Mutual Adaptation Enhanced by Social Laws

Claudia V. Goldman* **Jeffrey S. Rosenschein**
Institute of Computer Science
The Hebrew University
Givat Ram, Jerusalem, Israel
clag@cs.huji.ac.il, jeff@cs.huji.ac.il

Abstract

We investigate the issue of multiagent learning with focus on the interactions among the learning agents. Each agent learns directly from the other agents that operate in the same environment. Each agent computes a function that expresses its satisfaction level after it performs an action, and after the other agents have expressed their degree of approval of the agent's behavior. Each agent adjusts its behavior based on an adaptation criterion applied to its satisfaction level. Our aim is that the agents learn to adapt their behaviors.

In this work, we impose a social law on the agents' behavior while they learn to adapt to the other agents in the environment. Each agent has attached a level of cooperation that induces it to follow the rule to different degrees. We show that the level of mutual adaptation of the agents can increase when they follow a social law.

Keywords: Multiagent Learning, Social Laws.

*Supported by the Eshkol Fellowship, Israeli Ministry of Science

1 Introduction

The Satisfaction Level approach [4] is a new way to look at the process of adaptation in a multiagent system. The problem we face is that of a society of agents (not necessarily cooperative nor homogeneous) that act in the same environment. In the general framework, each agent has a set of actions it can perform, and a set of goals it needs to achieve. This includes the case when the agents' actions come from the same set and they might use the same resources. In such situations, agents' actions might conflict. This is the well-known problem of coordination in Distributed Artificial Intelligence (DAI).

The approaches taken to induce the agents operating in a multiagent system to coordinate can be divided into two classes: those that program the agents with defined protocols and those that let the agents learn how to behave. The first approach suffers from known disadvantages when comparing non-learning algorithms to learning algorithms. The first approach depends on predefined protocols of behavior that are hardly robust in dynamic environments. In addition, when agents build models of the other agents, there is always a level of uncertainty about the agents' next actions.

The learning approach in multiagent systems has focussed mainly on agents that learn from their environment [5, 8, 15, 14, 10, 11, 9, 1]. Thus personal influences of specific agents on other specific agents cannot be taken into consideration. For example, agents could not express their sympathy/anger for another agent's action, agents could not repay other agents or retaliate against them for their help/cooperation, agents could not train another agent directly how to behave. Another problem is there might not exist a central agent or program that can follow everything that is happening in a specific environment.

Littman also looked at Markov games that deal with the specific actions taken by the agents and how they pay back other agents. In his paper [6] he mentioned that this is a simple model that cannot deal with agents that learn to cooperate. Mataric [7] also mentions a function that expresses three types of reinforcement including the direct reinforcement from the agents acting in the same world. In this case the agents are homogeneous.

Our research focuses on four main characteristics of the agents:

- the agents are autonomous, their behavior is not directed by a defined protocol or any central authority.
- the agents are heterogeneous, i.e., they can be designed by different designers, and for different purposes; they might have selfish interests. The agents do not necessarily have to be cooperative. The agents might lie and distract the learning agent.
- the agents are capable of adapting their behavior to the behavior of the other agents. The agents develop coordination protocols by adapting to each other, while executing actions to achieve their goals.
- the agents might act in a multiagent environment; i.e., they might interact with other agents to cooperate, coordinate their actions, to avoid conflicts.

The model we present in this work does not explore a learning algorithm in which the agents learn a specific concept. The agents adapt their strategic behavior dynamically, during their interactions. Given, for example, that two agents have to perform actions using the same resource, one interesting question is how they could decide which action to perform so that they can achieve their goals. With our approach, the overall system does not deadlock, and each agent tries to get benefit from the existence of the other agents.

We propose an adaptation algorithm that takes into account the personal interests of each agent, and the reaction (feedback) from the other agents to each one of the others. The model of our multiagent system is based on a teacher-learner concept. The teacher (that needs to be understood as a trainer) tries to train the learner how to behave for its own (the trainer's) benefit. In the algorithm we suggest, the agents need to compute a parameter called the satisfaction level of each agent. This value characterizes the agent's behavior at each time, given the action it chose and the feedback it got from its trainer. Based on a criterion applied on this value, the agents decide whether they will change their strategy of action. The aim of the algorithm is that the agents change their actions so that they will increasingly adapt to each other during the time they interact. Experiments showing the emergent behaviors we observed were described in [4].

We have also tested whether the agents' characteristics can change the influence of the algorithm in such a way that the agent will be in a better situation relative to its goal. This can be expressed by the agent's type. We have checked in addition to the quality of the algorithm, the influence of different agents' types that run the algorithm. The types we considered included three alternatives: selfish, altruistic, and cooperative. A selfish agent cares only for its state, an altruistic agent cares only about the other (how to best satisfy the other), and a cooperative agent regards equally its state and all the other agents' states. We are interested in a number of related questions: How will pairs of different types behave, and adapt to one another? Can we learn which type an agent's designer should choose for its agent? What will happen if the designer chooses to make his agent selfish, what results might that lead to, or might it be better to design it as a random agent?

In order to make the algorithm more flexible, (i.e., so that the adaptation algorithm is as little controlled/fixed as possible), we let the agents adapt the threshold (under which they change their actions) dynamically.

We have combined the idea of state changing rules [3] or social laws [12, 13] with the adaptation algorithm. The work of Shoham et al. constrains the agents' choices so that they will not choose actions that will cause them any conflicts (e.g., in the traffic domain, they might give the agents a centralized protocol, such as one in which each agent holds the green light for a predefined time and then switches to the red light). The work of Goldman et al. lets the agents continue acting as they would have when no social norm is applied. The social law is "additive"; it is imposed on top of their individual choices among actions. That is, each agent might invest additional effort for the benefit of the society, by following the social law. This additional effort is quantified in the "level of cooperation" parameter, that we added to each agent. It lets the agent compromise or not during the interaction with the other agents. The question is whether a social law can enhance the level of mutual adaptation in a multiagent system where the agents adapt to each other dynamically.[1]

After the simulations we have run we can suggest a pair of levels of cooperation, that are likely to improve the level of mutual adaptation of the system.

2 A Multi-Adaptive Agent System

We proposed an adaptation algorithm in [4], where we looked at a multiagent system as composed of teachers and learners. Each agent plays at each time both roles. Each agent chooses an action to perform, and then it computes its level of satisfaction. This function is composed of two terms. One regards the own gain of the computing agent (i.e., the learner) after having performed the action it chose, and considering the state to which the world moved. The second term expresses the other agents' reaction to the learner action (i.e., this is a feedback value sent to the learner by each of its current trainers).

The learner will keep choosing the same action as it chose when it computed its satisfaction level, as long as this value does not drop below a certain given threshold. If its satisfaction level is indeed below it, the agent will change its strategy. The value of this function will decrease when the other agents are too angry with the learner's action, and their feedback is very negative, for example. Since the algorithm's aim is to adapt the agents, it causes the learner agent to compromise when its behavior is too greatly irritating the other agents in the system. On the other hand, the learner agent might be performing an action that is not so favorable for its own purposes, but it was lead to choose it because the other agents caused it to choose. Since the satisfaction level also takes into account the learner's own gain, the agent might be able to choose a better action for its benefit when its own gain gets too small and again the satisfaction level will drop below the threshold given. This shows that although our approach is to have agents that learn to adapt to each other, we also have to consider the self interests of each of the agents during the process of mutual adaptation.

In the work presented in [4] we investigate the influences that agents with different types might have on the level of adaptation of the system. We handled nine situations that combine agents of the following types: selfish, altruistic, cooperative. In this work we will concentrate on cooperative agents, which means that each agent weighs equally its own gain and the feedback of others. In addition, we here combine the mutual adaptation approach together with the social laws paradigm [3, 12, 13]. The main question we want to investigate is whether a social law imposed on a multiagent system can enhance the level of mutual adaptation of the agents.

In order to do this, we need first to define a criterion for evaluating the *level of mutual adaptation* of a multiagent system. In this concept, we want to express how good the agents are adapted at a given time. If for example, the agents coordinate when they choose the same actions, then we can quantify their adaptation by counting the times they choose the

[1] It is possible to interpret the concept of "level of cooperation" to electronic cash, and then it means how much money an agent is willing to invest to adapt, and to avoid conflicts.

same actions. As we will show later in our experiments, we might also be interested in agents that choose opposite actions (e.g., one agent chooses the red light, and the other the green light in the traffic lights domain). Then we consider the agents to be successfully adapted based on the number of times they have chosen the opposite actions.

In our case, we compute the level of mutual adaptation of the agents after segments of Δt units of time. At the end of a simulation we calculate the accumulated level of mutual adaptation of the system. This concept enables us also to compare multiagent systems, by telling us in which system the agents are more adapted. If the agents choose the "correct" set of actions at each unit of time, then we can say that there is perfect coordination in the system. Otherwise, if the agents coordinate $K < \epsilon$ times, we can claim that the agents in this system are ϵ adaptive.

Second, we need to define the *level of cooperation* of an agent [3, 2]. This is a parameter that the agent's designer decides on and attaches to its agent. The problem of determining the level of cooperation for an agent is domain dependent. This value characterizes the agent such that the agent will behave according to its strategy, but it is also ready to invest more effort for the benefit of the whole system. This effort is quantified in the level of cooperation. In our case, a state of the system is better than another when there are more agents that are willing to compromise. We assume that if more agents will compromise then it will be easier for the agents to adapt to one another. The modification we did to our original algorithm is that we added a social law that the agents will follow (given that their level of cooperation is larger than zero). Each Δt units of time each agent at its turn checks the following condition and decides about its next action appropriately.

$$If\ (Level\ of\ mutual\ adaptation)_i \leq (Level\ of\ cooperation)_j$$

$$Agent\ j\ compromises$$

$$else\ Agent\ j\ continues\ with\ its\ current\ strategy$$

where i indexes the current segment of time, and j indexes the agents in the system. The agent "compromising" means that the agent will adjust its threshold to be larger, for example in our current case (i.e., it will change its action earlier, the teacher does not need to wait too long for the learner to change the action that the teacher is not happy with). The level of mutual adaptation lies between zero and Δt. If the level of cooperation of agent i is equal to Δt then until the agents do not get to perfect coordination in the current segment of time, agent i will keep compromising. If the level of cooperation of agent i is larger than Δt then the same will happen. An agent cannot adapt more than Δt units. If the level of cooperation of agent i is zero then agent i will not follow the social law, and will behave according to the original algorithm. If the level of cooperation is larger than zero but smaller than Δt then we can distinguish between two cases. First, when the level of cooperation of agent i is larger or equal to the level of mutual adaptation, agent i compromises according to the rule. If the level of cooperation of agent i is strictly smaller than the level of mutual adaptation, the social law does not influence in this range. Agent i does not want to compromise that much.

In the next Section, we will present the experiments performed in the trafic lights scenario, to answer our central question: *Could Social Laws enhance the level of mutual adaptation of a Multi-Adaptive Agent System?*

3 The Traffic Light Domain — Experiments

An interesting domain in which we can investigate multiagent coordination strategies is the traffic domain. Cars drive on roads with different priorities. To avoid collisions, the traffic through intersections need to be controlled. Having agents controlling road intersections and junctions of secondary roads into main roads automatically and adaptively will lead to a better flow of cars on the roads given the constraints of the road at real time (i.e., the entire road system will benefit from that). Agents responsible for the main roads can be considered to be egoistic while agents responsible for secondary roads will need to compromise and let their cars wait for longer periods until they can go through.

Another realistic domain, very similar to the traffic domain, is a communication network where agents can be assigned nodes and they have to decide which messages would pass among those that arrive to the node from different channels.

To learn about adaptive strategies that can be developed by agents dynamically, we have been experimenting with a single traffic intersection [4]. There are two agents; i.e., two traffic signal controllers. A_1 is responsible for the traffic flow in the vertical direction, while A_2 is in charge of the horizontal road. Time is considered to be discrete. The traffic at each road is represented by a simple queue. At each period of time, an amount of X new cars is added to each agent's queue. X equals the number of cars that enter the same road in the previous tick of time or this number increased or decreased by one. The agent that holds the green light can free at most a predefined number of cars from its road.

The case study we face in this paper consists of two agents, A_1 and A_2, each one choosing an action $a^t_{A_i}$, at each period of time t (time is considered discrete). The agents' goals are to maintain their queue as short as possible given the cars that enter each queue at each point in time, and to avoid collisions (when both agents are allowed to choose the green action at the same time). We want the agents to learn to adapt to each other, by coordinating their actions in the intersection through continuous interactions.

At time t, each agent knows its state $x^t_{A_i}$ (i.e., how many cars A_i has at time t), and what is the reaction of its teacher to the learner's behavior so far. After getting the feedback from its teacher, the learner is able to compute its level of satisfaction as explained in Section 2. The satisfaction level function we have implemented is the following:

$$SL(learner, t) = LearnerType * OwnGain(learner, t) + (1 - LearnerType) * Feedback(teacher, t)$$

$$+ punish(agent, t)$$

where,

$$OwnGain(learner, t) = Cars(learner, t) + \theta * TimeGreen$$

$$Feedback(teacher,t) = \begin{cases} 0 & \text{if Cars(teacher,t) = 0} \\ -Cars(teacher,t) - TimeFunc(TimeRed) & \text{otherwise} \end{cases}$$

In this case the LearnerType is cooperative. The agent compares the value expressing its level of satisfaction to a threshold that represents its lower limit for continuing to provoke, and below it, it starts compromising. Based on this result, the agent chooses its next action at time t+1.

We have implemented the algorithm presented in Section 2 in the traffic domain. Both agents might choose to perform a red or a green action. Collisions might occur when both agents choose the green light, and both roads will be blocked when both agents choose the red light.

For all our experiments, we created twenty four files with data representing the number of cars entering each road. These are randomly created files that differ in the initial number of cars, and in the random value that is added to the previous number of cars that entered the road (e.g., zero, one or -1). We did not limit the capacity of the roads.

For these experiments, a third term was added to the original satisfaction level function, to punish the agents when they collide or block the roads. This is a first attempt to combine our model with the classical reinforcement learning scenarios. We let the world punish both agents, when they collide or block the intersection. In order to break symmetry, we punish only one agent each time that both of them do not coordinate. This agent is chosen randomly.

We tested the agents behavior, by computing the level of accumulated mutual adaptation at the end of a simulation of 1000 ticks of time. We run each simulation for each pair of input files we have created (a total number of twelve pairs). We chose Δt to be five units of time. We let each agent have in each simulation a level of cooperation that lies between zero and five.

First, we checked the similarity of the inputs, by running the simulations for each pair of inputs, for every pair of levels of cooperation, and we computed the standard deviation of the mutual adaptation achieved. For example, see Figure 1.

Then for each pair of input files, we also plot the three dimensional graph, for any pair of cooperation levels, and the level of mutual adaptation achieved. See Figures 2,3,4,5,6,7:

These results were gotten when A_1 was the agent that started with the green light and A_2 with the red one. We have also checked whether these results remain consistent when A_2 starts with the green light. Indeed, we got similar results. In ten cases the maximal level of mutual adaptation is gotten when both agents were given cooperation levels of five (the maximal level enabled). In the two remaining cases both agents were given cooperation levels equal to four. In all the cases, the accumulated mutual adaptation level is larger than the one gotten when the social law was not applied. In addition, we notice that a social law for specific pairs of cooperation levels might harm the mutual level of coordination of the agents. In this case, the difference between one agent's cooperation level and the other's is at least three, with one agent having a very small level, and the other a very large one. Even though the environments were created randomly, the agents behaved consistently; simulations for different input files arrived at similar levels of accumulated mutual adaptation.

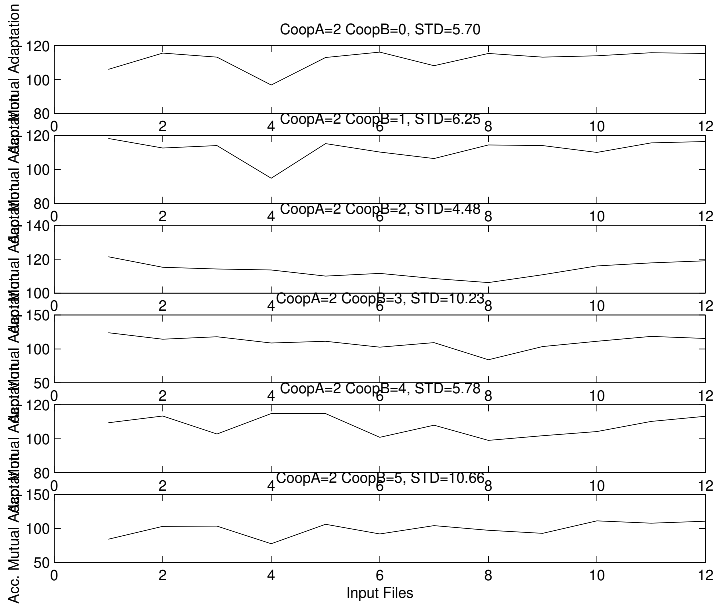

Figure 1: Accumulated Mutual Adaptation achieved when A_1's level of cooperation is 2 and A_2's varies from 0 to 5

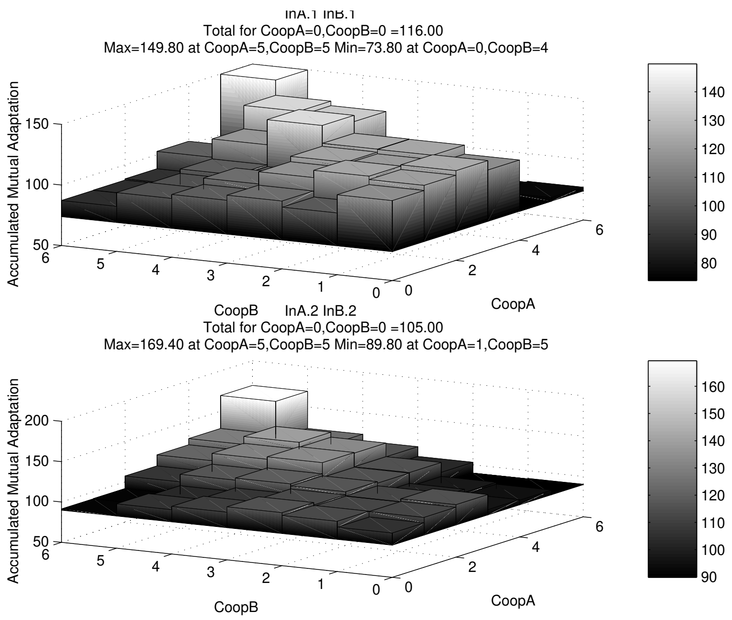

Figure 2: Accumulated Mutual Adaptation achieved for inputs 1 and 2

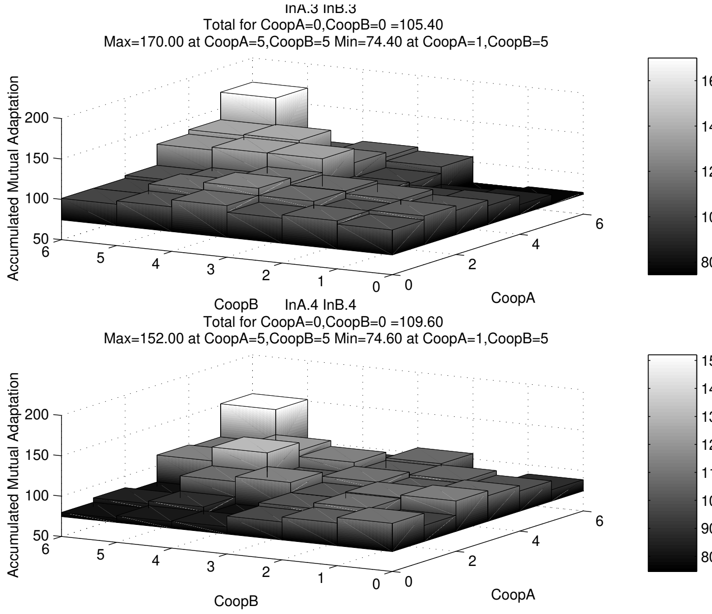

Figure 3: Accumulated Mutual Adaptation achieved for inputs 3 and 4

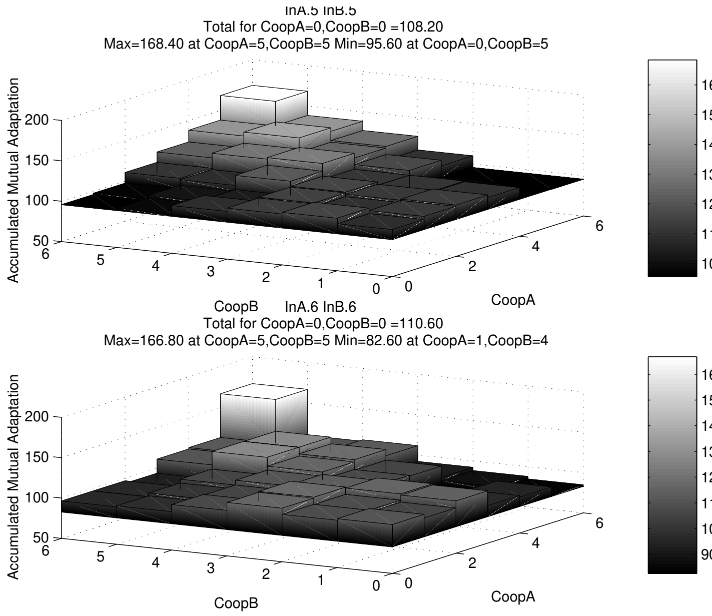

Figure 4: Accumulated Mutual Adaptation achieved for inputs 5 and 6

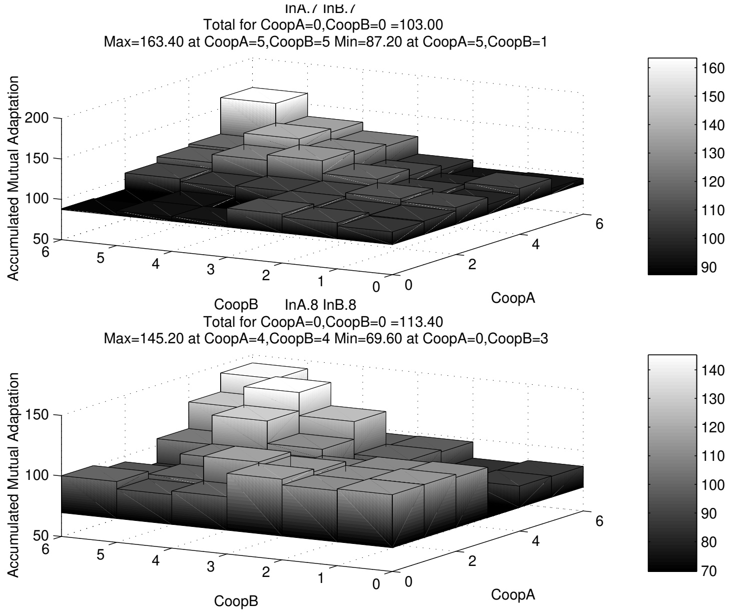

Figure 5: Accumulated Mutual Adaptation achieved for inputs 7 and 8

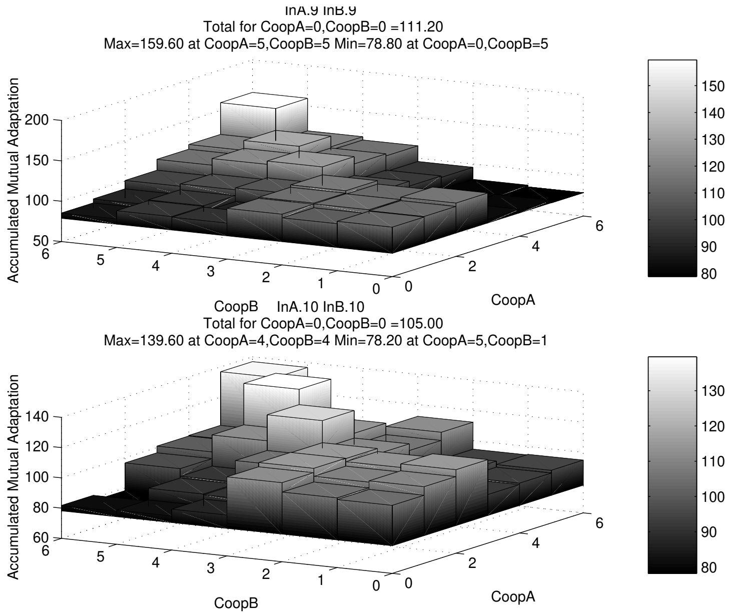

Figure 6: Accumulated Mutual Adaptation achieved for inputs 9 and 10

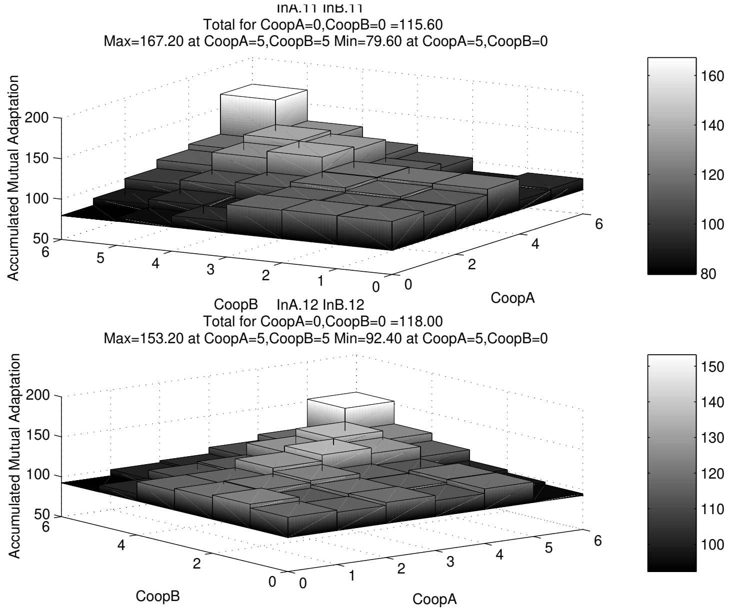

Figure 7: Accumulated Mutual Adaptation achieved for inputs 11 and 12

The evidence suggests that if two agents interact in an environment similar to this, it is worthwhile assigning them cooperation levels equal to five (the maximal values).

Then, we compare the accumulated mutual adaptation in the following cases: when both agents choose their actions randomly, without regard to the social law, when one agent is random, and the other is a cooperative learning agent (whose level of cooperation is zero), when both agents are cooperative and learn according to the adaptation algorithm but they do not follow the social law, and when both agents are cooperative, choose their actions based on the adaptation algorithm, and follow the social law. We present a table (see Figure 8) with the level of mutual adaptation achieved in each one of these cases. For the second column, we point out the pair of levels of cooperation that achieved the maximal level of mutual adaptation.

If we look at the average case, then when both agents are random they will achieve a level of mutual adapatation of 500 ± 6.7. When one agent is random and the other adaptive, they coordinate in $\sim 500 \pm 24.5$ cases. When both agents are adaptive, but they do not follow the social law, they get to a level of 553.5 ± 27 mutual adaptation. When both agents are adaptive and do follow the social law, they coordinate in 793.6 ± 50 cases out of 1000 units of time. Thus, we observe that it is worthwhile having adaptive agents; when the agents were adaptive, they reached improved coordination levels with or without the social law. Moreover, when the social behavior norm was added to the multi-adaptive agent system, we significantly improved the level of mutual adaptation of the agents.

#Simulation	Two Cooperative Learning Agents Both cooperation levels = 0	Two Cooperative Learning Agents with cooperation levels that maximize the level of mutual adaptation	A1 is a random agent A2 is a cooperative learning agent with cooperation level = 0	Both agents are random
1	580	(5,5) 749	542	489
2	536	(5,5) 847	483	486
3	571	(5,5) 850	475	510
4	489	(5,5) 760	535	489
5	557	(5,5) 842	470	492
6	553	(5,5) 834	482	493
7	540	(5,5) 817	469	490
8	567	(4,4) 726	498	488
9	556	(5,5) 798	492	491
10	525	(4,4) 698	522	485
11	578	(5,5) 836	492	486
12	590	(5,5) 766	523	482
Average	553.5	793.6	498.6	490

Figure 8: A comparison of the Accumulated Mutual Adaptation values achieved

4 Conclusions

Our work focuses on *Multiagent Learning Systems* which are composed of different autonomous agents that learn directly from one another. We have suggested the use of a trainer-learner model, where the trainer instructs the learner agent how the learner should behave in order to please the trainer. The learner chooses its actions according to its level of satisfaction, which is based on its own gain (resulting from its actions) and on the teacher's feedback. This approach is useful when there is a multiagent system, that we do not want to control centrally, but we want the agents to learn how to coordinate dynamically, without the need of a given coordination protocol. The agents react to each other and by applying an adaptation criteria they decide which action to perform.

We have implemented our model in the traffic light domain. Each agent has been given a level of cooperation that expresses how much additional effort it is willing to invest for the benefit of the system. We showed that if we add a social law in the multi-adaptive agent system, we can significantly improve the level of mutual adaptation of the agents.

References

[1] Minoru Asada, Eiji Uchibe, and Koh Hosoda. Agents that learn from other competitive agents. In *Machine Learning: Proceedings of the Twelfth international conference, Workshop on Agents that Learn from Other Agents*, 1995.

[2] Claudia V. Goldman. Emergent coordination through the use of cooperative state-changing rules. Master's thesis, Hebrew University, 1993.

[3] Claudia V. Goldman and Jeffrey S. Rosenschein. Emergent coordination through the use of cooperative state-changing rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 408–413, Seattle, Washington, 1994.

[4] Claudia V. Goldman and Jeffrey S. Rosenschein. Societies of intelligent software agents. to be submitted. *Adaptive Behavior Journal, Special Issue on Simulation Models of Social Agents*, 1998.

[5] John J. Grefenstette. Lamarckian learning in multi-agent environments. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 303—310, 1991.

[6] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning: Proceedings of the Eleventh international conference*, pages 157—163, 1994.

[7] Maja J. Mataric. Learning to behave socially. In *Proceedings From Animals to Animats 3, Third International Conference on Simulation of Adaptive Behavior*, pages 157—163, 1994.

[8] Maja J. Mataric. Reward functions for accelerated learning. In *Machine Learning: Proceedings of the Eleventh international conference*, pages 157—163, 1994.

[9] Tuomas W. Sandholm and Robert H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems Journal Special Issue on the Prisoner's Dilemma*, submitted.

[10] S. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, 1994.

[11] Sandip Sen and Mahendra Sekaran. Multiagent coordination with learning classifier systems. In *Workshop on Adaptation and Learningin Multiagent Systems at the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Quebec, 1995.

[12] Yoav Shoham and Moshe Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Jose, California, July 1992.

[13] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, page to appear, 1993.

[14] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Machine Learning: Proceedings of the Tenth international conference*, pages 330—337, 1993.

[15] Christopher J.C.H. Watkins and Peter Dayan. Technical note Q-learning. *Machine Learning*, 8:279—292, 1992.