Using Reasoning Patterns to Help Humans Solve Complex Games

Dimitrios Antos and **Avi Pfeffer**
School of Engineering and Applied Sciences
Harvard University
{dantos, avi}@eecs.harvard.edu

Abstract

We propose a novel method for helping humans make good decisions in complex games, for which common equilibrium solutions may be too difficult to compute or not relevant. Our method leverages and augments humans' natural use of arguments in the decision making process. We believe that, if computers were capable of generating similar arguments from the mathematical description of a game, and presented those to a human decision maker, the synergies would result in better performance overall. The theory of reasoning patterns naturally lends itself to such a use. We use reasoning patterns to derive localized evaluation functions for each decision in a game, then present their output to humans. We have implemented this approach in a repeated principal-agent game, and used it to generate advice given to subjects. Experimental results show that humans who received advice performed better than those who did not.

1 Introduction

We are concerned with helping people make good decisions in complex games. A natural and standard way to do that is to compute a Nash equilibrium of a game and recommend it to the human decision maker. This approach has several drawbacks: First, for many games, and in the general case, it is very hard to compute Nash. Second, for most games of interest (e.g., repeated games) there is a multitude of equilibria, and it is hard to know what equilibrium to suggest to the human. Third, a Nash equilibrium is optimal with respect to ideal "rational" play, but not necessarily with respect to the actual play of the other players. For example, in the rock-paper-scissors tournaments Nash equilibrium players came in the middle of the pack [Billings, 2000]. A Nash equilibrium adviser for rock-paper-scissors would not be very useful.

The most fundamental problem with this approach, however, is that it takes the human out of the decision making process. If a human decision maker bears ultimate responsibility for the outcome of the decision, he or she may want some say in making it. In a recent study [Cason and Sharma, 2007], researchers recommended correlated equilibria to subjects in a two-player game. They found that subjects were reluctant to follow the recommendations, *even when they knew the strategy the other player was being recommended to play*. How much more this would be the case when the human has no reason to believe the other player will be playing the same Nash equilibrium strategy that is being suggested to them.

Furthermore, humans have judgment, insight and intuition that are not available to the computer in its equilibrium analysis. Computers are capable of fast execution of algorithms, which allows them to excel in quantitative analyses. In games, defined as sets of agents, strategies and utility functions, a computer will be able to evaluate probability distributions over outcomes, expectations over the players' utilities, and employ numerous techniques to help them maximize their payoffs. On the other hand, humans have good intuitions as to what models or strategies their opponents are most likely to construct and follow. Because people have reasonably accurate beliefs about how other people tend to behave, they gain the edge over computers in identifying what is "reasonable" and "natural." In addition, humans may be able to cut through complex games to identify key patterns and simplifications that lead to good strategies.

In this paper we present an approach for leveraging and augmenting the decision making capabilities of humans in games. Our approach is to generate and present arguments in favor of different strategies to the human player. We believe that a process of argumentation is a natural part of the human decision-making mechanism. By "argumenation" we mean that people need to *justify* their choices to themselves. In other words, they need to answer "why" a particular strategy is good, i.e., what will be accomplished by it, how it will affect other agents' choices in the future, how it is going to benefit them, what the risks are, et cetera. For example, in a repeated ultimatum game the proposer might think as follows: "If I give the responder a very small fraction of the resource, she may believe I am selfish and thus choose to punish me in subsequent rounds; if, one the other hand, I am very generous, I may needlessly compromise my own per-round gain."

There are two ways a computer might assist such an argumentation process. One is by quantifying arguments, providing numeric estimates of the benefits and risks of particular strategies. Even when humans are aware of an argument, such numeric arguments may be hard for the human to quantify. For example, in the ultimatum game, how much will a player lose in future rounds if the opponent thinks he is selfish? The

second way is by suggesting arguments to the human that the human may not have thought of at all. This is particularly the case in more complex games where it is hard to identify all the relevant arguments.

In this work, we employ the theory of reasoning patterns [Pfeffer and Gal, 2007] to generate arguments for and against each action that the agents take in a particular game. We then present these arguments to human subjects as "advice" in a controlled experiment, while other subjects in the same experimental session receive no such advice. There is both a qualitative and a quantitative component in this advice. If our reasoning is correct, one would expect players with access to the advice to perform significantly better on the average than their fellow players with no such access.

Our results indicate that, indeed, performance can be boosted. In the experiment we have conducted, subjects had to play a repeated *principal-agent game* with each other. We observed on the average 37% higher scores from subjects who actively used the advice option, compared to those who either did not have access or chose not to use the advice.

The paper is organized as follows. Section 2 describes the principal-agent game that our experiment employed. In section 3 we present the theory of reasoning patterns and how it has been used to generate arguments in our game. Next, section 4 discusses the details of our advice-giving methodology, while section 5 presents the results of our experiment. The paper concludes with a discussion and future extensions.

2 The principal-agent game

To test our hypothesis that arguments provided to humans as advice are beneficial, we have constructed a simple, yet subtle game. Throughout the rest of the paper this will be referred to as the *principal-agent (p-a) game*, due to its resemblance with similar games in the economics literature. In those games, one player, called the "principal," must choose whether to hire, and how much to pay, another player, called the "agent." The agent has a *type*, which is hidden from the principal. The agent's choice consists of selecting an effort level. Both players' payoffs depend on the actions of the agent (assuming she was hired) and the payment exchanged. Notice that this generic model may serve as a good approximation for many real-life situations, such as hiring employees, constructing incentive-compatible payment schemes, etc.

Our version of the principal-agent game follows the same structure. The principal and the agent are both initially given some resources, which are private information to them. At first, the principal may choose to transfer part (or all) of his resources to the agent (his transfer may also be $\emptyset$). Then the agent may choose to spend some of her[1] resources to move towards a "goal." The principal's payoff is increasing in the amount of resource held in the end of the game, as well as the closeness of the agent to the goal. The agent's payoff is, however, dependent upon her "type." Agents can either be "resource-lovers," who value resources but not distance from the goal, or "goal-lovers," who value closeness to the goal but not resources. Clearly, the incentives of goal-lovers are aligned with those of principals, while those of resource-lovers are not.

In practice we use the repeated version of the above game. In each *round*, the game is reset, but the same principal is paired again with the same agent, whose type remains unchanged. This allows for nuanced behavior to form: first, the principal may form a belief over the agent's type and update it after every round, by observing her actions; second, the agent may find it beneficial to change her behavior to reveal or mask her true type, such that the principal may take actions in subsequent rounds that improve her score. We say that the agent thus manipulates the principal's beliefs. As is standard, to prevent players from reasoning backwards from the end of the game, we let the number of rounds be undetermined, with a 10% probability of ending the game after each round.

Why did we choose this particular game? There are many reasons: For one, it is difficult to solve it analytically. There is no simple way to model the principal's belief update function, which maps current beliefs and observed actions to updated beliefs. This is because the probability of an agent taking an action does not just depend on her type and her (privately known) resources, but also on the degree in (and direction to) which she has chosen to be manipulating the principal's beliefs in this round. A similar argument holds for the agent: to be able to reason about how the principal will interpret any one of her actions, she must be aware of the model the principal has constructed of her strategy. Technically, this is a Bayesian repeated game, in which types are constant throughout all stage games, and each stage game has imperfect information. Such a repeated game is very hard to solve. An easier class of games are repeated games of incomplete information, in which each stage is a separate Bayesian game, but even for that class an impossibility result has been proved for Bayesian learning [Nachbar, 2001]. Moreover, since this is a repeated game, one would expect a multitude of equilibria, alluded to by the folk theorems. In addition, people's play in this game may deviate significantly from equilibrium. Thus the traditional game theoretic approach suffers from the disadvantages outlined in the introduction: equilibria are hard to compute, numerous, and it is not clear that they are relevant.

Yet, despite its complexity, the principal-agent game has some undeniable appeal. Although optimal strategies are difficult to compute, something can be said of what might constitute a "good" or "reasonable" strategy. First, both players' scores in any given round depend directly on their actions: for the principal, a larger transfer is costlier, and for the agent the move affects her score either by influencing distance (for the goal-lover) or resources possessed (for the resource-lover). Second, the goal-lover agent has an incentive to perform moves that reveal her type more effectively to the principal. This is because, if the principal becomes convinced that the agent is a goal-lover sooner, he might transfer larger amounts of resource in more future rounds. Conversely, the resource-lover has an incentive to hide her type, by not performing moves that are "characteristic" of resource-lovers (e.g., not moving), so as to maintain "useful doubt" in the principal's mind. Therefore, the game is a good example of what is *technically hard* but *intuitively approachable*.

[1] Throughout the paper we refer to the principal and the agent with masculine and feminine pronouns, respectively.

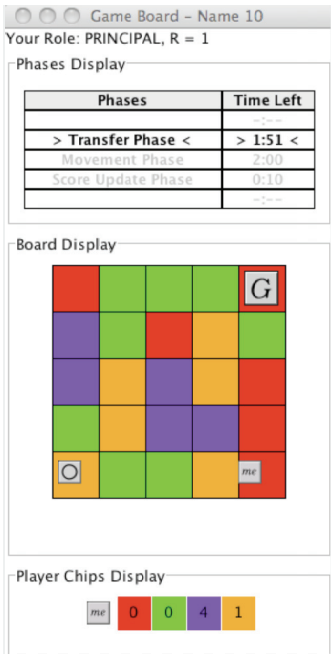

Figure 1: The user interface of the CT game

2.1 Experiment implementation

For our experiment, we have implemented the principal-agent game in the Colored Trails (CT) framework [Grosz *et al.*, 2004], a client-server architecture system for studying multi-agent decision making. CT is, in the simplest sense, a board game played over a computer network. In our experiment there was a 5×5 board of colored squares (see Figure 1). The squares were uniformly and independently colored in each round with a palette of four colors (red, green, orange and purple). There were two players, the principal and the agent. The agent was either a "goal-lover" or a "chip-lover" with probability 0.5. The agent was positioned in the lower-left corner (coordinates (4,0)). At the same time, a goal square was always placed in the upper-right corner (0,4). Both players in each round were given five chips at random (i.e., any combination of colors adding up to 5 chips was equally likely), and each player could only see his/her own chips, not his/her opponent's. These chips could be used to move on the board: for the agent to move on a red square, for example, she had to give up a red chip.

Each round was then played in two phases: in the *transfer* phase, lasting 2 minutes, the principal could initiate a transfer of chips. The principal was allowed to transfer any subset of his/her chips, from the null transfer to everything. As soon as the transfer was carried out (or the 2 minutes expired), the round moved on to the *movement* phase, in which the agent could drag her icon on the board and move it (no diagonal moves were allowed), spending chips in the process. After the movement phase the round was ended. Players would get scores based on their actions (in a so-called *score update* phase), and a new round would begin between the same principal and the same agent with probability 90%. With the remaining 10% the game would end and, when all games between all subjects ended, the subjects were again randomly re-paired. Thus, for example, a subject who was a principal in the first game might assume the role of a goal-lover agent in the second game. Subjects were fully aware that the identity of their opponent would change between games, but not between rounds of the same game.

More specifically, the principal's scoring function was $\pi_p(c_p, c_t, x) = 50(|c_p| - |c_t|) + 65(8 - dist(x, G))$, where x is the position the agent has chosen to move to, c_p and c_t are the initial chipset given to the principal and the transfer made, $|\cdot|$ denotes set size, G is the goal position and $dist(\cdot, \cdot)$ measures the Manhattan distance between two points on the board. The scoring functions for the goal-lover agent was $\pi_g(x) = 20(8 - dist(x, G)) + 250 \cdot \mathrm{I}[x = G]$, where $\mathrm{I}[\cdot]$ is the indicator function. Finally the chip-lover gained $\pi_c(c_a, c_t, p) = 20(|c_a| + |c_t| - chips(p))$, where c_a is the agent's original chipset and $chips(p)$ denotes the chipset required to move from the original position along path p. All scoring functions were common knowledge to the subject pool.

Subjects were paid with real money according to their scores, as compared to those of other players. Scores were normalized within a single type, such that a subject who happened to play as a principal and a chip-lover was paid as a function of how high his/her score was compared to the average principal and the average chip-lover. Thus, no player could feel he/she was unfairly treated because he/she happened to be of one particular type more frequently, or because he/she assumed fewer types throughout the experiment. Proper procedures for teaching subjects how to play the game, letting them play a few test rounds, answering their questions and debriefing them in the end were followed as well.

3 The reasoning patterns

The theory of reasoning patterns was first presented in [Pfeffer and Gal, 2007]. It provides a succinct and comprehensive coverage of all the ways in which any player (in any game) may go about obtaining his/her preferred outcomes. The theory states that, if one restricts the strategy space of agents to a reasonable subset, there are just four ways to reason about making a decision. The reasoning patterns are defined as sets of paths in a multi-agent influence diagram (MAID) [Koller and Milch, 2003]. A MAID is a compact, graphical representation of a game. It contains three kinds of nodes: chance nodes, shown as ellipses, represent random variables in the environment; decision nodes, shown as rectangles, represent decisions the agents make; and utility nodes, shown as diamonds, represent the utilities of agents. Each decision or utility node is associated with a particular agent. Edges going into chance nodes or utility nodes indicate probabilistic dependence, similar to Bayesian networks. An edge going from a node X into a decision node Y represents the fact that the agent making decision Y knows the state of X at the time of making her decision. The reasoning patterns characterize the way information is created and manipulated in such a game. The four reasoning patterns are as follows: Each reasoning pattern is illustrated by its canonical representation in

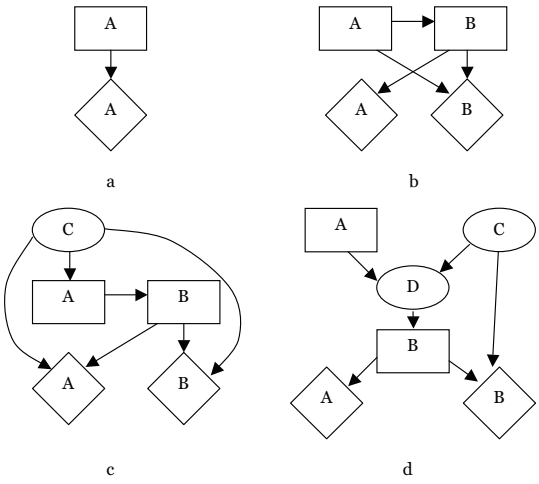

Figure 2: Examples of reasoning patterns

a MAID, shown in Figure 2.

- (a) *Direct effect:* This reasoning pattern captures cases where a player may take an action that will directly affect her utility. "Directly" here does not mean deterministically, or even that this action is the only one determining her utility. Rather, it means that no other player's decision intervenes between the action and its effect.
- (b) *Manipulation:* This pattern describes cases where a player A may take an action that will be visible to some other player B in the future and also affects B's utility. This way, A may alter the decision-making problem of B, who might be induced to take an action that affects A's utility favorably.
- (c) *Signaling:* This pattern captures transmitting information to another player. If A has access to a variable C that another player B cares about (i.e., B would like to know C's value to make a better decision), then A may take an action that B will observe and then infer something about C. Of course, A can use this to signal a value for C that will induce desired behavior out of B.
- (d) *Revealing-denying:* This last pattern describes situations where an agent A may take an action that can induce another agent B to more (or less) clearly observe the value of a variable C she cares about. Unlike in signaling, A does not himself know the value of C. Depending on whether B is more likely to take an action that is beneficial to A's utility when C's value is inferred with confidence, A has an incentive to either assist or obstruct this inference. (In the canonical representation, A may affect how much of C is inferable by B from observing D.) Note that here players A and B might be the same person—in this case the reasoning pattern describes information gathering (or exploration).

These four patterns are "complete," in the sense that a reasoning pattern exists for all decisions of every player in every game, unless the decision is "unmotivated," meaning the player stands to obtain the same exact utility in expectation whatever action she takes (i.e., it does not matter what she decides). Each reasoning pattern is defined formally as a set of paths in a MAID. The set of reasoning patterns for a MAID can be discovered in time polynomial in the size of the MAID and the number of reasoning patterns [Antos and Pfeffer, 2008].

3.1 Using reasoning patterns for argumentation

Reasoning patterns are naturally appealing for our purposes, which are to generate semantically rich, meaningful, hopefully intuitive arguments for how to play a game from its mathematical description. The main advantage of reasoning patterns is that, while they are deeply grounded in theory and easy to discover and quantify by a computer, at the same time they are describable in terms of strategies, even "stories" understandable by humans, e.g., "if you do x then this other player will want to choose y, which will boost your utility by k points."

After identifying the reasoning patterns in a game, the next step is to transform them into arguments. The idea is that every reasoning pattern r for decision d of a player could be represented with a scoring function $v_r : A_d \rightarrow \mathbb{R}$, where A_d is the set of actions available to the player in d. Intuitively, higher values of v_r mean better choices. This scoring function is constructed to be *localized* in nature, i.e., ignoring portions of the game that lie outside its main description. Technically, the portion of the MAID graph that lies outside the paths forming the reasoning pattern is ignored or consolidated. In order to achieve this, however, we need to make assumptions about the strategy used for decisions outside the reasoning pattern. The format of the scoring functions we propose for each of the patterns are as follows:

- *Direct effect:* We score an action with respect to direct effect in decision d by summing over the expected payoff the agent stands to make in all the utility nodes within the MAID that descend from that agent's decision node d, and which are not blocked by other agents' decision nodes. This measures exactly what direct effect captures: what the agent can accomplish by acting on her own, without depending on other agents' decisions.
- *Manipulation:* The scoring function for manipulation (of player A to player B) measures the expected increase in utility obtained due to B taking an action because she has observed A's decision. This increase is usually computed with respect to a reference action of A, which can be chosen arbitrarily, since it serves only for comparison purposes.
- *Signaling:* When A signals C to B we compute the extra utility obtained by A by causing B to update his probability distribution over C, and thus change the probability of his actions that affect A's utility.
- *Revealing-denying:* Similarly, we score an action with respect to revealing-denying by computing the incremental utility the player obtains by causing another's belief to be updated in a particular way.

Each such scoring function represents an argument. Different arguments may be computed for a particular decision.

Thus, if there are two reasoning patterns r_1 and r_2 of d, it might be the case that a particular action $a \in A_d$ fares well in v_{r_1} but poorly in v_{r_2}. This is because a might be a good action in the portion of the game captured by r_1, but not so good in the portion described by r_2. We note also the the total utility to the agent of taking a is not simply the sum $v_{r_1}(a)+v_{r_2}(a)$, if the MAID portions of r_1 and r_2 are overlapping.

3.2 Reasoning patterns in the p-a game

The MAID for the p-a game consists of connected copies of a single-round MAID. The MAID for two rounds of the p-a game is shown in Figure 3. In each round i, there are two decision nodes, P_i, the principal's choice of a chip transfer, and A_i, the agent's choice of a movement path. There are also four chance nodes, B_i, the board coloring (known to both players), C_i^p and C_i^a, the principal's and the agent's chipsets (known to their respective players), and T, the agent's type. Notice that the agent's type is not subscripted, because it remains the same throughout the game. Between rounds there are two types of arrows: *(i)* observation arrows, e.g., an arrow from A_{i-1} to P_i, meaning that the principal in round i has observed the agent's action in the previous round, and *(ii)* no-forgetting arrows, e.g., from P_{i-1} to P_i, meaning that the principal has not forgotten his action in the past.

Since the MAID for this game is infinite in size, we can only work with truncated, finite versions of it. We implemented the algorithm from [Antos and Pfeffer, 2008] to discover the reasoning patterns in a MAID covering the first 4 rounds of the game, and we got the following results:

1. *Direct effect for P_i:* For all rounds i, the principal's decision P_i has direct effect, meaning that his choice of a transfer affects his score in round i. This makes sense, since transferring more chips negatively impacts his score.

2. *Direct effect for A_i:* For all rounds i, again, the agent's decision A_i has direct effect, as her movement affects her score, e.g., if she is a chip-lover, spending chips to move decreases her current-round score.

3. *Manipulation for P_i to A_i:* For all rounds i, the principal manipulates the agent. This captures the fact that, if the principal transfers more chips to a goal-lover, the principal may allow her to move closer to the goal, and thus positively affect the principal's utility (which is decreasing in the distance between the agent and the goal).

4. *Signaling for A_i to P_{i+1}:* In every round, the agent makes an action that the principal will observe and update his belief over her type. His next-round transfer will then be decided under this updated belief and may be crucially affected by it. Therefore, the agent has an incentive to convince the principal that she is a goal-lover—it is easy to observe that it's safer for the principal to transfer more chips to the agent when he believes she is a goal-lover. Thus an actual goal-lover may want to move as close to the goal as possible, and a chip-lover may want to avoid total immobility, to maintain some "useful doubt" in the principal's mind.

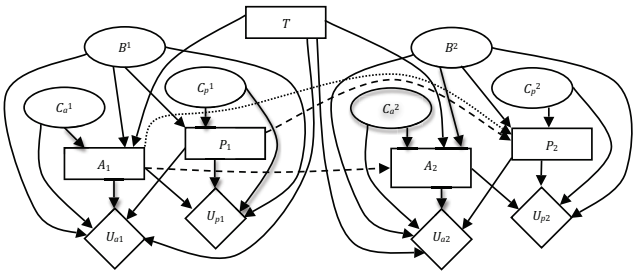

Figure 3: Two rounds of the p-a game

5. *Revealing for P_i to P_{i+1}:* In every round the principal's transfer serves two purposes: On the one hand, it allows the agent to move closer to the goal in that same round. On the other hand, it is a useful "exploration tool." In particular, by making a transfer and observing the agent's response, the principal updates his belief over her type. Some transfers are more helpful than others in that latter respect. We say that "the principal reveals the agent's type to himself in the future."

4 Generating advice

To generate advice for the p-a game, we use the five reasoning patterns applicable to every round of the game and the scoring functions we defined for each pattern. In particular, we compute $v_{r_i}(\cdot)$ for every pattern r_i of the principal (agent), and we output these values to the human subject acting as the principal (agent), along with natural language text explaining what these numbers mean. In this advice there is no direct suggestion for a "good" action the player would be better off taking. Our point is to "educate" the subjects with the intuition gained by examining the reasoning patterns, and then let them take over the task of meaningfully weighing them, combining them and reaching a final decision. Also, both players may seek advice while playing the game, but only for a particular action (transfer or move). For example, the principal can use the advisor interface to ask "what do you think about a transfer of 1 blue chip and 2 green?" We have disallowed vague questions of the "what do you think?" variety.

The advice for the principal consists of a general and a transfer-specific part. The general part explains the three relevant reasoning patterns. It also automates his belief update task, mentioning every time advice is sought what the probability of the agent being a goal-lover is. The specific part mentions (*i*) the cost of the transfer, (*ii*) the expected gain in points due to the agent moving because of the transferred chips (compared to making a null transfer), and (*iii*) the extra points expected to be gained in the next round due to a more refined distribution over the agent's type (exploration).

Similarly, the agent receives general-purpose advice, which explains that her move will be used by the principal to reason about her type, and that a chip-lover should try to use at least the chips transferred to her to the extent that the board color configuration allows (otherwise the principal will be sure of her true type and will rationally give her nothing in subsequent rounds). The move-specific advice mentions:

(*i*) the score obtained in that round by making that particular move, and (*ii*) the extra points expected to be gained in the next round due to controlling the principal's belief with that move (compared to not moving).

4.1 Strategy Assumptions

As mentioned earlier, in order to develop the scoring functions we need to make assumptions about the strategy used for decisions outside the reasoning patterns. Furthermore, if we are to update the principal's beliefs about the agent's type based on the agent's actions, we need to know what strategy the agent would use for each type. We argued earlier that the p-a game is difficult to solve, so how do we come up with these strategies?

To circumvent the problem, we plug in some approximations for the players' strategies. In particular, we define $\hat{\sigma}_p$, $\hat{\sigma}_a$ for the principal and the agent, respectively, to be *myopic quantal response strategies*, and use those whenever we need to make calculations using the other agent's strategy. This involves two assumptions: (1) the agents myopically only consider their utility in the current round when considering their play; and (2) they play a quantal response. In a quantal response equilibrium [McKelvey and Palfrey, 1995], an agent determines the utility $\pi(a)$ for each action a, given the agent's beliefs about the other agents' strategies. The agent then chooses action a with probability $\frac{e^{\lambda\pi(a)}}{\sum_{a'} e^{\lambda\pi(a')}}$, where λ is a parameter. As $\lambda \to \infty$, the agent is completely rational and chooses its best response with probability 1. If $\lambda = 0$ the agent chooses uniformly at random. Thus by choosing λ we can control to what degree agents choose their best scoring action. We set $\lambda = 0.5$ in the above formula, because it well explained the actual behavior of humans in a test run of the experiment.

Now, one might argue that the advice given is only as good as the assumptions made about strategies used in computing the scoring functions. That is certainly true. However, we did not go to great lengths to be particularly clever or realistic about the strategies, and nevertheless we got good results. We believe that the combination of myopic strategies (or at least strategies with short look-ahead) and quantal response may work well in many games. Quantal response is important because it smooths out any errors made in the myopic assumption. For example, myopic best response would tell a chip-loving agent never to move. Quantal response smooths this out so that the chip-loving agent moves, at least a small amount, a good deal of the time. This is important because then, if the principal observes that the agent moved a little bit, he can still place high probability on the agent being a chip-lover. If the principal believed the agent was purely myopic, then after the agent moved a little bit the principal would believe she is definitely a goal-lover.

Secondly, we need to clarify that, whenever a subject asked for advice, our algorithm had to perform several computations, including updating probability distributions and calculating expectations of values. Whenever possible, these computations were performed by exact inference or closed-form solutions. In cases where that was infeasible, however, sampling was used. For instance, to compute the expected number of points the principal could expect when dealing with an agent who was believed to be a goal-lover with some probability, the algorithm sampled several board color configurations, principal and agent chipsets for the next round, and then responses from both players according to $\hat{\sigma}_p$ and $\hat{\sigma}_a$, and then computed payoffs.

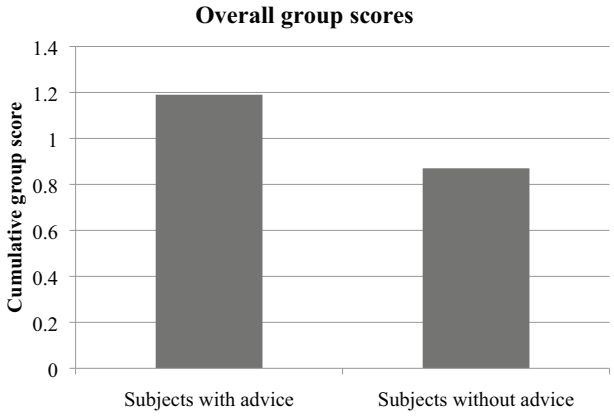

Figure 4: Cumulative scores

5 Experimental results

The experiment was performed with 18 subjects in two sessions (one with 10 and one with 8 subjects). Half the subjects in each session had access to the advice, while the remaining half did not, but were aware of some of their opponents being able to access it. Subjects were randomly paired and assumed different roles between games, but advice-seeking was either on or off for a particular subject throughout the experiment.

A subject's performance was calculated as follows: Suppose $R = \{p, g, c\}$, where p, g, c stand for "principal," "goal-lover," and "chip-lover." If subject i assumed roles $R_i = \{p, c\}$ during the experiment, and made an average π_i^p and π_i^c points respectively, his score is defined as $s_i = \frac{1}{2}(\frac{\pi_i^p}{\bar{\pi}^p} + \frac{\pi_i^c}{\bar{\pi}^c})$, where $\bar{\pi}^t$ is the average score of all players of type t. In general, the performance of a player i is defined as

$$s_i = \frac{1}{|R_i|} \sum_{t \in R_i} \frac{\pi_i^t}{\bar{\pi}^t}$$

Similarly, the performance of a group G containing the set of agents $A(G)$ is defined as

$$s_G = \frac{1}{|A(G)|} \sum_{i \in A(G)} s_i$$

Then let A, A^C be the two groups of subjects, those playing with advice and those not having access to it. Our experiment measured that $s_A = 1.19$ and $s_{A^C} = 0.87$, a performance boost equal to 37% from taking advice (see Figure 4). This result is statistically significant at the 1% level.

One other interesting result is that, if we examine the payoffs of players in A^C, but we take two cases: when the player was paired with someone also in A^C and when the player

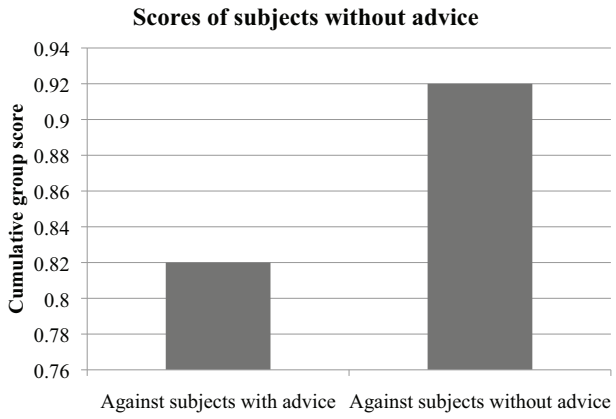

Figure 5: Cumulative scores of agents without advice

was paired with someone in A, we observe that in the first case subjects achieve an average of 11% higher scores (see Figure 5). This implies that our game is "competitive" with respect to advice-giving, in the sense that a subject is always better off using the advice, regardless of whether his/her opponent also has access to it.

One could also examine the "raw" in-game scores subjects attained, instead of their normalized payoffs. This serves to answer whether *social welfare* increases. (Observe that, if the advice causes both players' payoffs to increase by, say, 20%, then this will not be reflected in the normalized scores.) However, we do not observe that social welfare, defined this way, generally increases among subjects using the advice (see Table 1).

	principal	goal-lover	chip-lover
A against A	297.5	20.5	130
A against A^C	605	60.5	150
A^C against A	297.5	140	100
A^C against A^C	295	60	130

Table 1: Average (raw) scores

6 Discussion and Conclusion

We have shown that people who were given advice in the form of arguments based on reasoning patterns did better than people who were not given advice. There are two main questions that need to be discussed with respect to these results. First, how can the performance gap be attributed solely to the quality of the advice? And second, how do these results generalize to other games or situations?

The first question is important if one considers possible side-effects of the advice-giving mechanism. For example, one might claim that, for subjects that could request advice, just the fact that they spent more time on their decisions—even if the "content" of the advice was not helpful—improved their performance. Alternatively, the gap might be explained away from user interface modifications (e.g., that the advice-giving interface was in some sense "smart" and led subjects to better actions, without the advice being of any use per se). To ensure as much as possible that our GUI did not interfere we used the simplest possible form, a bare popup window with just text, no graphics and no explicit interaction within the advice window. The text contained the general-purpose advice (as explained in section 4) in one paragraph, followed by the transfer- (or move-) specific advice in one sentence for each relevant reasoning pattern. We also kept the phase time limit to 2 minutes for subjects who got the advice, so that any potential benefit from spending more effort in the decision-making process would be outweighed by less time to actually make the decision.

As to the second concern, we treat this first experiment as a first step in exploring the possibilities of our method. Although no claim can be made of its universality, we are confident that its usefulness is not restricted to the principal-agent game of our experiment, but is extensible—albeit not effortlessly—to other games where arguments can be similarly identified. In future work we plan to further develop and formalize our technique for quantifying and combining reasoning patterns to give more nuanced advice and maybe identify good and intuitive strategies automatically, without the intervention of humans, for direct use by computer agents.

Acknowledgements

This work has been supported by MURI grant FA9550-05-1-032, award number 5710001863.

References

[Antos and Pfeffer, 2008] D. Antos and A. Pfeffer. Identifying reasoning patterns in games. In *Uncertainty in Artificial Intelligence*, 2008.

[Billings, 2000] D. Billings. The first international RoShamBo programming competition. *International Computer Games Association Journal*, 23(1):3–8, 2000.

[Cason and Sharma, 2007] T. N. Cason and T. Sharma. Recommended play and correlated equilibria: a case study. *Economic Theory*, 33(1):11–27, 2007.

[Grosz *et al.*, 2004] B. Grosz, S. Kraus, S. Talman, and B. Stossel. The influence of social dependencies on decision making: Initial investigations with a new game. In *Autonomous Agents and Multi-Agent Systems*, 2004.

[Koller and Milch, 2003] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):118–221, 2003.

[McKelvey and Palfrey, 1995] R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10:6–38, 1995.

[Nachbar, 2001] John H. Nachbar. Bayesian learning in repeated games of incomplete information. *Social Choice and Welfare*, 18(2):303–326, 2001.

[Pfeffer and Gal, 2007] A. Pfeffer and Y. Gal. On the reasoning patterns of agents in games. In *National Conference on Artificial Intelligence (AAAI)*, 2007.