ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-209

EA^2: The Winning Strategy for the Inaugural Lemonade Stand Game Tournament

Adam M. Sykulski[1] and **Archie C. Chapman** and **Enrique Munoz de Cote** and **Nicholas R. Jennings**[2]

Abstract. We describe the winning strategy of the inaugural Lemonade Stand Game (LSG) Tournament. The LSG is a repeated symmetric 3–player constant–sum finite horizon game, in which a player chooses a location for their lemonade stand on an island with the aim of being as far as possible from its opponents. To receive a high utility in this game, our strategy, EA^2, attempts to find a suitable partner with which to coordinate and exploit the third player. To do this, we classify the behaviour of our opponents using the history of joint interactions in order to identify the best player to coordinate with and how this coordination should be established. This approach is designed to be adaptive to various types of opponents such that coordination is almost always achieved, which yields consistently high utilities to our agent, as evidenced by the Tournament results and our subsequent experimental analysis. Our strategy models behaviours of its opponents, rather than situations of the game (e.g. game theoretic equilibrium or off equilibrium paths), which makes EA^2 easy to generalize to many other games.

1 INTRODUCTION

Distributed decision–making using learning in repeated games is a growing area of research in the computer science, engineering and economics communities. Such agent–based systems typically assume that the agents are identical, which allows the convergence of their parallel actions to be proven analytically (e.g. [6, 9]). At the same time, however, autonomous agents are being used increasingly in open systems. Although open systems are addressed in fields such as mechanism design, and in specific domains such as poker research, little effort has gone into foundational research on repeated interaction and agent design for open systems.

The Lemonade Stand game (LSG) Tournament[3] was designed to confront this shortcoming within the current multi–agent systems and game theory research literature. In the LSG, each day an agent chooses a location for their lemonade stand on an island from one of 12 possible sites (like the hours on the face of a clock), with the aim of being as far from its competitors as possible so that it attracts the most customers. An agent's payoff for a location is directly proportional to the distance to the other agents. The game is repeated for 100 days — on each day the players choose their locations simultaneously and with no communication. Movement during the day is not allowed; players can only move to new locations on the next day. The Tournament was proposed by Martin Zinkevich, and was constructed with the aim of discovering principles for designing an agent that would directly engage heterogeneous, rather than homogeneous, opponents in a repeated game setting. It provides a setting in which standard analysis is not useful for deriving strategies or refining the set of possible outcomes of the game. In other words, the game is not hard to play because its rules are complex. Rather, it is hard because the agents playing it are themselves hard to play.

This paper describes the winning strategy of the inaugural LSG Tournament. In more detail, the LSG is a symmetric three–player constant–sum game (i.e. in which all players' interests are opposed). It is difficult for a single agent to achieve a high score, because an increase in one player's payoff strictly decreases some other agent's payoff. Indeed, the only way that an agent can earn a high payoff is to tacitly coordinate with one other agent to exploit the third, or to accurately predict the future actions of its opponents. However, because of the LSG Tournament's structure, coordination of the sort that is typically obtained using identical agents playing repeated games is improbable. This is because, as each agent is entered independently of the others, they are unlikely to be similar enough to converge to a Nash or correlated equilibrium via the mechanisms that cause identical agents to converge in standard learning in repeated games results. On the other hand, predicting an opponent's actions seems plausible, but the actions of a boundedly–rational agent may be difficult to model, particularly if your own agent is similarly constrained. Consequently, the LSG Tournament strips away the assumptions of homogeneity and rationality present in much research into learning in games and mechanism design to focus on one key question in artificial intelligence: How to identify how to collaborate with others. By doing so, the LSG provides a framework for investigating domains in which agents can benefit from tacit collusion or coordination, such as auctions or voting problems, and in which agents may currently struggle to identify opportunities to collaborate with others, such as scheduling and planning in open systems.

Staged in January 2010, the LSG Tournament featured teams from universities such as Rutgers, Brown, Carnegie Mellon, Princeton, University of Michigan and UT Austin among others. Each team entered one agent, and the Tournament had a round–robin format. The LSG is connected to research into designing agents to play classic constant–sum games, such as Chess, Go or Poker. However, unlike these games, which have complicated rules and require a degree of skill to play, the LSG has very simple rules, which allows researchers to focus on the problems of collaborating with opponents. The Tournament itself is similar in flavour to the well known *Iterated Prisoner's Dilemma* (IPD) [1], in which two competing agents can collaborate using strategies such as *tit-for-tat* or *Pavlov* to gain higher payoffs. The LSG, however, presents an opportunity to study strategic interactions between competing agents in a scenario without a focal point to coordinate on and containing more than two agents. The Tournament is also comparable to the TAC Market Design Competition (or CAT) [13], in that, due to the structure of the game and since the agents compete directly with each other, there is no optimal solution to the problems they face, because their actions should be dependent on those of their competitors.

[1] Imperial College London, UK, email: adam.sykulski@imperial.ac.uk
[2] University of Southampton, UK, email: {acc,jemc,nrj}@ecs.soton.ac.uk
[3] `http://users.ecs.soton.ac.uk/acc/LSGT/home.html`

Our entry, EA^2, (named after the team–members), incorporates ideas of both learning to coordinate with one other player to exploit the third and accurately predicting future actions. It achieves this by playing strategically in response to the *high level behaviour* of both opponents, rather than their actions per se. Thus, it learns how to play the players in the game (i.e. it learns how its opponents respond to joint actions) rather than the specific patterns of actions that its opponents play. Specifically, EA^2 tries to find a partner with which to collaborate and share a high utility. It does this by measuring the proximity of each opponent to an "ideal", or perfectly-predictable, type using a number of metrics. These ideal types represent strategies that can be used most easily by EA^2 to form a collaboration. Then, using these metrics to capture the behaviour of both opponents, EA^2 selects a partner and a form of collaboration to attempt in the game (e.g. stick in one position, or follow an opponent's lead).

This paper is related to other work on learning in repeated games, and in particular with recent work on planning and learning against heterogeneous algorithms. The field is still new, and most of its literature focuses on how to maximize expected long term reward against specific classes of opponents [11, 2, 4]. The LSG Tournament has motivated new perspectives on designing *universal* algorithms that can handle the more general, and intrinsically more complex scenario where the agent does not *a priori* require the opponents to have any particular structure. As is expected, the entrants to the Tournament (described in more detail later on) were designed to play well against their preferred flavour of opponent but they (to a lesser or greater extent) generalize well across all flavours seen in the Tournament.

Moreover, we expect the general approach used in EA^2 to generalise to a broad range of scenarios. At an abstract level, the algorithm is effectively made of two parts, the first which maps opponents onto an appropriately defined space of ideal types, and the second which maps from the space of joint types to an action. In the future, specific versions of these components may be constructed for specific problems, but more interestingly, general methods for automatically generating each component could be derived from our basic designs (e.g. by automatically computing a set of ideal types for a game, or by learning the mapping from joint types to actions online).

The structure of the paper is as follows. In Section 2 we give background on repeated games and algorithms for playing them. The LSG is introduced in Section 3, together with an analysis of the equilibria and the benefits of coordination between two players. Building on this, in Section 4 we describe EA^2. We conduct a thorough experimental analysis comparison to other entrants to the Tournament in Section 5. Section 6 discusses future work.

2 BACKGROUND

In general, a *non–cooperative game in normal form* is a tuple $\langle \{A_i, u_i\}_{i \in N} \rangle$, consisting of a set of *agents* $N = \{1, \ldots, n\}$, and for each agent $i \in N$, a set of (pure) *actions*, A_i, and a *utility function* u_i. The joint strategy space of the game is given by $A = \times_{i \in N} A_i$, and an agent's utility function is a map $u_i : A \rightarrow \mathbb{R}$. In normal form games, the ith agent, simultaneously with the other agents $-i$, chooses an action from its own action set A_i and, on the basis of the actions performed by all the agents, receives a payoff $u_i(a_i, a_{-i})$, where a_{-i} is the joint action of all the players except player i. Stable points in games are characterised by the set of (pure) *Nash equilibria* (NE), which are defined as those joint strategy profiles, $a^{NE} \in A$, in which no individual agent has an incentive to change its strategy: $u_i(a_i^{NE}, a_{-i}^{NE}) - u_i(a_i, a_{-i}^{NE}) \geq 0, \; \forall \, a_i \in A_i, \; \forall \, i \in N$.

In a *constant-sum* game, the agents' utility functions always sum to the same value, i.e.: $\sum_{i \in N} u_i(a) = c, \; \forall a \in A$. In a *finite-horizon repeated game*, the agents repeatedly play a *stage game*, which is a single normal–form game, for a finite number of iterations. Let $a^t = (a_1^t, a_2^t, ..., a_n^t)$ be the joint action executed at iteration t. In such games, an agent's utility is the sum of its payoffs from the stage games. In the LSG, each stage game is a constant-sum game.

In this paper we focus on games in which an agent can observe the actions chosen by other agents, but does not know their decision–making processes. This differs from traditional approaches to learning in repeated games, which consider agents that play against identical opponents. Nonetheless, in the LSG an agent may still benefit from learning about other's decision–making processes from the history of joint actions. EA^2 and other entries to the LSG Tournament make use of ideas from both of these approaches to playing repeated games using learning algorithms, so we now discuss three algorithms that converge in self-play (i.e. play against identical versions of themselves), and then consider playing against heterogeneous algorithms.

First, *fictitious play* is a learning strategy designed to solve for NE in zero–sum games [3]. In classical fictitious play, an agent's *beliefs* over each of its opponents' actions are given by the frequency with which they have been played in the past. An agent then evaluates its actions by their expected payoff given its beliefs, and plays the highest value. In general, past observations can be discounted, and in the extreme case, if only the most recently observed action is used as beliefs, the resulting decision rule corresponds to the best-response dynamics. Furthermore, fictitious play can be weakened to allow for probabilistic action choices, while still guaranteeing convergence to NE [9]. Second, there are many learning algorithms that use *regret* to choose an action, including *regret-matching* [6], and *internal regret minimisation* [7]. These algorithms use various measures of regret for an action, which is generally measured as the total payoff that would have been accumulated if the agent had played that action at every point in the past. Typically, an agent selects an action from those which have positive regret; however, each algorithm uses variations of this general principle. Third, there are several algorithms that use the reinforcement learning paradigm [12]. Most of their multi-agent variations use past joint actions as states and learn mappings from those states to actions [10, 5], while others use a single state and treat the game as a multi-arm bandit problem [8].

Now we consider approaches to playing optimally against heterogeneous agents. Normally, there is no optimal strategy that is independent of the other agent's strategy (i.e. optimality in multi–agent interactions usually depends on the joint action of agents, and not just the single agent action), therefore, there is no hope in searching for a universal strategy that is optimal against any opponent. What does exist, however, are planning strategies that are optimal against specific classes of opponents. Such is the case of [2] for bounded memory opponents and [4] for unbounded memory opponents.

EA^2 is constructed using a mixture of ideas from the above approaches to playing repeated games, including using measures of the past history of play as a state and techniques for playing optimally against specific classes of opponents. We now describe the specifics of the LSG Tournament in detail, before detailing our entry.

3 THE LEMONADE STAND GAME

We first describe the stage game, before then discussing equilibria and the possibility of beneficial coordinated strategies between pairs of agents.

3.1 The Lemonade Stand stage game

Each day, the three players play the following stage game. There are twelve possible locations to setup a stand on the island's beach,

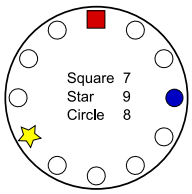


Figure 1. Example Lemonade Stand game.

$A_i = \{1, \ldots, 12\}$, which runs around the entire island. The twelve locations are uniformly spread around the perimeter, like the hours on the face of a clock. The customers are assumed to be distributed uniformly around the island and always purchase from the nearest stand. The utility to a player on a single day is given by the sum of the distance to the nearest player clockwise around the island plus the distance to the nearest player anti-clockwise. Distances are measured in the number of spots between players. If two players are located in the same position, both players receive 6 and the other player receives 12. If all three players are located in the same position then all players receive 8. Therefore, whatever the configuration of the players, their utilities always sum to 24 in each stage game. For example, if the players are located as shown in Figure 1 then 'square' receives utility of 7, 'star' receives 9 and 'circle' receives 8. The objective of the game is for each player to sequentially choose locations for their stand, in order to maximise their aggregate utility over 100 rounds. As such, a player wishes to be located as far as possible from its two opponents in each stage game.

3.2 Game analysis

The stage game of the LSG has many pure NE. In fact, any configuration where the players are located at different positions and all receive a payoff of at least 6 is a NE, and there are also NE in any configuration where two players are located on the same position and the other is exactly opposite. Alternatively, the set of NE can be viewed as all locations for a player that are on or in between the positions directly opposite the other two players. Figure 2 shows the NE locations for a third player, given players square and star are located as shown. For each configuration, the third player's best responses are anywhere in the larger segment between the star and square players, while the best responses that are consistent with a NE are those that are on or in between the positions directly opposite the other two players, as indicated by the arrows. Specifically, in (a) the third player's best responses are anywhere in the larger segment between the star and square players, while the best response in the NE set are between their respective opponent-opposite actions. In (b), where the opponents play opposite one another, the third player is indifferent between all positions, while in (c), where its opponents play on top of one another, the third player is indifferent between all positions except the location of its opponents.

Assuming one player is located at the top of the island, there are 53 possible pure strategy NE. This leads to a plethora of NE in the repeated game, so equilibrium analysis is not sufficient to determine what action to take. As a result, we resort to alternative reasoning.

In particular, for a 3-player game such as this, there is the opportunity for two players to coordinate to minimize the utility of the third player. The constant–sum nature of the game then allows for the two players to gain a higher than average utility. We draw attention to a particular type of coordination between two players, which forms the basis of EA^2. Consider two players that repeatedly sit on opposing sides of the island. The utility of the third player is restricted to 6, which it receives in all of the 12 possible positions – hence all locations are NE, as shown in Figure 2b. Thus, the two collaborating

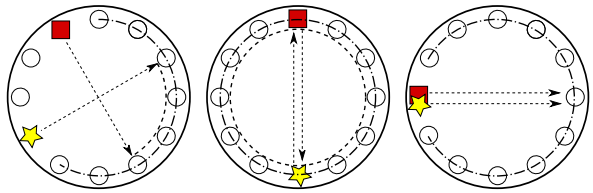


Figure 2. Best-responses for different opponent configurations: The dot-dashed segment indicates the third player's best-response actions, the dashed segment shows best-response actions consistent with a Nash equilibrium, and arrows point to the action opposite each opponent.

players share the remaining utility of 18. For example, a strategy that selects locations randomly (a strategy that is consistent with a mixed NE) over the 100 iterations can be defeated this way – with the two collaborating players receiving an average utility of 9 and the random strategy yielding an average utility of 6. Thus, a strategy that consistently forms such a collaboration frequently receives a high utility.

However, such coordination is difficult to achieve without explicit communication, correlation or pre–game agreements. For this reason, in EA^2 we adopt the approach of building a model of the high–level decision–making behaviour of our opponents, rather than focussing on their specific actions, and use this model to predict their reactions to our current actions. In the next section, we describe how EA^2 carries out this modelling and prediction, and how it uses this to select its actions.

4 THE WINNING STRATEGY

The EA^2 strategy attempts to coordinate with one other player in the LSG by sitting at opposite locations, to consistently score a high utility. It does this by measuring deviation of its opponents from *ideal types*, which are decision–making processes with which it could easily predict and coordinate with. We now discuss these ideal types, then describe how we measure an opponent's deviation from an ideal type. Finally, we describe how EA^2 maps from joint types (i.e. types of both opponents) and game configurations (i.e. current joint action profiles) to select an action in the LSG.

4.1 Ideal types

To begin, there are two obvious ways of initiating a coordinated joint strategy pair in the LSG stage game. First, a player could simply "stick' in the same location, and wait to see if an opponent moves into the location directly opposite. Second, a player could "follow" by moving into a location that is opposite another player. Based on these two patterns of play, we define two ideal types of strategies that an agent could easily coordinate with, as follows:

- A *Stick* strategy, which involves picking a starting location and staying at that spot for the duration of the game.
- A *Follow* strategy, which chooses an opponent and sits directly opposite the position of that opponent at the previous time step. Furthermore, a Follow strategy always chooses to sit opposite a Stick opponent if one is present.

We also define an ideal *Random* strategy, which simply selects a random location on each day, and is therefore impossible to coordinate with. Given this, it is clear to see in a game containing a Stick versus Follow versus Random will yield an average utility of 9, 9 and 6 to each respective player, because Follow will play opposite Stick, as in Figure 2b, and each will earn 9 in expectation. However, it is unlikely that a Stick and Follow strategy will always face such a favourable combination of opponents. Building on this, Table 1 displays average utilities for Stick, Follow and Random against various combinations of ideal type strategies. Notice that Random always performs

equally to or worse than Stick or Follow, although only against a Stick-Follow combination of opponents will a random strategy yield less than average utility. Moreover, which of Stick or Follow is best depends on the combination of opponents. This motivates the fundamental principle behind EA^2, which is flexible in choosing whether to stick or follow depending on the types of opponents it faces.

Table 1. Average utilities for Stick, Follow, Random and EA^2 against various combinations of ideal type opponents (maximum values in bold).

Opponents	Stick	Follow	Random	EA^2
Stick + Follow	**7.5**	6	6	7.5
Stick + Random	8	**9**	8	9
Follow + Random	**9**	8	8	9
2 Stick	8	**9**	8	9
2 Follow	**12**	8	8	12
2 Random	**8**	**8**	**8**	8

4.2 Classifying opponents

EA^2 takes the approach of measuring and classifying the characteristic behaviour of the opponents, based on their closeness to ideal types, and then playing an appropriate strategy in response that aims to form a collaborative partnership. The actions taken are intended to indicate the willingness of EA^2 to coordinate or collaborate with one of the opponents, in order to exploit the third player.

In more detail, EA^2 classifies its opponents by their proximity to playing either a stick or follow strategy, based on their previous actions. An opponent classified as playing a stick strategy is usually slow moving, or stationary, and is hence very predictable. An opponent classified as playing a follow strategy tends to play actions that are within the best response set from the previous time step (or an appropriately discounted average of recent time steps). For example, an agent using fictitious play will often choose locations opposite a stationary player, and is therefore selecting strategies similarly to an ideal Follow strategy. Finally, a player that chooses actions using alternative reasoning, will have low stick and follow indices, and will not be used by EA^2 as a collaborative partner. In what follows, we denote the player using EA^2 as player 0, and its opponents as $\{1, 2\}$, or any pair of agents $\{i, j\}$ when their identities are not needed.

In order to classify its opponents, EA^2 maintains a measure of a stick index, s_i, and a follow index, f_i, for each opponent. It also measures which player i is following using the index f_{ij} (where $j = N \setminus i$). The indices are calculated from the sequence of past actions of each player $A_i = (a_i(1), \ldots, a_i(t-1))$:

$$s_i = -\sum_{k=2}^{t-1} \frac{\gamma^{t-1-k}}{\Gamma} d\left(a_i(k), a_i(k-1)\right)^\rho , \tag{1}$$

$$f_{ij} = -\sum_{k=2}^{t-1} \frac{\gamma^{t-1-k}}{\Gamma} d\left(a_i(k), a_j^*(k-1)\right)^\rho , \tag{2}$$

$$f_i = -\sum_{k=2}^{t-1} \frac{\gamma^{t-1-k}}{\Gamma} \min_{j=N\setminus i} \left[d\left(a_i(k), a_j^*(k-1)\right)\right]^\rho , \tag{3}$$

where $\Gamma = \sum_{k=2}^{t-1} \gamma^{t-1-k}$. The metric $d(a_i(k), a_j(k-1))$ is the minimum distance between player i at time-step k and player j at $k-1$, and $a_j^*(k-1)$ denotes the location opposite from j. In Figure 2, the a^* of each agent are indicated by the arrows. The parameter $\gamma \in (0, 1]$ is the response rate (or discount factor), which exponentially weights past observations – a low response rate (γ close to 1) makes use of a long history of observations, whereas a high response rate corresponds to indices that are more adaptive to capturing sudden changes in the opponent's behaviour. The parameter ρ scales the distances between locations: $\rho < 1$ treats all behaviour that deviates from the ideal types relatively equally, while $\rho > 1$ places more value on behaviour that is close to ideal type behaviour. As such, with a $\rho > 1$, EA^2 can accommodate players that are noisy but select locations close to that of an ideal strategy. Notice that the indices are always negative – the greater the value of the index, the more this player follows the ideal type. An index value of 0, indicates an exact ideal type. The follow index f_{ij} measures whether player i is following player j by looking at the lag–one difference between their respective action sequences. Then the follow index f_i measures whether player i is generally following its opponents.

4.3 Mapping from types to actions

The stick and follow indices are used to classify the types of opponents at each stage of the game. Now, each combination of these observed types lends itself to a specific form of collaboration. Specifically, based on the combination of types, the confidence with which the classification is held, and the current configuration of actions, EA^2 selects an action that attempts to coordinate with one opponent or the other. We now discuss this mapping, and pseudo-code for the full EA^2 algorithm is given in Figure 3 for reference.

In its simplest form, collaboration with a player with a high stick index involves playing the action opposite them on Lemonade Island, while collaboration with a player with a high follow index is done by sitting still and waiting for the player to learn the appropriate response. Although the presence of a third player means that these basic heuristics will likely conflict with one another, the values in Table 1 are used to guide EA^2. Specifically, EA^2 classifies its opponents by comparing their stick and follow indices. Based on their proximity to the ideal types, EA^2 then selects an opponent with which to coordinate, and a particular action to play. This decision is based on the expected utilities to the combinations of ideal types given in Table 1, where the best ideal type is given in bold. We include a column for the expected utility of EA^2 – notice that the strategy identifies the best ideal strategy to play by using the follow and stick indices. The elements of EA^2 that correspond to this reasoning are seen in conditions C1, C2 and C3.1 and (by default) C6 in Figure 3. For example, if opponent i's behaviour is close to that of an ideal Stick strategy, and j's behaviour is not close to either Stick or Follow, then C1 is satisfied and EA^2 chooses to attempt to coordinate by following i by playing $a_i^*(k-1)$. This decision is guided by the expected payoffs given on row 2 of Table 1.

Nevertheless, the utilities in Table 1 are only guaranteed against exact ideal type strategies and collaboration with more sophisticated strategies may be more difficult to initiate and maintain. For example, a pattern of actions that coordinate with one opponent may be successfully broken by the third player by it playing on or near the coordinating opponent, or collaboration can be hard to initiate or maintain against noisy strategies or Follow strategies that choose to follow the other opponent. For these reasons, we have developed two further techniques to improve EA^2's utility against sophisticated opponents. First, if opponent i is following j (indicated through the follow index f_{ij}), but the opponent j is not particularly stationary (so has a low stick index s_j), then EA^2 sits on the previous location of j. In this way, EA^2 interposes in such a way that a coordination with i is encouraged. This element of EA^2 is found in C3.2.

Second, if the two opponents are coordinating and sitting opposite, then EA^2 finds itself in the "sucker" position. In this case, it deploys a "carrot and stick" strategy to break the coordination between its opponents. This involves playing a pattern of actions that are intended to induce one opponent to change its action, and instead coordinate with EA^2. In more detail, first, EA^2 identifies which opponent has a higher follow index and targets this agent with the "car-

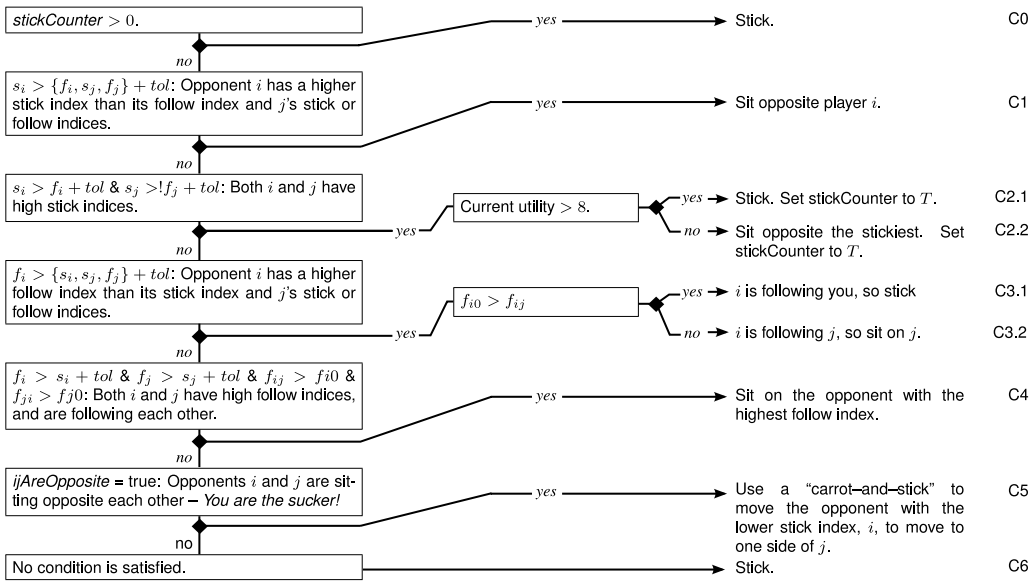


Figure 3. Pseudo-code for EA^2. The parameter tol controls the confidence with which a mapping must be held for a condition to be satisfied.

rot and stick". Second, it divides the action set into two sides based on the position of the third agent, and computes a measure of the bias that an ideal follower would have towards one side or the other. This measure compares a count of the proportion of recent best-responses that occur on either side of the third agent, and EA^2 uses it to decide which direction to try and push the targeted opponent. Third, it plays a pattern of actions that, to begin, discourages the targeted agent from playing directly opposite the third agent by sitting on top of it (the "stick") and later encourages the agent to move into the side identified as having a bias earlier by playing in the opposite side (the "carrot"). The exact pattern used is not important, but it must make the targeted agent's utility lower for sitting opposite the third agent than if it moves to the side and coordinates with EA^2, under the assumption that the targeted agent chooses an action in a similar fashion to an ideal follower. This element of EA^2 is employed if C5 is satisfied in Figure 3.

5 EMPIRICAL RESULTS: LSG TOURNAMENT

In this section we analyse our winning strategy and compare it to the other entries of the LSG Tournament. First, we comment on the final Tournament standings, which are given in Table 2. The Tournament was played in a round–robin format with each triplet combination of agents simulated for several repeats. The Tournament concluded with EA^2 shown to be the winner and Pujara and RL3 awarded a statistical tie for second place. Our parameters for the strategy were: $\gamma = 0.75$ (the response rate), $\rho = 0.5$ (the scale parameter), $tol = 0.1$ (the tolerance of our conditions) and an initial stick of 5 plays. These values are selected (in turn) to make our strategy adaptive to changes in opponent behaviour, to reward ideal type players, to only act when we are confident in our classifications, and to initially not appear too random to our opponents.

There were several interesting entries of note. Pujara is near identical to an ideal Stick strategy, only modified to move to a new location (and stick) when the utility falls below some acceptable level. Waugh, Schapire and FrozenPontiac all used modified versions of fictitious play, and are thus like Follow strategies. RL3 and ACT–R can either stick or follow to coordinate, just like EA^2. However, RL3 has additional heuristics such as trying to initiate a "sandwich attack" and exploit a Stick strategy whereas ACT–R rotates between Stick and Follow depending on their performance. Brown uses Bayesian and regret methods. Finally, Kuhlmann is simply a random strategy selecting between locations uniformly.

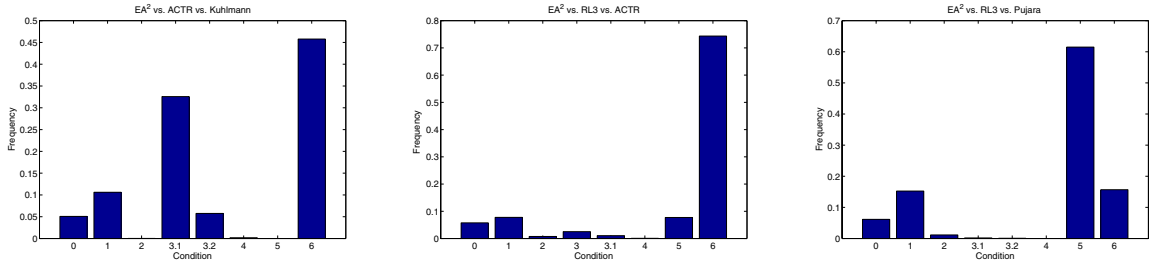
Table 2. Results of the LSG Tournament.

Rank	Strategy	Average Utility	Standard Error
1.	EA^2	8.6233	$\pm$ 0.0098
2.	Pujara (Yahoo! Research)	8.5236	$\pm$ 0.0122
2.	RL3 (Rutgers)	8.5143	$\pm$ 0.0087
4.	Waugh (Carnegie Mellon)	8.2042	$\pm$ 0.0121
5.	ACT–R (Carnegie Mellon)	8.1455	$\pm$ 0.0102
6.	Schapire (Princeton)	7.7519	$\pm$ 0.0110
7.	Brown	7.6746	$\pm$ 0.0084
8.	FrozenPontiac (Michigan)	7.5753	$\pm$ 0.0081
9.	Kuhlmann (UT Austin)	6.9873	$\pm$ 0.0063

Several strategies, therefore select actions similarly to our ideal types – which validates our classification technique. In addition, many of the strategies also explicitly search for a "sit opposite" collaboration. This, in part, explains our positive results – we are adaptive in choosing to stick or follow to collaborate with an opponent. To gain more insight, we also conducted a more thorough analysis against a smaller set of strategies from the Tournament. In particular, we consider Pujara as it is closest to a stick strategy, Schapire as the best follow strategy and Kuhlmann as a perfectly random strategy. We also include results with ACT–R, as it rotates between stick and follow strategies, and RL3 as the best performing strategy that does not fit our ideal types.

Table 3 reports on specific results against the combinations of the above strategies. In all but one combination (Pujara/RL3), EA^2 achieves above average utility. In particular, EA^2 appears to have consistently found a collaborator with which to exploit the third. In these instances, EA^2 scores almost equally with the collaborating opponent. Notice that the random strategy is always exploited, yielding utilities of close to 9 for EA^2, as motivated in the theoretical expected rewards given in Table 1. Finally, since Pujara is closer to an ideal type strategy then ACT–R and Schapire, we choose to collaborate with Pujara as its stick index is the highest of all indices.

To explain these results in more detail, in Figure 4 we show a breakdown of the conditions used in our strategy (as defined in the pseudo-code given in Figure 3) against various opponents. In (a), EA^2 plays against ACT–R and Kuhlmann. In this configuration,



(a) (b) (c)

Figure 4. The frequency of using each condition in the EA^2 pseudo-code given in Figure 3 for various combinations of opponents.

Table 3. Average utility to EA^2 against various combinations of opponents (we also give their closest ideal type). The average utilities of the opponents are also given below (row player first). The winner is denoted in bold type.

	RL3 *no type*	Pujara *stick*	ACT–R *stick/follow*	Schapire *follow*	Kuhlmann *random*
RL3 *no type*	–	6.84 *7.91, **9.23***	**8.64** *8.60, 6.75*	**8.93** *8.93, 6.14*	**8.87** *8.86, 6.27*
Pujara *stick*	–	–	8.80 ***8.84**, 6.36*	**8.85** *8.83, 6.31*	8.93 ***8.95**, 6.13*
ACT–R *stick/follow*	–	–	–	8.68 ***8.79**, 6.54*	8.70 ***8.71**, 6.59*
Schapire *follow*	–	–	–	–	**8.75** *8.74, 6.51*

ACT–R and EA^2 coordinate to both receive a utility of approximately 8.7. This coordination lasts the entire game, and causes the high usage of C6, because s_{ACT-R} and f_{ACT-R} both converge to 0, so none of C0–C5 are satisfied. Therefore this final condition is often used when a partnership has been formed and we wish to stick. Moreover, notice the high usage of both C3.1 (stick to lead an opponent) and C1 (follow an opponent). ACT–R cycles between a Stick and Follow strategy dependent on its performance, so EA^2 adapts to this and chooses to follow or stick to maintain the partnership. In addition, C3.2 is used to force a coordination with ACT–R even if it chooses to follow Kuhlmann. EA^2 is therefore adaptive in collaborating with a dynamic opponent within the same game.

In (b), EA^2 faces RL3 and ACT–R. In this configuration the strategy coordinates with RL3 yielding a reward of around 8.6 to both players. The collaboration is normally initiated through C1, where EA^2 follows RL3. Notice also the occasional usage of C5, which is the carrot and stick strategy. This has been used to attempt to stop collaboration between ACT–R and RL3 — the results indicate that it was successful as coordination with RL3 is almost always achieved.

Finally, in (c) we show the breakdown of conditions against the only losing configuration for our strategy – against Pujara and RL3. A large proportion of actions have been allocated to using the carrot and stick strategy. This indicates that Pujara and RL3 coordinated to sit opposite each other, leaving EA^2 as the "sucker". The carrot and stick strategy attempts to dislodge RL3 from the partnership with Pujara, but it is not successful. It appears that RL3 is simply quicker to move and sit opposite Pujara, and in so doing has exposed a weakness in our strategy – our action choice in the initial steps of the game. Thus, EA^2's initial behaviour is a weak aspect of the strategy, and can be improved for future versions of the Tournament.

The analysis in this section indicates that EA^2 consistently finds a suitable partner with which to coordinate, and then subsequently maintain this partnership for the remainder of the game. By so doing, it yields above average utilities against all combinations of opponents except one. Moreover, we have demonstrated that EA^2 is also able to coordinate with a dynamic partner that cycles between strategies.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we described the winning strategy for the LSG Tournament. To complement this, we also performed a more in depth analysis of EA^2's performance against other entrants to the Tournament. In particular, we demonstrated how EA^2 adapts to playing different ideal types depending on the opponents faced, which explains why it yielded the highest average utility of all entrants.

In future work, we intend to use the same principle of classifying and responding to our opponent's high–level behaviour in future versions of the Tournament. This will involve improving the ideal type classification technique and refining the mapping from joint types to actions. Beyond this, we expect the general approach used in EA^2 to generalise to a broad range of scenarios. Specifically, the algorithm's structure is comprised of two parts, a mapping from opponents' behaviour onto an appropriately defined space of ideal types, and a mapping from the space of joint types to an action. Given this structure, a key question we will pursue is whether general methods for automatically generating each of these components can be derived.

Acknowledgements This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Networks) project and is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

REFERENCES

[1] R. Axelrod, *The evolution of cooperation*, Basic Books, 1984.
[2] B. Banerjee and J. Peng, 'Efficient learning of multi-step best response', in *Proc. of AAMAS '05*, pp. 60–66, (2005).
[3] G. W. Brown, 'Iterative solution of games by fictitious play', in *Activity Analysis of Production and Allocation*, ed., T. C. Koopmans, 374–376, Wiley, New York, (1951).
[4] E. Munoz de Cote and N. R. Jennings, 'Planning against fictitious players in repeated normal form games', in *Proc. of AAMAS '10*, (2010).
[5] E. Munoz de Cote, A. Lazaric, and M. Restelli, 'Learning to cooperate in multi-agent social dilemmas', in *Proc. of AAMAS '06*, pp. 783–785, (2006).
[6] S. Hart and A. Mas-Colell, 'A simple adaptive procedure leading to correlated equilibrium', *Econometrica*, **68**, 1127–1150, (2000).
[7] S. Hart and A. Mas-Colell, 'A reinforcement procedure leading to correlated equilibrium', in *Economic Essays: A Festschrift for Werner Hildenbrand*, 181–200, Springer, New York, NY, USA, (2001).
[8] D. S. Leslie and E. J. Collins, 'Individual Q-learning in normal form games', *SIAM Journal, Control and Optimization*, **44**, 495–514, (2005).
[9] D. S. Leslie and E. J. Collins, 'Generalised weakened fictitious play', *Games and Economic Behavior*, **56**, 285–298, (2006).
[10] T. Sandholm and R. H. Crites, 'Multiagent reinforcement learning in the iterated prisoner's dilemma', *Biosystems*, **37**, 147–146, (1995).
[11] Y. Shoham, R. Powers, and T. Grenager, 'If multi-agent learning is the answer, what is the question?', *Artificial Intelligence*, **171**(7), 365–377, (2007).
[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
[13] P. Vytelingum, I. A. Vetsikas, B. Shi, and N. R. Jennings, 'The winning strategy for the TAC market design competition', in *Proc. of ECAI '08*, pp. 428–432, (2008).