Providing a Recommended Trading Agent to a Population: a Novel Approach

Efrat Manisterski and **Ron Katz** and **Sarit Kraus**
Computer Science Department, Bar-Ilan University, Ramat Gan, Israel
{maniste,sarit}@cs.biu.ac.il;katz@biu.013.net.il

Abstract

This paper presents a novel approach for providing automated trading agents to a population, focusing on bilateral negotiation with unenforceable agreements. A new type of agents, called semi-cooperative (SC) agents is proposed for this environment. When these agents negotiate with each other they reach a pareto-optimal solution that is mutually beneficial. Through extensive experiments we demonstrate the superiority of providing such agents for humans over supplying equilibrium agents or letting people design their own agents. These results are based on our observation that most people do not modify SC agents even though they are not in equilibrium. Our findings introduce a new factor —human response to provided agents — that should be taken into consideration when developing agents that are provided to a population.

1 Introduction

Designing automated trading agents has received increasing attention, as trading in many markets (e.g. NASDAQ) has now become fully electronic [Da-Jun and Liang-Xian, 2002]. In many domains, such as NASDAQ and e-bay, there is an entity which provides a generic agent for a considerable number of traders in the market. This entity might offer the agent for a fee or even for free, as a recommended agent. The goal of this entity is to provide an agent that maximizes the **overall** traders' utility when it is delivered to a population, for two reasons: First, one of the entity's goals is to provide good service to its users. Therefore, the entity should aim to maximize the average payoff of all the traders' agents in the environment. Second, web-marketplaces, such as e-bay, which host commercial transactions for a certain percentage of the transaction amount, are interested in increasing the total profits of all the traders, since they will receive higher fees as a result. Nevertheless, in these domains each client who obtains an agent has his own interests.

In this paper we discuss the question of which agent should be recommended to a population. We consider environments in which the fulfillment of the agreements reached by traders are unenforceable by the system, similar to many real-life domains. The game theory approach uses equilibrium agents [Da-Jun and Liang-Xian, 2002]. These agents' strategies are in equilibrium, namely no agent will be better off by deviating from its equilibrium strategy. However, although the equilibrium agent guarantees stability of the environment (there is no motivation for any trader to use any other agent), its outcome is not always Pareto-optimal, i.e. a solution may exist where some agents obtain **higher** utilities than they would have obtained by using the equilibrium strategy, while others do not attain lower utilities. Consider, for example, a seller and a buyer who have reached a sales agreement in a market-place which does not compel the traders to fulfill the agreement (e.g., Ebay). If the traders follow the equilibrium strategy, both sides will not fulfill the agreement they reached, (i.e. the seller will not send the goods and the buyer will not send the money). However, this solution is not Pareto-optimal, since it would be more beneficial, if the agreement would be fulfilled as agreed upon.

Different from the game theory approach, we present a new family of agents called semi-cooperative (SC). These SC agents are designed to reach mutually beneficial agreements and to keep their commitments (unless the other side has broken its agreements in prior transactions). In addition when trading with identical SC agents, they achieve a pareto-optimal outcome that is beneficial for both sides. Consider the trading environment in the previous example. If we would provide an SC agent and traders would adopt it without change, both sides would receive the money or the goods, as agreed. Consequently, traders would achieve better results than when using the equilibrium agent. However, in this example adopting an SC agent may not be beneficial, if many traders modify their SC agents to equilibrium agents that do not send money or goods. In such cases the traders that adopt an SC agent without changes will send money (or goods) without receiving goods (or money) in return.

Conversely, we claim that if a recommended agent is provided to a population, most people will not modify their agents, even if their strategies are not compatible with the equilibrium strategy. We restrict our argument to low-cost environments, so that if one of the traders breaks the agreement, the loss for the one that keeps it (e.g. sending the goods without receiving consideration) is relatively small. Such situations, where the traded costs of the goods are not very high (such as books, flowers and cinema tickets) are very common on the internet. There are several reasons for our claim that people do not tend to modify their agents. First, it is well known that people do not necessarily play according to the equilibrium [Grosz *et al.*, 2004]. Social factors such as equality, social welfare and risk taking play very important roles in people's decisions [Gal *et al.*, 2004]. Second, modifying an agent's code is time consuming and requires an understanding of the given code. For most users in low-cost environments it is simply not worth the effort, especially when the agent's code is complex and seems to be reasonable. Moreover, most traders are not programmers and modifying the code would require hiring a programmer thus incurring additional costs. Another factor is the credibility of an agent which is recommended by an authoritative and experienced entity. It is important to note that despite the above reasons, we believe that some people will change the recommended agent. However, we claim that the percentage of people who would change their agent is small, and thus it is still worthwhile to supply an SC agent which is more cooperative with others, rather than the non-cooperative equilibrium agent.

In order to examine our assumption we conducted extensive experiments which included 132 participants. In the first experiment, we provided three groups of human subjects with recommended agents for a resource allocation game. This game extends the well known Prisoner's Dilemma to simulate a common real-life bilateral negotiation environment. One group received an equilibrium agent, while the other two groups received SC agents; one with a simple negotiation method and the other with a more complex one. All subjects from the same group received the same initial code. Subjects were free to make any changes in their recommended agent. Note that the participants could change their agents not only to the equilibrium agent, but even to an agent which specifically exploits the recommended agent. The results show that most subjects did not change their agents. Moreover, in groups which were given an SC agent as a recommended agent, the overall utility of all the subjects was significantly higher than in the group that was given the equilibrium agent. Furthermore, even when giving subjects the opportunity to change their agents, after observing the results of playing with other agents in their group, subjects did not tend to change their agents and the superiority of the SC agents remained.

In addition, we compared the overall utility of recommending an SC agent to the overall utility when there is no recommended agent at all. For this purpose, we asked another group of subjects to design their own agents without proffering a recommended agent. We named these agents Peer Designed Agents (PDAs). The results show that the group that received a recommended SC agent yielded higher overall payoffs compared to the group which was not provided a recommended agent. Thus, we propose that entities wishing to maximize the overall utility in the environment offer the traders a recommended agent, rather than leave it totally open to the users' development.

The innovative contribution of this paper is in presenting a new approach for designing recommended agents that takes human behavior into consideration. To our knowledge we are the first to examine how human populations react to the existence of recommended computerized agents. Moreover, our approach saves the need of designing and implementing a mechanism for the trading environment. Mechanism design requires many resources, and it is not always applicable. Consider, for example, what it would take for E-bay to enforce all the agreements reached on the site. Our approach, on the other hand, induces cooperation and fairness without spending any additional resources.

The rest of the paper is organized as follows. In section 2 we discuss related work. In sections 3 and 4 we describe our negotiation environment and broadly present the semi cooperative agents (SC). Then we discuss the experiments we conducted in order to evaluate the SC agents' performance. Finally, in section 7 we conclude and suggest directions for future research.

2 Related work

Developing automated agents for a population has received extensive attention in recent years. In addition to equilibrium agents there are approaches which propose designing cooperative agent systems, e.g., [Zhang *et al.*, 2000]). In these systems the agents cooperate in order to achieve a common goal, or to maximize the overall system's utility. However, the agents in such systems are indifferent toward their own utilities. Therefore, these agent systems are inapplicable in our environment, where each agent attempts to increase its client's utility. Furthermore, there have been efforts to design a mechanism which induces cooperative behavior among selfish agents, e.g., [Mui *et al.*, 2002]. However, these mechanisms require resources and are based on assumptions such as repeated interactions and the ability to detect an agent's identity, which are not appropriate for our environment.

Some researchers have explored the behavior of evolutional agents that dynamically adapt to their environment, e.g., [Fatima *et al.*, 2005]. These agents are designated for environments in which learning and adaptation can be done during long periods. These agents are also used for modeling human behavior. However, the evolutionary model requires repeated interactions, while we consider negotiation environments where agents may interact only once. Other researchers have considered socially rational agents, whose utility depends on the overall utility of the society, e.g., [Hogg and Jennings, 2001]. In our environment, however, as in other e-commerce applications, traders do not take into consideration the overall benefit of the society. Other researchers have developed non-equilibrium agents for competing **against** a population of bounded rational agents such as humans, e.g., [Gal *et al.*, 2004]. In contrast, we have designed agents designated for the entire population. Moreover, none of the previous studies have considered human behavior when providing a recommended agent.

The concept which is most similar to the recommended agents considered here is a recommended play, in which subjects are presented recommendations of how to play a game, e.g., [Croson and Marks, 2001]). Some of the studies that explore recommended play, present mixed results concerning the influence of the recommended play on people's behavior. However, recommended play is fundamentally different from a recommended agent. First, most studies which examine recommended play induce playing the equilibrium strategy and are explored in games with simple strategies. In contrast, we suggest providing non-equilibrium agents and using more complex settings which reflect real-life negotiations. In addition, in order to use the recommended play people must actively choose it. However, in our case the recommended agent can be passively adopted without any additional activity required. Moreover, modifying the recommended agent does actually require effort and time.

3 Environment Description

In order to simulate a general trading framework, we have designed a resource allocation game. The advantage of this basic game is that it makes very few assumptions, and thus it can represent many situations in real economical markets. Moreover, such environments are very common and easy to manage via the Web. In the beginning of our 2-player game, each player i ($i \in \{1, 2\}$) is allocated an initial pool of resources R_i^{init}, which are attributed to several types. The goal of the game for each player i is to possess a specified set of resources G_i, which includes a certain quantity (zero or more) of each resource type. There are enough resources for both players to satisfy their goals, i.e. $G_1 \cup G_2 \subseteq R_1^{init} \cup R_2^{init}$. However, some of the resources needed by one player may be in the possession of the other. Thus, in order to obtain the goal set of resources, a player might need to exchange resources with the other party, according to a certain negotiation protocol. The protocol consists of a finite number of rounds n. Each round l is comprised of two phases: a negotiation phase and an exchange phase. In the negotiation phase one of the players makes an offer $O^l = (Ogive_1^l, Ogive_2^l)$ to exchange resources with the other player. In his offer the proposer i promises to send $Ogive_i^l$ resources to the other player and in return requests that player j send $Ogive_j^l$ to him. The receiver should inform the proposer whether he accepts or rejects the offer. Afterwards, there is an exchange phase, in which each player i sends a set of resources S_i^l to the other player. In this game, agreements are not enforced by the game controller, allowing players to break agreements. Therefore S_i^l can be different from $Ogive_i^l$. The exchange is executed simultaneously, so that no player knows how many resources, if any, the other party has actually sent till he has already sent his resources. The next round is identical to the first one except for the role of the proposer, which is alternated between the two players. The performance of each player is determined by its score at the end of the game. The score of player i ($score_i$) possessing the set of resources R_i^{end} at the end of the game is $Score_{goal}$ if the player holds the whole target set. In addition, each resource the player possesses imparts him with $Score_{res}$ points. Formally

$$score_i = \begin{cases} Score_{goal} + |R_i^{end}| Score_{Res} & G_i \subseteq R_i^{end} \\ |R_i^{end}| Score_{Res} & \text{otherwise} \end{cases}$$

The resource allocation game, supplies a general negotiation platform which is more similar to real-life negotiations than typical economic games.[1] This game is a simplified version of the Colored-Trails framework, which is a paradigm for examining negotiation environments which involve automated agents [Grosz *et al.*, 2004].

In our experiment the configuration of the parameters is: $n = 10$, $Score_{goal} = 200$ and $Score_{resources} = 10$. Thus, obtaining the target set becomes the most important component of the scoring function. We set the initial state in such a manner that one player possesses all the resources that comprise a target set, which means he is not dependent on the resources of the other. The other player, on the other hand, is dependent on the other player's resources. More specifically, he needs two resources from the other player in order to complete the target set. Both players have extra available resources for which they can negotiate. Our game setting enables an examination of situations in which the equilibrium strategy is not Pareto-optimal, and cooperation between the players yields better results for both parties: The dependent player can obtain the resources he needs to complete the target set, and the independent player can increase the number of resources he obtains by reaching agreements in which the number of resources he sends is lower than the number of resources he receives in return. Moreover, our game setting enables the examination of two different situations: the dependent player gains from the exchange of resources significantly more than he is required to pay, while the independent player only slightly improves his profit from the exchange. Both situations are common in real life. Consider for instance, a researcher who crucially needs to buy some books for an important research she is conducting. The seller will gain some money from the transaction. The researcher, however, is going to gain much more from the deal.

4 Recommending Semi-cooperative Agents

A recommended agent for this game can be described by its strategies, i.e. the agent's decisions in each possible phase of the game: the offers the agent makes, its responses to the offers it receives and the resources it sends. To describe the agents, when agent A plays the role of player 1 and agent B is player 2, we denote the set of resources that player i possesses at the end of the game and its score as $R_i^{end}(A, B)$ and $score_i(A, B)$, respectively. We use N_i^l to denote the set of resources that player i needs in round l in order to complete its goal set. Finally, me signifies the index of the player the agent plays and $other$ indicates the index of the other player.

In our resource allocation game the combination of unenforceable agreements and a finite horizon of the game leads to a theoretical equilibrium result that no resources will be exchanged. Therefore, an equilibrium agent (EA) in the context of this game is any agent that never sends any resources, i.e. $\forall i$, $\forall l$ $S_i^l = \emptyset$. As a result, at the end of the game each player will possess only its initial set of resources, $\forall i$, $R_i^{end}(EA, EA) = R_i^{init}$.

[1]The game is similar to the prisoner's dilemma by having the choice of cooperating (sending the promised resources) or not cooperating (not sending the promised resources). However, our game also includes negotiation processes to resemble real life situations.

In contrast to the equilibrium agent, we define a semi cooperative agent (SC) as an agent which

1. When the agent plays against an identical agent, both agents obtain higher scores than the scores they have at the beginning of the game.
2. When the agent negotiates with an identical agent, both agents reach a pareto-optimal solution.
3. It fulfills its agreements as long as the other side does not deceive it.
4. In each stage of the game, it commits to activities that are beneficial for both sides (for itself and for its opponent).

In particular in our game properties 1 and 2 imply that when an SC agent plays against an identical agent, both agents satisfy their goals, i.e. $\forall i,\ G_i \subseteq R_i^{end}(SC, SC)$. According to property 4, for each round l the SC agent proposes and accepts agreements that are beneficial to both sides. In this game, an agreement is beneficial for player i if it increases the number of resources it possesses, or if it increases the number of resources it needs in order to satisfy its goal, i.e. $|Ogive^l_{other}| > |Ogive^l_{me}|$ or $Ogive^l_{other} \cap N_{me} \neq \emptyset$. Note that unlike a cooperative agent which agrees to send resources to the other agent for free if it maximizes the joint utility, the SC agent does not accept such an offer. When providing an SC agent as a recommended agent, if players adopt it without change, the dependent players will satisfy their goals, and the independent players will obtain additional resources. Therefore both players will benefit more by using semi-cooperative agents than when both of them use the equilibrium agents (in case we provide equilibrium agents). Consequently, providing semi-cooperative agents is superior over providing equilibrium agents if the users do not modify their agents, $\forall i\ score_i(SC, SC) > score_i(EA, EA)$. However, semi-cooperative agents can obtain a lower score than equilibrium agents, if one of the users modifies his recommended agent. For example, consider a scenario in which user US_1 adopts the SC recommended agent without change while user US_2 modifies its agent to an equilibrium agent that does not send any resources. In this scenario, the agent of US_1 will send resources to the agent of US_2 but will not receive resources in return. As a result, US_1's score in this game will be lower than if US_1 would have used an equilibrium agent, which would have let him at least reserve all his initial resources. Nevertheless, if most users do not modify their agents, the revenue of an SC agent in games where both sides use SC agents will compensate for the loss from games where the SC agent plays against a modified agent.

4.1 Experiment

In order to investigate which agent should be provided to a population, an SC or an equilibrium agent, we conducted the following experiment. In this experiment we examined the behavior of three groups of programmers who received recommended agents' codes for the resources allocation game. For this purpose, in addition to the equilibrium agent **(EA)** we designed two SC agents[2], distinguished by their level of complexity, as described below.

Description of SC Agents

The Simple semi-Cooperative Agent (SCA)

In general, this agent is a simple agent in the sense that it agrees to any beneficial offers (it doesn't try to increase its score by being greedier at the beginning of the game) or try to learn the opponent type (see SSC's strategy for detecting liars). More specifically:

Sending strategy: This agent is honest and fulfills its agreements, unless the other agent deceives it. Once this player has been deceived, it stops sending resources.

Negotiation strategy: This agent's logic is quite simple. For round l of the game; if the agent has already satisfied its goal ($N^l_{me} = \emptyset$) it accepts any offer that: (1) Gives the other agent resources it needs to satisfy its goal, $Ogive^l_{me} \subseteq N^l_{other}$ (2) Increases the number of resources it possesses ($|Ogive^l_{other}| > |Ogive^l_{me}|$). If the agent hasn't satisfied its goal ($N^l_{me} \neq \emptyset$), it accepts any offer which: (1) Promotes it to satisfy its goal, $Ogive^l_{other} \subseteq N^l_{me}$, $Ogive^l_{other} \neq \emptyset$ (2) It needs to send only the resources not necessary for it to reach its goal, $Ogive^l_{me} \subseteq Av^l_{me}$, where Av^l_i denotes the agent's set of available resources.

The Complex semi-Cooperative Agent (CCA)

In general, this agent has a complex logic as it tries to increase its score by being greedier at the beginning of the game. In addition, it tries to minimize the damage in case it plays against a deceiving player. More specifically:

Sending strategy: CCA is honest and usually fulfills its agreements. However, unlike the SCA, this agent is also cautious and protects itself from agents that do not fulfill their agreements. CCA identifies situations in which there is a high probability that it will be deceived by the other agent which will not send the resources agreed upon:(1) In situations in which the other agent has already satisfied its goal and if it will keep the agreement, the CCA agent will satisfy its goal, i.e., $N_{me} \subseteq Ogive^l_{other}$. In these situations there is a higher probability that the other agent will not send the promised resources, since both agents are about to possess their target sets and no future interactions are expected. In such cases, there is no motivation for the other agent to maintain its credibility and to avoid deceiving. (2) When the other agent has already deceived in the past. (3) When there are very few rounds till the end of the game. In all the situations mentioned above, the CCA waits to send its resources until one round after receiving the resources from the other agent, i.e, $S^{l+1}_{me} = Ogive^l_{me}$ if $Ogive^l_{other} = S^l_{other}$.

Negotiation strategy: In round l of the game, if the agent has already satisfied its goal ($N^l_{me} = \emptyset$) it accepts offers in which it promises to send the other agent only **one** resource which the other agent needs in order to satisfy its goal, i.e $|Ogive^l_{me}| = 1$ and $Ogive^l_{me} \subseteq N^l_{other}$. The logic behind sending one resource at a time is to minimize the damage in case the other side is a liar and does not fulfill its agreement. The number of resources it demands for sending this resource decreases over time. Similarly, if the agent has not

[2]The behavior of the recommended SC agents can be any behavior that satisfies conditions 1-4 in the SC definition. Indeed in our experiment we used two variations of SC agents: a simple agent that negotiates several resources at a time and a complex agent that negotiates one resource at a time.

satisfied its goal ($N^l_{me} \neq \emptyset$), it accepts any offer in which it is promised one resource which it needs to satisfy its goal, i.e. $|Ogive^l_{other}| = 1$,($Ogive^l_{other} \subseteq N_{me}$). The number of resources it agrees to send in return increases over time.

Experiment Design

In our experiment 99 subjects were divided to three equal groups. Each group was provided with a different recommended agent's code[3] EA, CCA or SCA. All programmers of the same group received the same initial code. We also presented a short description of the recommended agent's strategy. The group that received the EA agent was also explained the meaning of their agents being equilibrium agents. We conveyed to the subjects in all the groups that the provided agent is recommended. Each programmer was requested to submit an agent's code aiming to maximize its score (as specified below), which was not necessarily equal to or based upon the recommended agent. They were required to submit their final agents in 10 days. We explained that each agent would play in a tournament against all the other 32 subjects' agents who had received the same recommended agent. Each agent played both as the role of the dependent and the role of the independent agent against all the other agents. It is worthy to note that this experimental methodology is innovative since no one, as far as we know, has ever examined the influence of providing a recommended agent to a population of programmers.

All subjects who participated in our experiment were upper-class undergraduate and graduate computer science students at Bar-Ilan University, who are experienced programmers. The subjects were motivated to perform well by receiving a course grade. A subject's grade was calculated according to the score obtained by his agent accumulated over all its 64 plays. We emphasized that the grade is based only upon the subject's own score, and not upon other agents' scores - similar to real negotiation environments, where traders gain money only according to their own agents' performance. The game was explained in class and the participants were able to ask questions. In addition, they received an accurate written description of the game. For debugging purposes, we gave participants a server upon which they could run their agents. We made sure that each subject knew how to run the server and we offered any technical support needed.

Our main hypothesis was that the groups provided with SC agents would achieve better results (as specified below) than a group provided with the equilibrium agents. The results were evaluated based on two aspects: (1) The average score of the players that adopted the recommended agent without changing it. (2) The overall score of all the agents in the environment, including agents that were modified to deviate from the recommended strategy. This hypothesis was based on our assumption that people tend not to change their agents, as explained in the introduction.

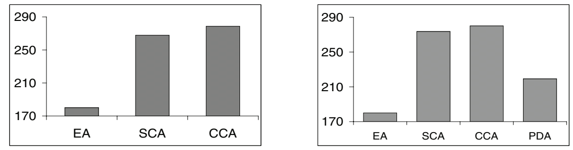

Figure 1: (a) Recommended agent's average score; (b) Average score of all the agents in the environment.

4.2 Results

The average scores achieved by the agents in the tournaments are presented in Figure 1. The results depicted in Figure 1.a show that the average scores achieved both by SCA and CCA agents (which were not modified) are significantly higher than the EA's average score (Mann-Whitney test, $p < 0.01$). Similar results are presented in figure 1.b, which depicts the average scores of all agents in the environments (including agents modified to deviate from the recommended agent strategy). The superiority of using the SC agents over using the EA agents applies to both the dependent and the independent roles.

These results support our hypothesis which claims that providing SC recommended agents to a population yields better results than providing the equilibrium agent. We have based our hypothesis mainly upon the assumption that people tend not to change the recommended agents. This assumption was supported by the experimental results. In all three groups, most subjects did not modify their agents. In particular, only 5 subjects (of 33) modified the recommended agent in the EA group, 10 in the SCA group and 4 in the CCA group.

The provided agents' performance is determined not only by the number of people that change their agents, but also by the way they modify their agents. Some changes do not affect the recommended agents' score and therefore are not relevant. Other changes do reduce the recommended agents' score. Therefore we classify the modified agents according to the way they affect the recommended agent's score, when they play against it. More specifically the modified agents can be classified into three types. These types are determined by comparing the performance of the modified agent when playing against a recommended agent to the recommended agent's performance when playing against another recommended agent: (1) **Counter agents** - agents that achieve higher scores than the recommended agent. The **best-response** agent is the agent that achieves the highest score among all the counter agents. (2) **Agents with neutral changes**- agents that achieve the same score as the recommended agent. (3) **Inferior agents**-agents whose scores are lower than the recommended agents. In the EA group, all the modified agents were agents of type 2, and no resources were sent during the EA group tournament. Unlike the EA group, in the SCA and the CCA groups most of the modified agents were counter agents (8 out of 10 and 3 out of 4, respectively). Surprisingly, only one subject modified its agent to the best-response agent.

Comparing the performance of SCA and CCA groups

[3]The code provided was written in C++ or in Java according to the subject's preference.

reveals higher average scores of the latter, though non-significant (see Figures 1.a and 1.b). Similarly, as mentioned above, there were more modified agents in the SCA group. This can be attributed to the higher sophistication of the CCA's negotiation method. Indeed, no changes were made in CCA's negotiation method, and the only changes were in the send method. The SCA's negotiation method, on the other hand, was naive, i.e. it accepted any offer that increased its score, and therefore called for more modifications. Another observation, which indicates that the recommended agent's strategy affects the way people modified their agents, is that no subject modified its agent to become an inferior agent (type 3).

5 Users' Response to their Agents' Performance

In the previous section we showed that most subjects did not modify the provided SC agent. However, consider a scenario in which people are able to observe their agents performance after playing several games. Even though most people would not have modified their agents before observing their agents' results they may modify it after such observation. In order to investigate whether people modify their agents after observing their results, we provided the subjects with feedback on their agents' performance, after they submitted their agents (as described in the previous section). Each subject received a file which detailed the final score of each of his agent's games and the other agent's scores. In order to keep the other player anonymous we identified each player by a random number. In addition, the subject received a log file for each game its agent played, which included the course of the game.[4] After observing these log files, subjects had one week to revise their agents and resubmit them.

We hypothesized that the number of subjects that would modify their agents at this stage would still be small enough, such that it would still be profitable to provide the SC agent. There are several reasons for our hypothesis. First, as shown in Figure 1, the SC agents achieved high scores in the tournament. Thus, we assumed that most players would be satisfied with their score and would not modify their agents. Second, the tasks of reading all log files and understanding the course of the games is very exhausting and time consuming. Third, even if subjects recognize that other players modified the original agent, it is not trivial to derive which are the modified agents codes. Fourth, even if we assume that someone succeeded in recognizing the hidden codes of all the agents in the environment, sometimes it is very difficult to calculate the strategy that will yield better results against all other modified agents. Thus we believed that most people would not modify their agents after observing their performance. The experiment results support our hypothesis. Only one subject modified his agent.

In order to examine how seriously the participants treated the experiment (including the first part), we asked them to fill out an anonymous questionnaire after the experiment. It was revealed that the exercise was considered very seriously, and 96% of the respondents read their agents code very carefully. We believe that this percentage is even higher than the percentage of traders who would read a recommended agent's code in the real world, for several reasons. First, not all traders are programmers. Therefore, in order to read the recommended agent's code or modify it they must make an effort and consult a programmer. Second, in our experiment we explicitly mentioned the possibility of changing the recommended agent's code. In real world situations, however, we believe that many traders would not be aware of this possibility.

[4]The subjects could not observe the courses of other players' games, nor other subject's total score, similarly to many real negotiation environments in which traders are enable to see only their own trading results.

6 Comparison: SC and Peer Designed Agents

In the previous sections we discussed the question of which kind of agent a central entity should provide to a population, given that it should provide a certain recommended agent. However, frequently the central entity has the choice of deciding whether to provide a recommended agent or to let traders design their own agents by themselves. It is not trivial that the first option is preferable, for a few reasons. First, it is well documented that people do not play and do not design agents according to the equilibrium strategy [Grosz *et al.*, 2004]. Second, when each trader designs his own agent, no trader knows the other agents' strategies. This is in contrast to the recommended agent approach, in which traders can exploit their knowledge about the strategy of others and modify their agents to achieve higher scores at the expense of others.

In order to compare the recommended agents to Peer Designed Agents (PDAs), we asked another group of 33 subjects to design agents that play the resource allocation game, without providing any recommended agent code. In addition, the subjects were asked to present verbal documentation explaining their strategy. In order to help subjects to focus on strategy matters and not on coding, we provided them with an agent skeleton which included three functions: Accept, Send and Propose. The subjects were instructed to implement these functions.

Following the high performance of SC agents (described in Figure 1) we hypothesized that the group that received a recommended SC agent would yield a higher overall utility compared to a group that had not received a recommended agent. We based our hypothesis on the assumption that providing a default SC recommended agent would cause some people to adopt it and to play cooperatively, though they would not have designed such a cooperative agent by themselves. As expected, the average score of all the games played by the PDAs (shown in Figure 1.b) is significantly higher (Mann-Whitney test, $p < 0.01$) than in the EA tournament. However, this score is significantly lower (Mann-Whitney test, $p < 0.05$) than the average score of agents in both SCA and CCA groups. In particular, three PDAs were equilibrium agents, and ten others never sent their opponents all the resources required for satisfying their goals. The remaining 20 agents enabled their opponents to satisfy their goals in some of the games. This explains the higher performance of the PDAs compared to the EA group. However, there were games, where the 20 agents played against each other without satisfying the goal of both players. This is because the PDAs were heterogeneous and two PDAs could have different exchange rates. Thus, in some of games they could not reach an agreement during negotiations. In contrast, since most subjects adopted the recommended agents in the CCA and SCA tournaments both players usually satisfied their goals.

7 Conclusions And Future Work

In this work we present a novel approach for providing automated trading agents to a population. We focus on bilateral negotiations with unenforceable agreements. Unlike the traditional Game Theory approach that recommends the provision of equilibrium agents which are not always pareto-optimal, we propose to provide a new type of agents called semi-cooperative agents that are not in equilibrium. When these agents negotiate with each other they reach a pareto-optimal solution that is mutually beneficial. The theoretical risk in providing such agents is that people can change their agents in order to increase their own utility at the expense of others. However, we hypothesized that if people are provided with agents that are reasonable and beneficial for both sides, most people will not spend time and effort to change it. Therefore, we designed semi-cooperative agents which yield higher benefits for both sides using it, compared to the case of both sides using equilibrium agents. In order to test our approach we designed an unenforceable resource allocation game, which simulates real life bilateral negotiations. Indeed, experimental results of this resource allocation game fully supported our hypothesis. The experiments show that providing SC agents to a human population yields better results than providing equilibrium agents or letting people design their own agents. Our results, based on 99 programmers who mostly did not change their agents, has led us to assert that common clients, who are typically non programmers, will not invest the efforts needed to change the code in complex real world situations. In light of these findings we propose that users' reactions to recommended agents should be taken into consideration when developing agents that are provided to a population.

Our research focuses on negotiation environments, which are very common in commercial life. However, the SC principle may also be successfully implemented in other environments, such as: information exchange, where the information transferred is not necessarily reliable, or tasks distribution, where each agent decides how much effort to invest in carrying out its part in the assigned task. In the future we intend to examine our approach in such environments. In addition we plan to develop a formal model that will support our approach and explore our agents in real trading problems.

8 Acknowledgment

This work was supported in part by NSF #IIS0222914 and ISF #1211-04. Kraus is also affiliated with UMIACS.

References

[Croson and Marks, 2001] R. Croson and M. Marks. The effect of recommended contributions in the voluntary provision of public goods. *Economic Inquiry*, 39(2):238—249, 2001.

[Da-Jun and Liang-Xian, 2002] C. Da-Jun and X. Liang-Xian. A negotiation model of incomplete information under time constraints. In *AAMAS02*, pages 128–134, 2002.

[Fatima *et al.*, 2005] S. Fatima, M. Wooldridge, and N. R. Jennings. A comparative study of game theoretic and evolutionary models for software agents. *AI Review*, 23(2):187–205, 2005.

[Gal *et al.*, 2004] Y. Gal, A. Pfeffer, F. Marzo, and B.J. Grosz. Learning social preferences in games. In *AAAI04*, pages 226–331, 2004.

[Grosz *et al.*, 2004] B. Grosz, S. Kraus, S. Talman, S. Boaz, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. In *AAMAS04*, pages 782–789, 2004.

[Hogg and Jennings, 2001] L. M. J. Hogg and N. R. Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 31(5):381–393, 2001.

[Mui *et al.*, 2002] L. Mui, A. Halberstadt, and M. Mohtashemi. Notions of reputation in multi-agents systems:a review. In *AAMAS02*, pages 280—287, 2002.

[Zhang *et al.*, 2000] X. Zhang, R. Podorozhny, and V. Lesser. Cooperative, multistep negotiation over a multi-dimensional utility function multi-agent systems negotiation. In *ASC00*, pages 136—142, 2000.