Design of Social Agents

Roman Gorbunov, Emilia Barakova, and Matthias Rauterberg

Eindhoven University of Technology, Den Dolech 2, Postbus 513, 5600 MB Eindhoven, The Netherlands

Abstract. Social behavior, as compared to the egoistic and rational behavior, is known to be more beneficial to groups of subjects and even to individual members of a group. For this reason, social norms naturally emerge as a product of evolution in human and animal populations. The benefit of the social behavior makes it also an interesting subject in the field of artificial agents. Social interactions implemented in computer agents can improve their personal and group performance. In this study we formulate design principles of social agents and use them to create social computer agents. To construct social agents we take two approaches. First, we construct social computer agents based on our understanding of social norms. Second, we use an evolutionary approach to create social agents. The social agents are shown to outperform agents that do not utilize social behavior.

1 Introduction

Collaboration, trust and other social norms are crucial for good operation of teams. Our long term goal is to create a model of social norms that will enable prediction of behavior and interactions within a team. On the basis of such a model we want to test hypotheses of how these interactions within a group will develop.

Human and animal models of behavior inspired studies on social learning [4,19,18,17,21] and models of social norms such as trust and collaboration [19,18,17,14]. Collaboration has also been studied within economics, sociology, and game theory. In economics, models of a rational actor who collaborates to improve organizational effectiveness have been developed [15,11,9]. In sociology [5], the non-rational and social aspects of collaboration has been explored. Gorbunov at al. [8] have proposed a model that accounts for the deviations from the rational behavior.

Games, in particular are an establishing field that makes possible realistic social interactions to be exposed, monitored and trained [2,1,3,12,13,16]. Hennes [12] Voynarovskaya et al.[20] and Gorbunov at al. [8] have developed a game-based method for monitoring the social relations in small groups of individuals. It is based on the records of a negotiation game that monitors of the collaborative behavior of the players. The Colored Trails game framework [7] in a three-player negotiation variation [6,20] was used for this study. The conclusions of the study have been restricted by the limited amount of data and the

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part I, LNCS 6686, pp. 192–201, 2011.

difficulties to constrain the experiment by having a control over one or several individuals.

We propose to use a multi-agent system approach for the purpose of creating an agent-based support tool for team collaboration as a first step towards prediction of conflicts in small teams of people. We design the agent-players that in controlled manner can express collaborative or egoistic behavior. We hypothesize that the agent with persistent collaborative behavior will be more successful in collaboration game. We also expect that the outcome of the experiments will give us better insights of how to improve the agents so they can be used as a controlled substitute of a human player.

Moreover, we want to understand the mechanism of collaborative behavior, i.e. how collaboration is maintained, how to avoid exploitation of collaborative agents. This understanding will help us interpret the data from MARS 500 experiment [12,20,10].

2 Methods

2.1 Settings of the Interactions

In our study we use the prisoner dilemma, a classical problem in the game theory, which used to study collaborative behavior. Every player can chose either cooperation or defection. If both players choose to cooperate, than they get 3 points each. In contrast, every player gets only 1 point if both of them choose to defect. In other words, the mutual cooperation is more beneficial than the mutual defection. On the other hand, the defection is always more beneficial than then cooperation under any given strategy of the partner. If one player defects and another one cooperates than the defecting and cooperating player get 5 and 0 points, respectively. Because of the above mentioned properties of the game, players face a dilemma: to cooperate or to defect.

In our study the interaction between two players (computer agents) is organized in the following way. One of the agents starts the game by choosing one of the two available moves (cooperation or defection). Another agent "sees" the move chosen by the first agent and uses this information to make its own move. After two agents made their moves, each of them gets points determined by the above given payoff matrix. The third move is made by the first agent. The third and the second moves together determine another portion of points that will be delivered to the two interacting agents. In other words, the second move determines the payoffs two times (first, in combination with the first move and then in combination with the third move). The agents make subsequent moves in the above described way until a predefined number of moves is made.

2.2 Model of the Agents

To formalize algorithms imbedded into the computer agents we consider a model similar to the Turing machine. Every agent has a fixed and limited number of

internal states. The previous move of the partner is considered as the input for the algorithm. The input and the internal state of the agent determine its output (the next move) that can be either cooperation or defection. The mapping of the inputs and internal states into the output is given by a fixed matrix. Additionally to the generation of the output the agent changes its internal states. The new internal state, like the output, is determined by the input and previous internal state. The mapping of the input and the current internal states into the new internal states is also given by a fixed matrix.

2.3 Design Principles

The maximization of the personal benefit of an agent in a sequence of interactions with a random partner is the main design criterion that we used for the construction of the social computer agents. In other words, our intention is to create an egoistic agent whose the only goal is to maximize its own benefit in the environment populated by other agents whose playing strategies are not known.

In the prisoners dilemma the defection is always the most beneficial choice independently on what choice, defection or cooperation, is made by the partner. As a consequence, the agents, that are rational in the classical game theory sense, always choose to defect. In other words, the rational agents demonstrate mutual defection that is known as Nash equilibrium for the prisoners dilemma. Further on we will call the agents of this kind as "defective" agent.

The permanent defection is one of the two simplest possible strategies and the permanent cooperation is the second one. We will call the agents that permanently cooperate as "cooperative" agents. This strategy can also be considered as a rational one since the mutual cooperation is more beneficial than the mutual defection.

However, it has to be emphasized that the cooperation is beneficial only if it is reciprocal. The cooperation with an agent that defects can be considered as an exploitation of the cooperative agent by the defective one. This kind of interaction is very beneficial to the exploiting agent and very unbeneficial to the exploited one. Thus the cooperative agent cannot be considered as a successful solution for the environment populated by agents of different kinds.

To prevent an exploitation by a partner the agent should exhibit a reciprocal behavior (defect with the defective agents and cooperate with the cooperative agent). We will call the agents of this type as "reciprocal" agents. The reciprocal agents can be considered as more successful than the above introduced defective and cooperative agents. In contrast to the cooperative agents, the reciprocal agents cannot be exploited by the defective agents since they will behave defectively with the defective agents. In contrast to the defective agents, the reciprocal agents will establish and maintain cooperation with each other if they are constructed to start from the cooperative move. The weak point of the reciprocal agents is that they are not able to exploit cooperative agents. In other words, the reciprocal agent cooperates with its partner even if it is not a necessary condition for the cooperation of the partner. This kind of behavior is in a disagreement

with the above introduced goal to find a behavior that maximizes the personal benefit of the agent.

The above described requirements can be summarized in the following list of the social skills that have to be embedded into the social agents:

- To avoid the exploitation by the partner, the social agent should be able to reciprocate defection of the partner (react defectively on the defection).
- To maintain the cooperation, the social agents should be able to reciprocate the cooperation of the partner (react cooperatively on the cooperation).
- To establish the cooperation, the social agents should be able to propose the cooperation in the phase of the mutual defection (react cooperatively on the defection).
- To exploit the partner, the social agents should be able to try defection in the phase of the mutual cooperation (react defectively on the cooperation).

As we can see, the above described requirements can be competing. So, the agent needs to be able to decide what logic to apply at the current stage of the interaction. Moreover, the agent needs to remember the responses of the partner. In more details, it should not keep trying to establish a cooperation if all these attempts fail. Similarly, the agent should not continuously try to exploit its partner if it sees that these attempts are not successful. On the other hand, the agent should not cooperate if it is not required for the cooperation of the partner.

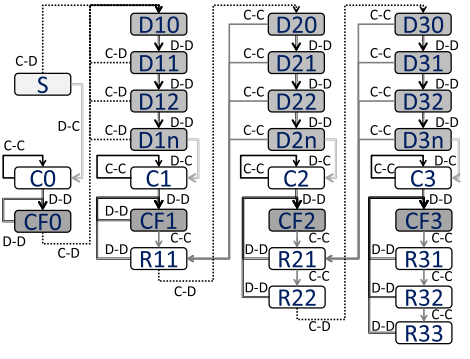

Fig. 1. Schema of the constructed social agent

We implement the above described design principles by the following algorithm. The agents start cooperation by proposing cooperative move as a response on the defective move of the partner. Doing that agent switches to the state of cooperation (CN in the Fig.1). If partner answers defectively on cooperation initiated by the agent, the agents switches to the cooperation-failed (CF_n) states. In this state the agents never initiates a new cooperation. If partner starts cooperation, the agent will defect, immediately or after several steps of cooperation, to check if its partner can be exploited. In more details, for several steps agent

plays only defectively. If at this stage the partner made at least one cooperative move, the agents concludes that exploitation was successful and try to exploit the partner again. If the agent has already tried to exploit after n steps of cooperation and this attempt failed, the next attempt to exploit will start after $n+1$ steps of cooperation. If the agent tried to exploit after N_{max} steps of mutual cooperation and this attempt failed, it never tries to exploit again. After every unsuccessful attempt of exploitation agents tries to establish cooperation. More details about the algorithm are given in the Tab.1. In this table different columns and rows represent different inputs (moves of the partner) and internal states of the agents, respectively. The cells of the tables contain outputs (moves of the agents) and a new state of the agent. The defective and cooperative moves are denoted as $\ominus$ and $\oplus$, respectively.

Table 1. Algorithm of the Constructed Social Agent

Internal State	Cooperative input	Defective input	No input
S	$\ominus$, D10	$\oplus$, C0	$\oplus$, C0
D10	$\ominus$, D10	$\ominus$, D11	
C0	$\oplus$, C0	$\ominus$, CF0	
D11	$\ominus$, D10	$\oplus$, C1	
CF0	$\ominus$, D10	$\ominus$, CF0	
C1	$\oplus$, C1	$\ominus$, CF1	
CF1	$\oplus$, R11	$\ominus$, CF1	
R11	$\ominus$, D20	$\ominus$, CF1	
D20	$\oplus$, R11	$\ominus$, D21	
D21	$\ominus$, D20	$\oplus$, C2	
C2	$\oplus$, C2	$\ominus$, CF2	
CF2	$\oplus$, CF2	$\ominus$, CF2	

2.4 Evolution of Agents

In our study we apply an evolutionary optimization to find a well performing agent that can be compared with the constructed one. The evolution starts from a random agent containing the same number of states (12) as the constructed one. At every step of the evolution we perform a random mutation of the agent by changing one element in one of the two matrices that define the agent behavior. If the mutated agent performs better as the parent agent, we replace the parent agent by its child and continue the procedure.

3 Results

An interaction between any two agents can be described by the ordered sequence of moved made by the two agents. If the interaction between the two fixed agents is long enough, the sequence of moves converge to a periodic pattern (because of the limited number of the internal states of the agents). As a consequence the performance of the agents, which is defined as the average score per game,

depends on the length of the interaction. In particular it is meter if the convergence is reached and, if is the case, at which phased of the periodic patter the interaction is stopped. To get a representative performance of an agent we have to make sure that the interaction between the agents is long enough to suppress the dependency of the performance on the length of the interaction.

The performance of two agents can also depend on which agent started the interaction first. To deal with this ambiguity, for any pair of agents we run two sequences of interaction such that every agent starts one sequence.

For 29 agents we run the interactions containing 1000 steps. We found out that for all considered agents the difference between the performances on the step 999 and 1000 do not exceed $1.58\ 10^{-3}$. This number is small in comparison with the variation of the performances of different agents, indicating that 1000 steps are sufficient to reach the convergence. Further on we will use this number of games to calculate performance of agents, unless otherwise stated.

The performance of an agent also depends on the partner. Moreover, even if agent A outperforms agent B while both playing with partner C it is still possibe and agent A underperforms B while both playing with another partner D. To get a representative performance of an agent we create a larger pool of random partners and let the agent to play with every partner from the pool. Then the performances of the agent with different partners are averaged to get the representative performance. In our study we populated the pool by agents that have no more than 12 internal states. The probability of an agent to be in the pool does not depend on the number of internal states.

We chose the size of the pool of partners in the following way. For a given random agent we have generated 10 different pools of the same size. Every pool has been used to calculate the performance of the given agent and then the difference between the maximal and minimal performance was calculated. If this difference was larger than 0.1 we double the size of the pool and repeat the same calculations. We start the procedure from the minimal possible pools containing only one partner and grow the pool in the above described way until the difference between the minimal and maximal performances of the considered agent is smaller than 0.1. The above procedure has been performed for 7 different random agents. We found out that for the all considered agents the range of the performances became smaller than 0.1 with the pool containing less than 10 thousands agents. Further on pools containing 10 thousands agents are used, unless otherwise stated.

We used the above given values for the length of the interaction and the size of the pool to estimate the performance of the constructed social agent. We found out that the score per game for this agent is equal to approximately 3.16. To get an idea of how good this performance is we have calculated performances of 4855 randomly generated agents. The distribution of the performances of these agents is shown in the Fig.2. As we can see in the figure, the constructed social agent outperform majority of randomly generated agents. In more details, approximately 97.8 % of random agents were outperformed by the constructed agent.

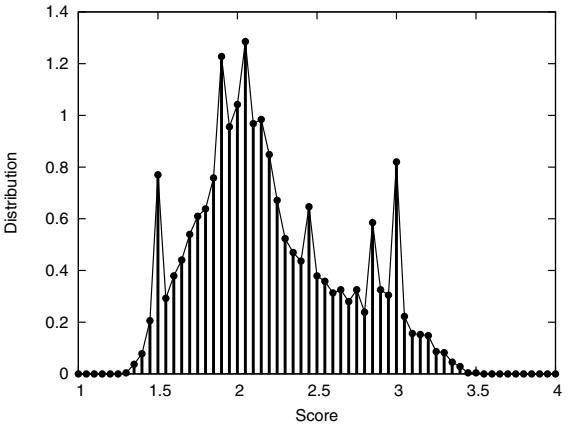

Fig. 2. Distribution of the performances of the random agents

However, we have to note that the performances calculated in this way should depend on the way the pool is populated. In our study we used a procedure in which any two agents, that have the same number of the internal states, have the same chance to be put into the pool. Moreover, the probability of an agent to be added to the pool does not depend on the number of the internal states. In other words we got a "uniform" distribution of the agents in the pool and, as a consequence, the performance of agents, calculated in this way, can be considered as more "representative" or "universal".

However, we need to remember that the assumption of the homogeneous distribution of the agents in the pool is not always valid. In particular, the pool can be populated by artificial agents that were constructed with the intention to increase the personal performance. Or, as another example, the content of the pool can be a product of an evolution. In both cases the chances of an agent to be in the pool are higher if its performance is better.

To overcome the problem of the dependency of the score on the way the pool of partners was generated, we propose to compare performances of two agents by letting them to interact with each other. In more details, we want to model performances of two different agents in the pool that is equally populated by the agents of the two considered types. To model this situation we do not need to use a larger pool because the interactions of a given agent with the partners of the same type will always give the same score per game. The agent of a chosen type needs to interact with one partner of the same type and one partner of another type. Interactions with partner of two different types of agents gives two values of the performance that are then averaged to get the performance of the given agent in the considered situation. The performances of the two agents, calculated in this way, can be interpreted as their abilities to survive, in the evolutional sense, in the pool that is equally populated by the agents of the two considered types. We would like to emphasize that the performances of agents, calculated in the above described way, are not transitive. If agent A outperforms agent B and agent B, in its turn, outperforms agent C, it does not necessarily means that agent A outperforms

agent C. As the consequence, the second way to compare performances of agents cannot be used in the evolutionary optimization.

We used the above described procedure to estimate performances of the constructed and evolved agents. The constructed social agent has been compared with 10 thousands randomly generated agents. In more details, for every pair of the random and constructed agent we calculated the performances of the both agents while interacting with each other. The obtained performances are shown in the Fig.3. The x-axis and y-axis correspond to the scores of the constructed and random agents, respectively. As we can see in the figure the performance of the constructed agent was always better than those of the confronted random partner. This represents the fact that the constructed agents cannot be exploited by their partners.

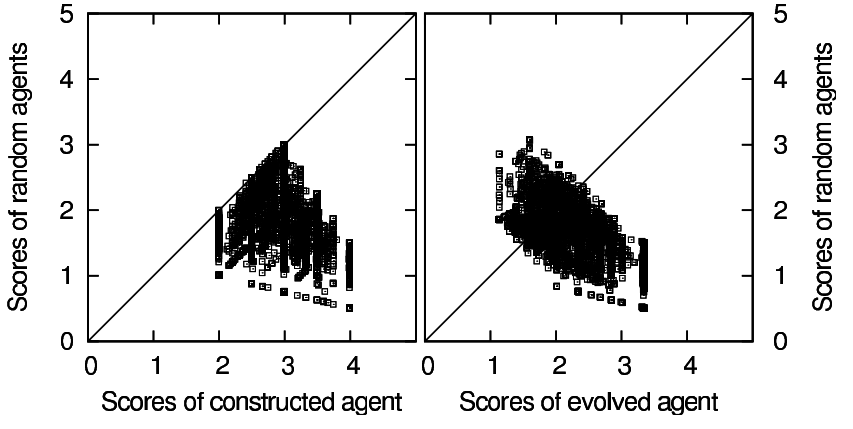

Fig. 3. Performances of the constructed and evolved agents compared with the perfomrances of the random agents

In the same way we have estimated the performance of the best agents found by the evolutionary optimization (see Fig.3). As we can see in the figure, the random agents outperformed the evolved one in many cases. In other words the evolved agent does not have mechanisms that protect it from the exploitation. As we have shown before, the evolved agent significantly outperformed the constructed one in the pool of the random partners (with the average score per game equal to about 3.6 and 3.2 for the evolved and the constructed agents, respectively). However, in spite of that fact, we found out that in the pair interactions with the random agents the average performance of the evolved agent was smaller than those of the constructed agent (about 2.6 and 3.1, respectively). It can be explained by the fact that the constructed agent interact with each other better than the evolved agents. The average performance of the constructed agents in the pool of the constructed partner is about 3.0 while the evolved agents in the pool of the evolved partner have performance equal to about 1.7.

4 Discussion

In our study we provided computer agents that are able to increase personal performance by utilizing social interactions with random agents with unknown strategies. In more details, we provided a mechanism that allows agents to establish and maintain cooperation. The presented agents establish and maintain cooperation only if it is necessary and sufficient condition for the cooperative behavior of the partner. Moreover, the constructed agents are able to prevent exploitation by the defective partner. And finally, the presented agents can exploit those partners that allow the exploitation. We have demonstrated that the constructed agents perform outperform majority of random agents and even the agents that have been obtained by evolutionary optimization. In this way we proposed the computational model that describes cooperation and exploitation.

The proposed model provides a mechanism of how collaboration and exploitation can be established and maintained. As a consequence we contribute to a better understanding of these social phenomena. The presented model can be used to identify collaborative and exploiting patterns in humans behavior and, in this way, explain human behavior better. Moreover, the constructed agents mimic human social behavior and, as a consequence, can better play against human players in collaborative computer games.

Acknowledgments

The research reported in this paper is supported by NWO User Support Program Space Research. The project number is ALW-GO-MG/07-13.

References

1. Barakova, E.I., Gillessen, J., Feijs, L.: Social training of autistic children with interactive intelligent agents. Journal of Integrative Neuroscience 8 (1), 23–34 (2009)
2. Barakova, E.I., Lourens, T.: Expressing and interpreting emotional movements in social games with robots. Personal and Ubiquitous Computing 14, 457–467 (2010)
3. Bekker, T., Sturm, J., Barakova, E.: Design for social interaction through physical play in diverse contexts of use. Personal and Ubiquitous Computing 14 (5), 381–383 (2010)
4. Bonnie, K.E., Horner, V., Whiten, A., de Waal, F.B.M.: Spread of arbitrary conventions among chimpanzees: a controlled experiment. Proceedings in Biological Science 274(1608), 367–372 (2007)
5. DiMaggio, P.J., Powell, W.W.: The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. American Sociological Review 48, 147–160 (1983)
6. Ficici, S.G., Pfeffer, A.: Modeling how humans reason about others with partial information. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 315–322 (2008)
7. Gal, Y., Grosz, B.J., Kraus, S., Pfeffer, A., Shieber, S.: Colored trails: a formalism for investigating decision-making in strategic environments. In: Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games, pp. 25–30 (2005)

8. Gorbunov, R., Barakova, E., Ahn, R., Rauterberg, G.W.M.: Monitoring interpersonal relations through collaborative computer games (2011) (submitted)
9. Grant, R.M.: The resource-based theory of competitive advantage: Implications for strategy formulation. California Management Review 34, 119–135 (1991)
10. Gushin, V.: Psychological countermeasures during space missions: russian experience. Journal of Gravitational Physiology 9 (1), 311–312 (2002)
11. Hamel, G., Doz, Y.L., Prahalad, C.: Collaborate with your competitors, and win. In: Hamel, G., Doz, Y.L., Prahalad, C. (eds.) Harvard Business Review, pp. 133–139 (1989)
12. Hennes, D., Tuyls, K.P., Neerincx, M.A., Rauterberg, G.W.M.: Micro-scale social network analysis for ultra-long space flights. In: The IJCAI-09 Workshop on Artificial Intelligence in Space, Pasadena, California, USA (2009)
13. Hennes, D., Tuyls, K.P., Rauterberg, G.W.M.: State-coupled replicator dynamics. In: 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2009), pp. 789–796 (2009)
14. Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Li, P., Lee, K.W.: Bio-inspired multi-agent collaboration for urban monitoring applications. In: Liò, P., Yoneki, E., Crowcroft, J., Verma, D.C. (eds.) BIOWIRE 2007. LNCS, vol. 5151, pp. 204–216. Springer, Heidelberg (2008)
15. Pfeffer, J., Salanick, G.R.: The external control of organizations. Harper and Row, New York (1978)
16. Rauterberg, M., Neerincx, M., Tuyls, K., van Loon, J.: Entertainment computing in the orbit. International federation for information processing 279, 59–70 (2008)
17. Tomasello, M.: The Cultural Origins of Human Cognition. Harvard University Press, Harvard (1999)
18. Vanderelst, D., Ahn, R.M., Barakova, E.I.: Simulated trust: Towards robust social learning. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, pp. 632–639 (2008)
19. Vanderelst, D., Ahn, R.M., Barakova, E.I.: Simulated trust: A cheap social learning strategy. Theoretical Population Biology 76, 189–196 (2009)
20. Voynarovskaya, N., Gorbunov, R., Barakova, E., Ahn, R., Rauterberg, M.: Nonverbal behavior observation: Collaborative gaming method for prediction of conflicts during long-term missions. In: Yang, H.S., Malaka, R., Hoshino, J., Han, J.H. (eds.) ICEC 2010. LNCS, vol. 6243, pp. 103–114. Springer, Heidelberg (2010)
21. Whiten, A., Spiteri, A., Horner, V., Bonnie, K.E., Lambeth, S.P., Schapiro, S.J., de Waal, F.B.: Transmission of multiple traditions within and between chimpanzee groups. Current Biology 17 (12), 1038–1043 (2007)