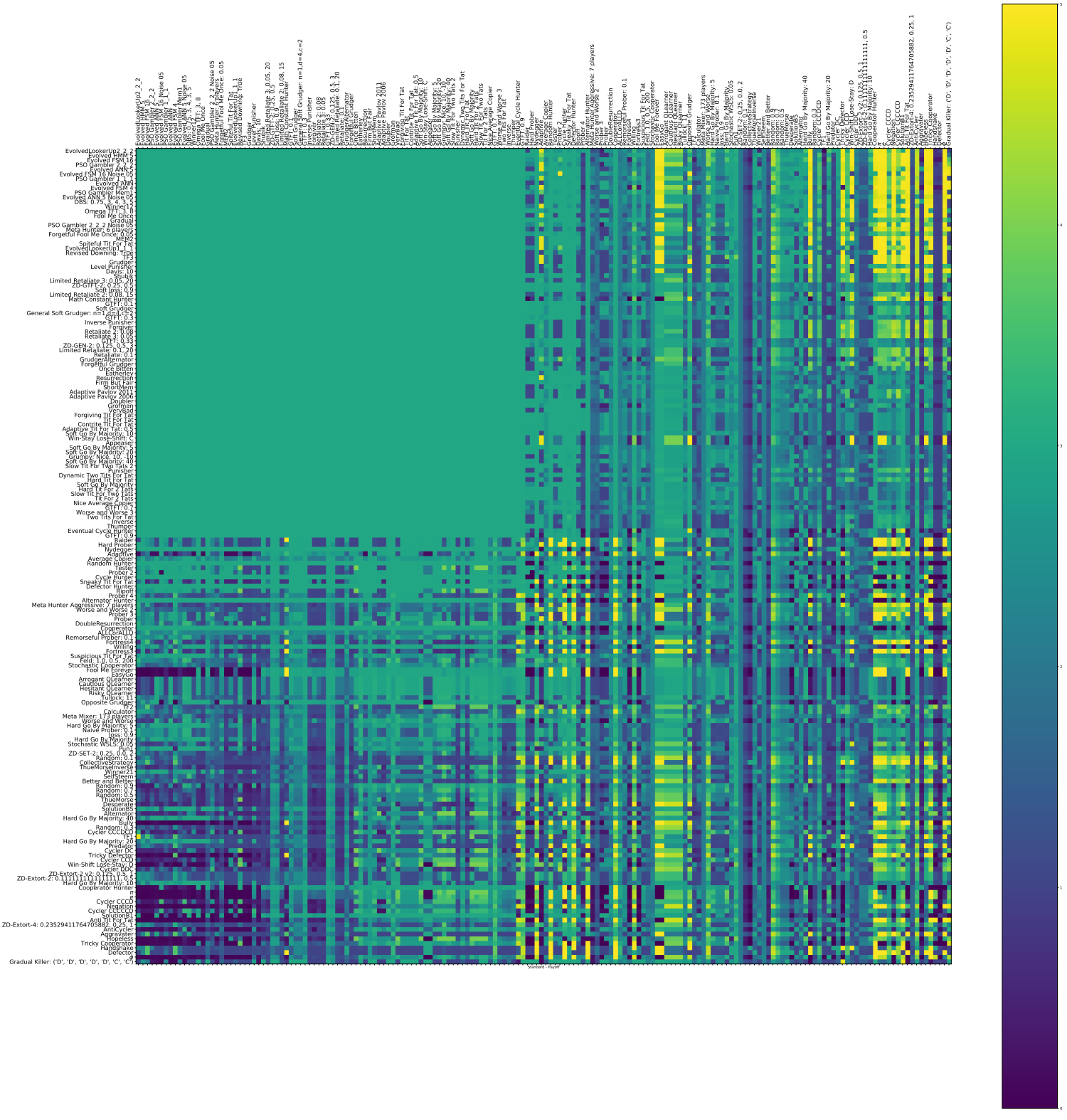

Figure 1: Pairwise Match results for the Standard Tournament

3.2 Noisy Tournament

We also ran noisy tournaments in which there is a 5% chance that an action is flipped. The best performing strategies in mean payoff are DBS, designed to correct for noise, followed by two strategies trained in the presence of noise, Spiteful TFT (TFT but defects indefinitely if the opponent defects twice consecutively), two strategies trained without noise, then OmegaTFT (also designed to handle noise).

The strategies trained in the presence of noise are also among the best performers in the absence of noise. The cluster of mutually cooperative strategies is broken by the noise at 5%. A similar collection of players excels at winning matches but again they have a poor total payoff.

The strategies tallying the most wins are somewhat similar, with Defector, the handshaking CollectiveStrategy, and Aggravate appearing as the top three again, followed by Spiteful TFT and strategies designed to be retaliatory. Players ranked 12, 13, and 15 (TF3, Fool Me Once, and Evolved ANN 5) all appear in the top 15 strategies by score, indicating that noise breaks the asymmetry between wins and scores to some extent.

Rank	Player	Median Score
1	DBS: 0.75, 3, 4, 3, 5	2.4658
2	Evolved ANN 5 Noise 05	2.4052
3	Evolved FSM 16 Noise 05	2.3709
4	Spiteful Tit For Tat	2.3500
5	Evolved ANN	2.3375
6	Evolved ANN 5	2.3321
7	Omega TFT: 3, 8	2.3160
8	Fool Me Once	2.3159
9	MEM2	2.3141
10	Math Constant Hunter	2.3133
11	Adaptive	2.3051
12	Hard Prober	2.3046
13	Davis: 10	2.3040
14	Grudger	2.3036
15	TF3	2.3006
16	Meta Hunter: 6 players	2.2991

Table 2: Noisy Tournament: Top Ranking Strategies by Median Score in a 15000 Tournaments with 5% noise

Noisy - Payoff

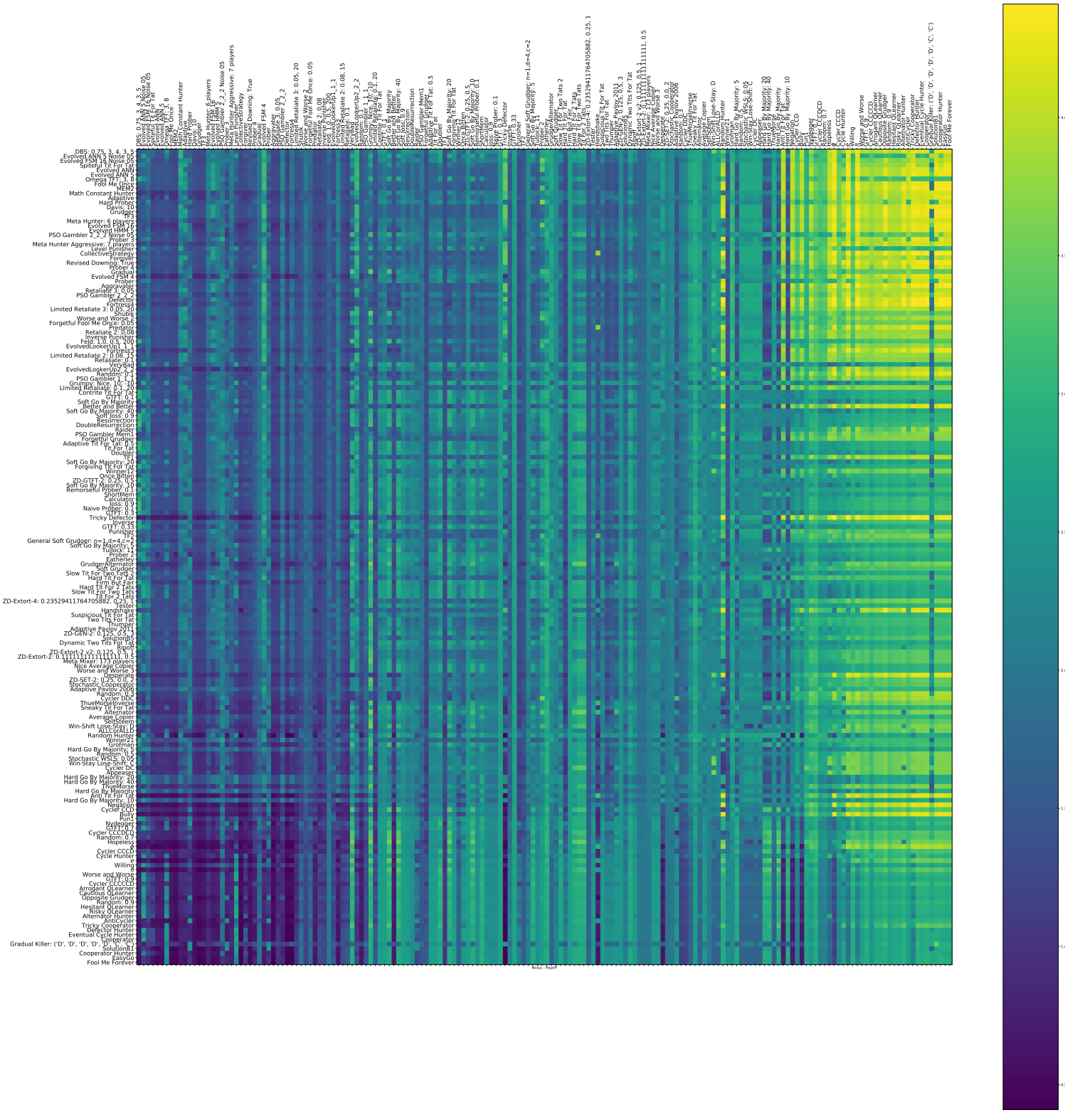

Figure 2: Pairwise Match results for the Noisy Tournament

4 Methods

We trained a variety of strategies using evolutionary algorithms, and in the case of PSO Gambler particle swarm algorithms. The evolutionary algorithms used standard techniques, varying strategies by mutation and crossover, and evaluating the performance against each opponent for many repetitions. The best performing strategies in each generation are persisted, variants created, and objective functions computed again. This process continues for approximately 200 generations or until strategies no longer improve significantly.

All training code is available on github. There are objective functions for * total payoff * total payoff difference * total Moran process wins (fixation probability)

New strategies can be easily trained with variations including noise, spatial structure, and probabilistically ending matches.

5 Discussion

The tournament results indicate that pre-trained strategies are generally better than human designed strategies at maximizing payoff against a diverse set of opponents. A simple evolutionary algorithm produces strategies based on multiple standard machine learning techniques that are able to achieve a higher mean score than any other known opponent in a standard tournament. Most of the trained strategies use multiple rounds of the history of play (some using all of it) and outperform memory-one strategies (though the trained memory one strategy performs well). The generic structure of the trained strategy did not appear to be critical – strategies based on lookup tables, finite state machines, and stochastic variants all performed well. Single layer neural networks also performed well though these had some aspect of human involvement in the selection of features. The success of the other strategy types suggests that a deeper network that incorporates feature engineering would likely also perform well.

In opposition to historical tournament results and community folklore, our results show that complex strategies can be very effective for the IPD. It is not the complexity of strategies that is disadvantageous; rather that directly designing a broadly effective strategy is no easy task. Of all the human-designed strategies in the library, only DBS consistently performs well, and it is substantially more complex than traditional tournament winners like TFT, OmegaTFT, and zero determinant strategies. Furthermore, dealing with noise is difficult for most strategies. Two strategies designed specifically to account for noise, DBS and OmegaTFT, perform well and only DBS performs better than our trained strategies.

Of the strategies trained to maximize their mean score all are generally cooperative, not defecting until the opponent defects. Maximizing for individual performance across a collection of opponents leads to mutual cooperation despite the fact that mutual cooperation is an unstable equilibrium for the prisoner's dilemma. Specifically we note that the reinforcement learning process for maximizing payout does not lead to exploitative zero determinant strategies, which may be a result of the collection of training strategies, many of which retaliate harshly.

Finally, we note that as the library grows, the top performing strategies sometimes shuffle, and are not retrained regularly. Most of the strategies were trained on an earlier version of the library (v2.2.0) that did not include DBS and several other opponents. The precise parameters that are optimal will depend on the pool of opponents. Moreover we have not extensively trained strategies to determine the minimum parameters that are sufficient – neural networks with fewer nodes and features and finite state machines with fewer states may suffice. See [**ashlock2013impact**] for discussion of resource availability for IPD strategies.

Future work: * spatial tournaments and other variants * Additional strategy archetypes by the Ashlocks, e.g. function stacks, binary decision players * further refine features and training parameters

Acknowledgements

This work was performed using the computational facilities of the Advanced Research Computing @ Cardiff (ARCCA) Division, Cardiff University.