First Cognitive Capabilities in the Anticipatory Classifier System

Wolfgang Stolzmann* **Martin V. Butz**** **Joachim Hoffmann***
David E. Goldberg**

* Institute for Psychology III
University of Würzburg
Germany
{stolzmann,hoffmann}@psychologie.uni-wuerzburg.de

** Dep. of General Engineering
University of Illinois
Urbana-Champaign, IL 61801
{butz,deg}@illigal.ge.uiuc.edu

Abstract

This paper adds a new viewpoint to the Anticipatory Classifier System (ACS). It approaches the system from a psychological perspective and thus provides new insights to the current system. By simulating previously published rat experiments, the paper compares the behavior of the ACS with the behavior of the rats. Two further cognitive mechanisms are introduced to the ACS resulting in an animal-like behavior in the presented simulations. Moreover, the paper gives empirical evidence that the evolving generalized, internal environmental model is usable in the ACS for the mental adaptation of actions and thus enables reinforcement learning by mental simulation.

1. Introduction

Learning Classifier Systems (LCSs) learn generalized stimulus-response rules which are modified according to the reward experienced in an environment. The applications cover 'one-step' problems, mainly used for data-analyzing tasks, but also 'multi-step' problems where the reward is deferred. The XCS classifier system (Wilson, 1995), which learns based on an accuracy measure of its reward prediction, proved the generalization and reward-learning capabilities of LCSs. More than twenty years ago, Holland outlined the possibilities of how to evolve environmental representations with a classifier system (Holland and Reitman, 1978). Later, he developed the possibility of doing lookahead-planning with classifier systems without changing the rule structure, nor the production rules in the system (Holland, 1990). In 1991 the first implementation of the proposed system was published (Riolo, 1991), which was called CFSC2. Despite the capability of lookahead-planning and the proposed multi-step predictions, the system appears to be confusing. The coupling of actions is done over more than one step which results in an unnecessary complexity.

The Anticipatory Classifier System (ACS) (Stolzmann, 1997) enhances the structure of the stimulus-response rules in LCSs by an effect part. Thus, it uses a straight-forward approach to enable an LCS to do lookahead-planning and form an internal model. Drescher published another approach with three part rules he calls schemata (Drescher, 1991). Although many similarities can be deduced, the ACS differs from Drescher's model in its representational simplicity and its population-based rule generation. One of the foundational problems in Drescher's approach is how to notice the result of an action. The ACS approaches this problem by using a learning process derived from psychology, which is called the Anticipatory Learning Process (ALP). It generates new rules in a population in a much more directed way than the random generation – combined with Genetic Algorithms (GAs) – in most of the LCSs. However, the rules still maintain a certain generality. It was shown that the ACS is able to learn latently (Stolzmann and Butz, 2000), meaning, to be able to learn an internal environmental model without experiencing any reward. Moreover, it is able to plan in order to improve the performance in latent learning tasks. Recently, the application of a GA in the ACS proved that it is possible to combine the directed specialization pressure of the ALP with a genetic generalization pressure in order to generate accurate, maximally general rules in the system (Butz et al., 2000). The capability of generating more general rules distinguishes the ACS from the Dyna architecture in reinforcement learning (Sutton and Barto, 1998).

Except for the planning, which was solely applied in latent learning tasks (Stolzmann and Butz, 2000), the cognitive capabilities in the ACS were not further investigated as yet. The objectives of this paper are threefold. (1) It approaches the ACS from the psychological viewpoint and asks if cognitive processes are going on. Is the ACS really learning according to learning theories in psychology? (2) It discusses whether the behavior of the ACS is comparable to animal behavior in appropriate test environments. (3) It shows that the ACS is able to use the evolving environmental representation to adapt its behavior in an environment without experiencing reward connections actively, which is not possible with the common reinforcement learning techniques. The last point is strongly related to the insights discussed elsewhere (Rosen, 1985 and Drescher, 1991) that it is mandatory for a system to represent anticipations in some form to be able to reason and adapt behavior internally.

The paper approaches the objectives by carefully analyzing the processes in the ACS in the next section and comparing them directly to the underlying psychological learning theory. Next, the paper simulates previously published experiments with rats (Colwill and Rescorla, 1985 and Colwill and Rescorla, 1990). By introducing two new algorithms which possibly reflect cognitive processes in the rats, it will be shown that the ACS can mimic the observed behavior of rats, and can adapt its behavior through mental simulation. Finally, the results will be discussed in further detail.

2. The ACS in Detail

The ACS combines an LCS with the psychological learning theory of anticipatory behavioral control (Hoffmann, 1993). In order to realize this theory in the ACS, the structure of a classifier is enhanced by an effect part. This effect part is the key to the anticipatory learning process (ALP). It enables the ACS to learn a complete internal representation of an environment, consequently allowing for many cognitive processes.

2.1 The framework of the ACS

The framework of the ACS is strongly related to other LCSs. This section gives an overview of the framework as well as reveals the similarities to, and differences from, other LCSs.

2.1.1 The Interaction with an Environment

Like other LCSs, the ACS interacts with an environment. At a certain time t, it perceives the actual situation in the current environment as a string $\sigma(t)$. Each attribute in the string does not need to be coded binary, but may only take discrete values. Thus, a situation $\sigma(t) \in \{s_1, ..., s_m\}^L$, where L is the length of the string, m is the number of possible different values, and $s_1, ..., s_m$ are the different possible values. The interaction occurs with executable actions α, which are also coded with discrete values (i.e. $\alpha \in \{r_1, ..., r_n\}$ where n is the number of different actions and $r_1, ..., r_n$ are the different actions). Finally, after the execution of an action $\alpha(t)$, it perceives a payoff $\rho(t)$, which is a simple real value, and the successive situation $\sigma(t+1)$.

2.1.2 The Representation of the Knowledge

The knowledge of the ACS is represented by a population of classifiers similar to LCSs. However, a classifier has an additional effect part, a mark, and a quality measure. Thus, each classifier consists of the following parts:

- The condition part C specifies the prerequisites for the application of the rule.
- The action part A specifies an action that the classifier wants to execute.
- The effect part E anticipates the changes caused by action A under the condition C.
- The mark M specifies all different attributes of the situations σ where the classifier did not anticipate correctly.
- The quality q measures the accuracy of the anticipations.
- The reward measure r predicts a combination of the immediate payoff ρ of the environment and the reward measure of the best possible action in the resulting state.

The condition and the effect part consist of the values perceived from the environment and of '#'-symbols (i.e. $C, E \in \{s_1, ..., s_m, \#\}^L$). A '#'-symbol in an attribute of the condition part, which is called 'don't-care' symbol, matches any perceived value in this attribute. This symbol enables the ACS to ignore irrelevant attributes. A '#'-symbol in an attribute of the effect part, which is called 'pass-through' symbol, predicts that this attribute will remain the same (i.e. the corresponding value in the perception does not change after the execution of an action). The action part consists of any action which is possible in the environment (i.e. $A \in \{r_1, ..., r_n\}$). The mark has the structure $M = (m_1, ..., m_L)$ with $m_i \subseteq \{s_1, ..., s_m\}$. If a classifier is applied in the situation $\sigma(t) = (\sigma_1, ..., \sigma_L)$ and does not work correctly, then the mark $M = (m_1, ..., m_L)$ of the classifier is modified by $\sigma(t)$: If the classifier was marked before, then $m_i = m_i \cup \{\sigma_i\}$, otherwise $m_i = \{\sigma_i\}$ for all i. Thus, the maximum size of M is $m * L$. Consequently, the mark gives information about both, that the classifier anticipates incorrectly sometimes and about the properties of the situations in which it does anticipate incorrectly. Using these properties, the ACS is able to differentiate the classifier from these unsuccessful situations.

2.1.3 A Behavioral Act

A behavioral act in the ACS is similar to other LCSs but differs in the anticipatory parts. Further, the ACS does not use a message list. It starts by forming a match set M_{set} out of the current population. The match set contains all classifiers whose conditions are satisfied by $\sigma(t)$. Next, an action is selected. Various explore/exploit biased action selections are possible. Pure exploration would be to choose at any point the action randomly while pure exploitation would be to choose the best action (specified by the classifier which predicts the highest reward measure r). In our case, we intend to simulate animal behavior and thus cannot distinguish between different action selection schemes at different times. Therefore, we always use a weighted roulette wheel selection scheme which chooses an action out of the prediction array. The prediction array specifies the average reward predicted for each action present in the match set, which is similar to the prediction array in XCS (Wilson, 1995). After the action is selected, an action set is formed out of the match set containing all the classifiers that specify the chosen action. Then, the action is executed followed by the payoff $\rho(t)$, and the perception of the next situation in the environment $\sigma(t+1)$. Considering $\rho(t)$ and $\sigma(t+1)$ the action set is modified by the ALP and the reinforcement learning algorithm. Figure 1 gives an overview of the framework. In the next section we will introduce the ALP and the reinforcement component and reveal the psychological motivations.

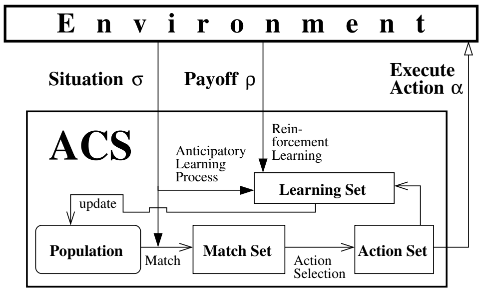

Figure 1: The ACS learns from from the immediate payoff $\rho(t)$, as well as from the comparison between its anticipations and the consecutive perceptions $\sigma(t+1)$.

2.2 The Learning Processes in the ACS

The Learning in the ACS begins with completely general knowledge (i.e. for each possible action there is a classifier, whose C and E parts contain only '#'-symbols). Two learning processes gradually increase the knowledge. While the ALP is directly derived from the psychological theory of anticipatory behavioral control (Hoffmann, 1993) and forms the internal environmental representation, the reinforcement learning component realizes the reward distribution.

2.2.1 The Theory of Anticipatory Behavioral Control

Many observations in psychology have led to the learning theory of anticipatory behavioral control. The first principle is that learning can only occur if a need is present that wants to be satisfied. The observation of an operational drive in humans, in addition to the known drive of satisfaction, showed that learning cannot rely solely on the satisfaction of the basic needs such as food, water, or sex. Rather, another type of need must exist in humans. In animal learning it was shown that rats are able to learn without any direct reinforcement (i.e. they learn latently, Tolman, 1932). This and many other experiments showed that a need for accurate anticipations must exist in higher mammals.

The theory of anticipatory behavioral control evolved out of this observation. Figure 2 shows the basic idea of the theory. First, a behavior R is always accompanied by the anticipated consequences E_{ant} and the actual given situation abstracted to a condition S_{start}. Second, a continuous comparison takes place between the anticipations and the successive perceptions. If the comparison was valid or invalid, it leads to an increase or decrease of the relation between the anticipation and the corresponding stimulus-response relation, respectively. Finally, inaccurate anticipations lead to further differentiation of the conditions.

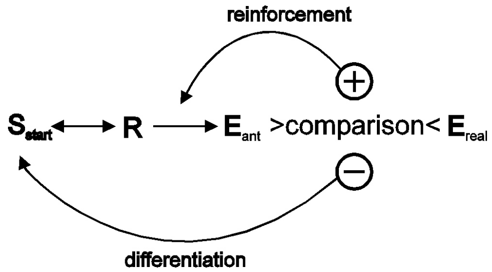

Figure 2: The anticipatory behavioral control theory states that learning is based on the comparison of the anticipated effects E_{ant} and the real effects E_{real}.

This learning theory is translated into an artificial system (i.e. the ACS, Stolzmann, 1997). The following section explains how this is done and directly compares the involved mechanisms to the theory.

2.2.2 The Anticipatory Learning Process

In order to realize this learning theory in an artificial system, it is necessary to represent the anticipations in some form. This could be done by a recurrent neural network or by explicitly linking rules to possible effects represented by other rules. The most straight-forward approach, though, is to build S_{start}-R-E_{ant} rules directly. This is done by the ACS. An action A is always accompanied by the anticipated effects E and the actual situation represented by the condition C as stated in the first point of the theory.

The ALP together with its application in the whole action set realize the other two points in the learning theory. The ALP distinguishes the following cases:

- First, a classifier anticipates the resulting situation $\sigma(t+1)$ incorrectly. As in the theory, this results

in a decrease between the relation of the anticipation and the stimulus-response part, represented by a decrease in the quality of the classifier. A decrease in the quality of a classifier indeed results in a decrease in this relation as the overall relation between condition-action part and effect part is represented in an action set of rules and not one single rule. Thus, a decrease of the quality of one rule results in an implicit increase in the quality of the other rules (which may be more accurate) and thus an increase between other relations (*S*-*R*)-*E*. Furthermore, the classifier is marked by $\sigma(t)$. The mark is going to help the ALP to differentiate the rule from this unsuccessful situation. In more detail, three different cases are distinguished:
  - In the *correctable case* the rule of a classifier can be further specialized in the effect part (i.e. '#'-symbols can be changed to specific values). In this case, the ALP generates a new classifier which specifies all changing attributes. Thus, a rule that was believed to cause some effects produces a new rule that predicts more specific effects.
  - In the *not correctable case* the rule of a classifier cannot be made more specific (i.e. it already specified another effect which contradicts the current effect). In this case, the ALP does not generate a new classifier.
  - In the *useless case* the ALP experiences no change in the environment at all and does not generate a new classifier either.
- Second, a classifier anticipates $\sigma(t+1)$ correctly (i.e. it is in the *expected case*). In this case, the quality of the classifier is increased. With the same argument as in the first case, this results in an increase in the corresponding (*S*-*R*)-*E* relation overall. Furthermore, if the classifier has a mark M, then the ALP generates a new classifier with a more specific condition part. In the condition part C of the new classifier the attributes are specialized where $\sigma(t)$ differs from M (i.e. $\sigma_i \neq m_i$). This represents the further differentiation of the conditions stated in the theory.

q is increased and decreased by using the Widrow-Hoff delta rule (Widrow and Hoff, 1960) with learning rate $b_q = 0.05$ (i.e. increase: $q = (1-b_q)*q+b_q$ and decrease: $q = (1-b_q)*q$). If the quality of a classifier decreases under the threshold $\theta_i = 0.1$, the ACS considers it *inadequate* and deletes it. On the other hand, if the quality increases over the threshold $\theta_r = 0.9$, the ACS considers the classifier as *reliable*. The environmental model is represented by all reliable classifiers. A somewhat more technical description of the ALP can be found elsewhere (Stolzmann, 2000).

Overall, the ALP and its application in the whole action set resemble the psychological learning theory. The ACS now has the aspired operational drive, as the correct classifiers will eventually rule the population. Moreover, the ALP enables the ACS to learn latently an internal representation of the environment. However, the ALP does not resemble any other drives in the system. The ALP does learn an internal representation but it will never find out which of the effects are desirable and which ones are not. Hoffmann as well states that his theory does not cover all the influences in a system but he points out that the theory makes it possible to satisfy the other present drives in a more directed way. This ability is examined in section 3..

2.2.3 Reinforcement Learning in the ACS

The other drives of the ACS are represented by the reward measure r and the chosen action selection (see Section 2.1.3). Other, more sophisticated approaches are imaginable such as representing different needs by different attributes for different reward predictions and specific drive measures to satisfy the needs (Holland and Reitman, 1978). The reward measure r is updated using a method similar to Q-learning. It modifies the r values of the classifiers in the resulting learning set. The update is again done with the Widrow-Hoff delta rule with a learning rate $b_r = 0.05$:
$r = r + b_r * (\gamma * max_{cl \in M_{set}(t+1)}(q_{cl} * r_{cl}) + \rho(t) - r)$
$\gamma = 0.95$ is the discount factor.

3. Cognitive Behavior of the ACS

After the analysis of the present cognitive processes in the ACS, this section will compare the ACS with the behavior of rats. We hope that such a comparison will give further insight into the current cognition in the ACS as well as show the animal-like behavior of the system. Moreover, we will develop two new algorithms which will show the possibility of mental reward propagation in the ACS.

3.1 The Response-Effect (R-E) Learning Task

The experiment simulated in this section was originally investigated in order to reveal if and in what way rats evolve *R*-*E* associations (Colwill and Rescorla, 1985). Figure 3 gives an abstract view of the experiment. First, we will provide more information about the original experiment. Next, we will explain the simulation with the ACS and introduce two new algorithms. It will be shown that the ACS is able to solve the task and is able to adapt its behavior mentally.

3.1.1 Details of the Rat Experiment

Colwill and Rescorla (1985) tested 32 rats in each of their two *R*-*E* experiments. First, each rat was put into a Skinner box where two manipulanda were available (i.e. a lever and a chain). Pressing the lever led to the appearance of one type of reinforcer and pulling the

Training	Devaluation	Test
$R_1 \rightarrow E_1$	$E_1 \rightarrow LiCl$	R_1 vs. R_2
$R_2 \rightarrow E_2$		

Figure 3: In the *R-E* task first a relation between an action and the resulting reinforcer is taught. Next, one reinforcer is devalued, and finally, the activity in the test phase is monitored.

chain to the other type (sucrose or food pellet). The rats were trained for five days. Overall, they experienced 204 times one or the other response-effect association. In the first experiment, the rats only had one manipulandum present in each training session while in the second both were always present at the same time. In the latter setting the authors report a special temporal schedule procedure in order to prevent the rats from only acting upon one manipulandum.

Next, the one reinforcer was devalued by giving an intraperitoneal injection of lithium chloride (LiCl) after the consumption of the reinforcer. This method is known to quickly result in the aversion of the paired food substance. The devaluation took place without the active relation between action and effect (i.e. the food was simply presented and after the consumption devalued). Over five two-day cycles the reinforcers were discriminated. One day one reinforcer was devalued and the other day the other was reinforced. The devaluation or reinforcement procedure took half an hour. Thus, it is not really possible to say how many actions or devaluations took place during the devaluation phase.

Finally, the behavior was monitored in one twenty-minute test trial. In this trial both manipulanda were present. However, there was no reinforcement and the executed actions (i.e. pressing or pulling) resulted in no changes in the environment.

Figure 4 shows the behavior of the rats in the test phase of the first experiment. The rats learned the *R-E* relation and thus executed the action which led to the devalued reinforcer significantly less often. Note that there was a difference observable when sucrose or pellet was the reinforcer.

3.1.2 The R-E Simulation

The simulation includes two steps. In the first step, the ACS can act upon the manipulanda and consequently either press or pull. In the second step the allotted food can be consumed. The different environmental situations are coded as a four bit string. The first two bits indicate if sucrose or the food pellet is present. The second two bits indicate if the lever or chain is available. Thus, a situation $\sigma = 0110$ would indicate that the food pellet and the lever are currently present. After the consumption of the food, a payoff ρ is provided. As in the rat

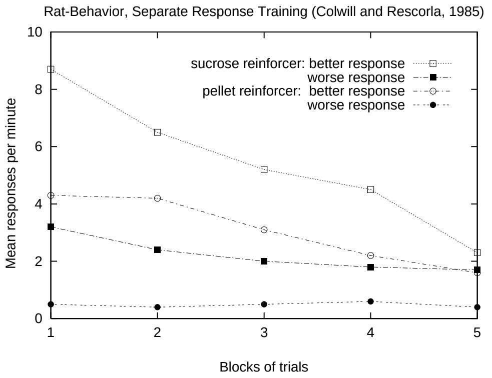

Figure 4: The behavior of rats in the response-effect (*R-E*) learning task shows that rats are able to learn the *R-E* relation.

experiments, the simulation is divided into three phases. In the first phase, the two steps are executed resulting always in a payoff of $\rho = 100$. Herein we simulated the variation of the experiment where either the one or the other manipulandum is present in the first step. When both manipulanda were present, we encountered problems getting the ACS to execute both actions; much like the rats. However, in this case when selecting an action at random with a probability of 0.1 and otherwise selecting as before, the results are very similar to those presented herein. In the second phase, neither manipulandum is present and one or the other type of food is provided randomly. The consumption in this phase leads to a payoff of $\rho = 100$ for the valued reinforcer and $\rho = 0$ for the devalued one. In the third phase both manipulanda but no food are presented and the number of executions of the actions is counted. No action leads to any effect in this phase. All presented results are averaged over 100 runs.

Simulating the experiment in this two-step style reveals the limits of the common reinforcement learning and the need for an internal model of the environment. Since the evolving low r value of the devalued consumption rules are never experienced directly after the action upon the manipulanda, the r value cannot be backpropagated by the reinforcement learning component. This is the point where ACS needs to use its representation of the environment in order to realize that the one action in the first step is superior to the other because it predicts the consequent eating of the better reinforcer. In this section we present two different processes which show that the ACS is able to use the evolved representation. A direct comparison to the results of the rat experiments will provide indications that the artificial and real cognitive mechanisms resemble each other.

The One-Step Mental Acting Algorithm The first algorithm is derived from Dyna-Q (Sutton, 1991). The Dyna approach forms an environmental representation similar to the ACS. However, it does not generalize any rules but simply stores each S-R-E relation in a table. In addition to the reinforcement learning (Q-learning) and model building, the Dyna-Q algorithm learns from the execution of hypothetical actions which is called the *random-sample one-step tabular Q-learning*:

1 Choose a previously observed state s
  and an action a at random
2 Predict the next state s' and the reward r
  according to the internal world model
3 Apply Q-learning to s, a, s', r (i.e.
  $Q(s,a) = Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)]$)

Different from the Dyna approach, the ACS does not have any concrete situation-action relations available in its memory since it represents the environment in a generalized form. Thus, it is not directly possible to choose a concrete previously observed situation and action but it is only possible to choose a classifier that represents a somewhat more generalized condition-action relation. Also, the classifiers do not represent a specific payoff prediction. Whereas Dyna models the payoff resulting from an action execution in each situation directly, the reward measure in the classifiers in ACS predicts a combination of immediate payoff and the maximum of the consequent reward measures r of the matching classifiers in the resulting state, which is actually more similar to the policy measure $Q(s,a)$ in Dyna. Moreover, the specified effect is generalized as well, so that it is not possible to specify the resulting situation completely. Thus, it needs to be assured that the assumed relation between the chosen classifier and a consequent classifier is reliable. Our *One-Step Mental Acting* algorithm works as follows:

1 Choose a reliable classifier cl
  from the population at random
2 Form a linked set L_{set} of classifiers whose
  conditions at least specify the changing
  attributes in cl and match the
  anticipation of cl
3 Backpropagate the reward r of the classifier
  with the best action to classifier cl (i.e.
  $r = r + b_r * (\gamma * max_{cl \in L_{set}}(q_{cl} * r_{cl}) - r)$)

Line 1 in the algorithm selects a reliable condition-action relation that is going to be updated. The choice of only reliable classifiers ($q > \theta_r$) assures that the anticipations of the chosen classifier are accurate. The conditions in line 2 assure that the classifiers in the consequent set are related to cl by demanding that they rely on the attributes that cl predicts to change. Finally, line 3 executes the reward backpropagation from this set to the selected reliable classifier.

Despite the demanded tight linking of classifiers in line 2, a closer look at the classifier list revealed that the algorithm occasionally distributed reward across the barrier of one trial. This is a problem, since, as mentioned above, the ACS does not predict the immediate resulting payoff from the environment. Thus, applying this algorithm, the reward prediction of a classifier that normally receives a high direct payoff from the environment will decrease and one that receives a low direct payoff will increase. In our problem, however, the cross-trial propagation cannot be prevented since the chosen coding results in a final state that is identical with one of the starting states.

Figure 5 shows the behavior of the ACS when it additionally executes hypothetical actions. If not stated differently, the devaluation phase took place 100 times and two mental actions were executed after each trial. The figure shows that the ACS is able to use its environmental representation to find connections between classifiers in the population and is thus able to backpropagate reward without experiencing the relation directly. The comparison with the rat results, which are averaged over all results published by Colwill and Rescorla (1985), shows that the ACS behaves very much like the rats. Like the rat curve, all ACS curves show a decrease in the execution of the better action. When increasing the devaluation phase to 200 steps, the ACS becomes even more certain about the relation and executes the presumably better action more often. This also happens when the number of mental actions is increased.

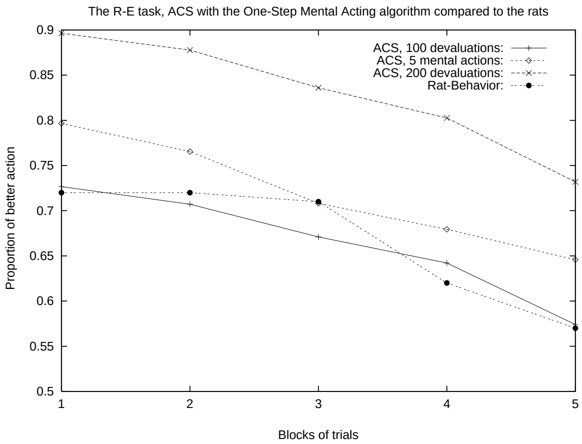

Figure 5: The ACS with One-Step Mental Acting algorithm is able to mentally propagate reward according to the relevant R-E relation and consequently executes the presumably better action in the test phase more often.

The Lookahead-Winner Algorithm The second algorithm is derived from the ideas of a tag-mediated lookahead (Holland, 1990) and the accordant implementation in the CFSC2 (Riolo, 1991). The

ACS proved its capability of lookahead-planning too (Stolzmann and Butz, 2000), but lookahead was not yet combined with the reward-learning process. We combine the lookahead capability with the reward-prediction capability and thus use an enhanced action selection method which takes the best reward prediction in the expected resulting situation into account. The *Lookahead-Winner* algorithm works as follows:

1 Generate current prediction array P
2 For each possible action a
3   Predict the next state s' with the classifier cl_a that has the highest quality q_{cl_a}
4   If a change is predicted
5   Modify the corresponding prediction value P_a according to the best reliable classifier cl' that matches s' (i.e. $P_a = (P_a + r_{cl'} * q_{cl_a} * \gamma)/(1 + q_{cl_a} * \gamma)$)
6 Choose an action by roulette wheel selection using the resulting prediction array

After the generation of the prediction array (introduced in section 2.1.3) in line 1, the algorithm looks one step ahead for each possible action by predicting the resulting situation with the classifier that is most reliable upon the classifiers in the corresponding (implicit) action set (line 3). Only when a change is predicted by this classifier (line 4) does it make sense to take the best reward in the resulting state into account (line 5). Finally, in line 6 an action is chosen out of this modified prediction array using a roulette wheel selection as before.

In contrast to the mental approach in the first algorithm, the Lookahead-Winner algorithm actively takes place before the execution of an action. Problematic in this algorithm again is the division of simulations into trials. Since the ACS does not 'know' about that, the Lookahead-Winner algorithm can mislead the system in states near to the goal state. For now we implemented a one-step Lookahead-Winner version. Deeper versions are certainly imaginable but then the exponential increase in complexity should be considered.

Figure 6 shows the behavior of the ACS when it uses the Lookahead-Winner algorithm. The figure shows that the algorithm anticipates the next situation correctly and thus prefers to execute the action that previously lead to the food that was not devalued in phase 2. The ACS again closely approximates the behavior of the rats. When the devaluation phase is extended to 200 steps, the behavior again changes in favor of the presumably better action. However, the change is not as large as in the one-step mental acting algorithm.

One problem encountered is rather an artifact of the coding of the experiment. In the test phase, unlike the learning phase, both manipulanda are presented. Thus, it is possible that after the execution of an action, again

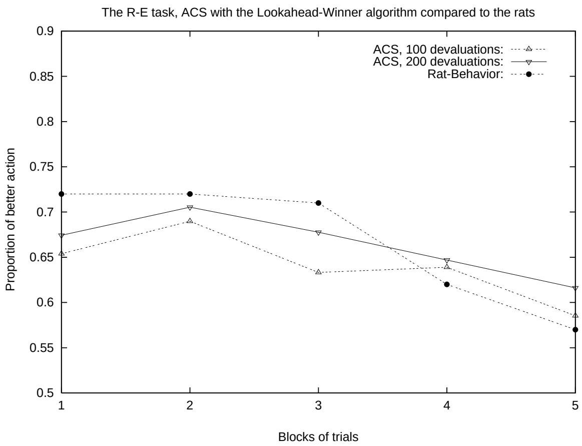

Figure 6: Applying the Lookahead-Winner algorithm, the ACS anticipates the relevant *R-E* relation. Thus, it executes the presumably better action in the test phase more often.

a reliable action classifier matches. Since this classifier predicts a better reward than the devalued eating action, the reward of this classifier is considered. After a couple of executions of both actions upon the manipulanda the action classifiers become unreliable and consequently the reward prediction of the appropriate eating rule is considered. This explains the increase in the execution of the better action from block one to block two. This effect could not be observed in the mental acting algorithm because of the demanded tight linkage between the related classifiers.

3.2 The Stimulus-Response-Effect (S-R-E) Learning Task

In the last section, we introduced the One-Step Mental Acting algorithm and the Lookahead-Winner algorithm, applying them to the *R-E* learning task. In this section we consider another more challenging task (Colwill and Rescorla, 1990). Before, the responses needed to be connected to the occurring effect. Now, different presented stimuli need to be connected to the evolving *R-E* relations as well. The question in the rat experiments was whether or not rats are able to mentally reinforce *S-R-E* relations. Here we ask if the ACS is able to reinforce such relations without actually executing them. Figure 7 illustrates the essentials of the experiment.

Training	Devaluation	Test
$S_1 : R_1 \to E_1, R_2 \to E_2$	$E_1 \to LiCl$	$S_1 : R_1$ vs. R_2
$S_2 : R_1 \to E_2, R_2 \to E_1$		$S_2 : R_1$ vs. R_2

Figure 7: In the *S-R-E* task different stimuli must be considered in addition to the *R-E* relations.

3.2.1 Details of the Rat Experiment

16 rats were tested in this experiment. The methodology was virtually the same as the rat experiments introduced earlier. The main difference was that a discriminative stimulus was present in each trial (i.e. S_1 = light and S_2 = noise). Furthermore, the training phase was altered such that during the first 64 trials (four sessions with 16 trials each) only one manipulandum was available. After that, both manipulanda were present for another 176 trials (eleven sessions with 16 trials each). Also, there was an inter-trial interval (ITI) between the presentation of light or noise where no stimulus was presented and the actions did not lead to food. For further details please refer to the cited paper.

The published results in Figure 8 (Colwill and Rescorla, 1990) show that the rats learned the *S-R-E* relation. They considered the actual situation, and executed the action which leads to the devalued reinforcer significantly less often.

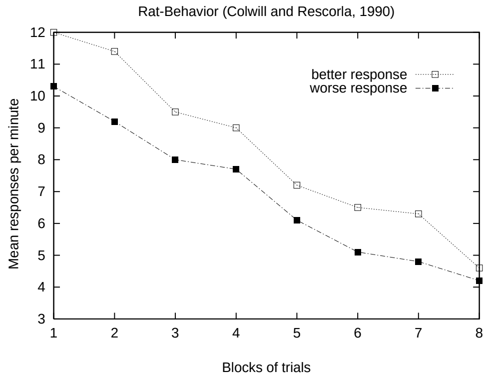

Figure 8: The behavior of the rats in the test phase of the *S-R-E* task shows that the rats are able to learn the relations.

3.2.2 The S-R-E Simulation

The situations in this experiment are coded identically to the first simulation but with two additional attributes. They indicate whether one or the other stimulus is present. Thus, the state $\sigma = 000110$ indicates that there is no food available right now (the first two bits), that the lever is not but the chain is present (the third and fourth bit), and that an additional light stimulus is perceived but noise is not (the fifth and sixth bit). The ITI was presented in 20% of the trials.

As in the rat experiment, we divided the first phase into two parts. In the first part, which included 64 trials, only one of the manipulanda was present in each trial. During the next 176 trials (i.e part two) both manipulanda were present. Next, the second phase took place for 100 trials (unless stated otherwise) where situations were presented that indicate the presence of one or the other type of food and the absence of both, manipulanda and stimuli. As before, in this phase, $\rho = 100$ when the good reinforcer was consumed, and $\rho = 0$ when the devalued reinforcer was consumed. The test phase was executed for 80 trials in order to average 10 steps in each block and compare the eight blocks with the results in the rat experiment. No ITI was presented in the test phase and no action lead to any effect. All results are again averaged over 100 runs.

According to the experiment, (*R-E*) relations are necessary at first which then need to be further differentiated. This is realized in the ACS. At first, it generates a classifier out of the most general one which anticipates the emergence of one type of food. Later, it realizes that the classifier predicts the emergence of the correct food (i.e. the correct bit) in only 50% of the cases. Thus, it remembers the unsuccessful state (with the wrong additional stimulus) in the mark. Back in the successful state, it generates another more specific classifier. This new classifier distinguishes between the two stimuli and therefore always works correct.

In Figure 9 we can observe that the One-Step Mental Acting algorithm is able to reinforce such *S-R-E* connections mentally. A closer look at the population revealed that the algorithm again related classifiers over the trial barrier. Despite that, it is able to learn the distinction mentally and act similarly to the rats. As before, the preference of the presumably better action increases when the devaluation phase is extended. However, when the number of executions of the one-step mental acting algorithm is increased, the behavior approximately stays the same.

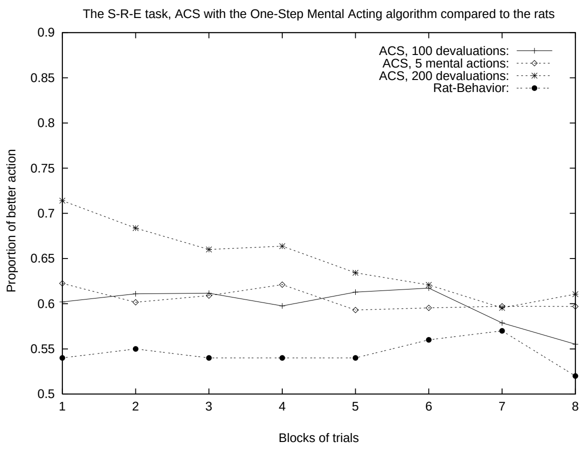

Figure 9: Due to the mental reward propagation, the ACS with One-Step Mental Acting algorithm prefers to execute the action in the test phase which lead in phase one to the in phase two not devalued reinforcer respecting the stimulus.

The Lookahead-Winner algorithm is able to solve the task as well (Figure 10). An increase of the number of devaluations does affect the behavior less than the One-Step Mental Acting one. Moreover, the proportion of the better action is higher which indicates that this algorithm does make fewer mistakes in its predictions. This can be explained by the trial argument. The One-Step Mental Acting algorithm again backpropagated reward over the trial barrier. Thus, it misleads the internal reward predictions. Since the Lookahead-Winner algorithm does not change reward prediction values in the classifiers, wrong combinations can affect the current performance but not the internal representation as a whole. During the test phase the Lookahead-Winner algorithm considers the eating effects correctly. Therefore, the trial problem does not occur when using this algorithm.

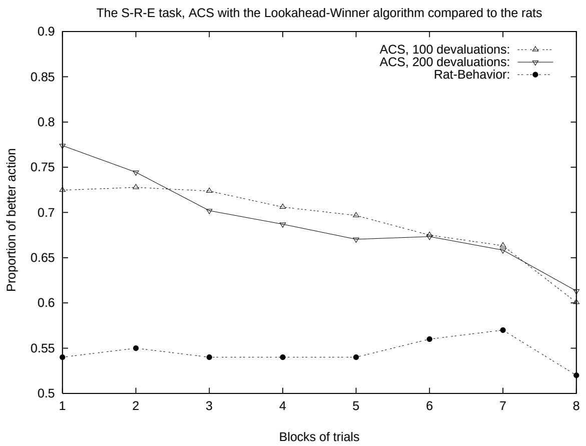

Figure 10: The ACS with Lookahead-Winner algorithm anticipates the correct *S-R-E* relation and consequently executes the presumably better action with respect to the stimulus in the test phase more often.

3.3 More Behavioral Similarities

Apart from the illustrated similar behaviors, three more characteristics were reported in both rat experiments. (1) The rats decreased the actions upon the manipulanda during the test phase. (2) After the test phase the devalued reinforcer was presented to the rats and almost none of the rats consumed it. (3) A preference of sucrose was observable.

In order to simulate the first two points we provided another action possibility (i.e. a do nothing action) that never had any effect. Monitoring the executions in the test phase in this setting revealed the tendency of the ACS to increasingly execute the do nothing action much like the rats. However, the amount of executing this action depended strongly on the reward scaling, and on the base level of executing an action that predicts no effect. If an action has no effect, which is handled by the *useless case*, then the quality of the classifier is simply decreased. Thus, the execution of such an action completely depends on the base level of executing a not presented action in a match set. Similarly, we were able to determine that the ACS would not execute the consumption of the bad reinforcer when the base level is set sufficiently high.

In order to simulate a preference of sucrose, we need to give different reward levels to different reinforcers. However, the rats certainly distinguish internally the different reward levels and consequently a distinction between internal and external reward would be more appropriate. The internal reward measure could be altered over time thus representing the internal emergence of a certain need (e.g. eating: earlier evolved eating rules would predict more and more internal reward). Doing that, the possibility of an internal reward distinction evolves. Defined needs could have different increase times, be more or less specific, and reach different necessity levels. Holland and Reitman (1978) outlined these thoughts and introduced a similar implementation in a simpler framework.

4. Discussion

In the introduction we asked if cognitive processes are present in the ACS. The careful analysis of the ALP gave evidence that the ACS is a learning system with cognitive abilities. Every part of the cognitive theory of anticipatory behavioral control finds its counterpart in the ACS. Like rats, it is able to distinguish between different reaction-effect (*R-E*) relations and, furthermore, relate them to different stimuli. Many similarities between the observed behavior in the ACS and the rats gave evidence that the ACS can indeed behave alike animals. First of all, the ACS preferred the presumably better action approximately as often. Also, since no action lead to any effect in the test phase, the level of executing the better action decreased. Moreover, it was possible to show that the overall execution of actions upon the manipulanda in the test phase decreased as well. Finally, the consumption of the devalued reinforcer was suppressed.

The two introduced algorithms are directly comparable to cognitive processes as well. Both algorithms showed that the ACS is able to use its evolving internal environmental representation to distribute reward. This mental distribution of reward enables the system to adapt to changes in the environment.

The *One-Step Mental Acting algorithm* executes hypothetical actions and thus acts or reasons mentally. This is similar to any mind game we play with ourselves. The mind game could range from directed thoughts (e.g. problem solving) to completely non-directed ones (e.g. dreaming). Right now, the One-Step Mental Acting algorithm is very tight (i.e. it believes only in connections between rules when the first rule directly induces the other rule). When actually developing a completely autonomous system, which does not need to solve tasks all the time, processes similar to dreaming could be imaginable. Rules could be paired with other rules that are linked not as tightly as in the current algorithm. This would result in a sort of mental exploration of improbable thoughts. The more improbable connections are considered the more the system drifts into a dream.

In contrast to the One-Step Mental Acting algorithm, the *Lookahead-Winner* algorithm does not alter the internal reward predictions. It rather resembles a kind of deeper consideration before executing an action. In many situations it is not necessary to react immediately, and in this case higher mammals certainly take the believed consequences of an action into account. Thus, the more time to react, the more successive states should be considered by a system using lookahead.

The paper showed that the ACS is indeed a cognitive system, one which behaves much like the cognitive system in rats. Furthermore, it revealed that the ACS is able to adapt its behavior through mental simulation. We hope that the psychological perspective on the ACS provided in this paper leads to further thoughts and ideas and encourages the further development of the system.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments and remarks.

The work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-97-1-0050. Research funding for this work was also provided by the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Additionally, research funding was provided by the German Research Foundation DFG. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, the U. S. Army, or the U. S. Government.

References

Butz, M. V., Goldberg, D. E., and Stolzmann, W. (2000). Introducing a genetic generalization pressure to the anticipatory classifier system - Part 1: Theoretical approach. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*. In press.

Colwill, R. M. and Rescorla, R. A. (1985). Postconditioning Devaluation of a Reinforcer Affects Instrumental Learning. *Journal of Experimental Psychology / Animal Behavior Processes*, 11(1):120–132.

Colwill, R. M. and Rescorla, R. A. (1990). Evidence for the hierarchical structure of instrumental learning. *Animal Learning & Behavior*, 18(1):71–82.

Drescher, G. L. (1991). *Made-Up Minds, a constructivist approach to artificial intelligence*. MIT Press, Cambridge, MA.

Hoffmann, J. (1993). *Vorhersage und Erkenntnis [Anticipation and Cognition]*. Hogrefe, Goettingen, Germany.

Holland, J. (1990). Concerning the emergence of tag-mediated lookahead in classifier systems. *Special issue of Physica D*, 42:188–201.

Holland, J. H. and Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. In Waterman, D. A. and Hayes-Roth, F., (Eds.), *Pattern Directed Inference Systems*, pages 313–329. Academic Press, New York.

Lanzi, P. L., Stolzmann, W., and Wilson, S. W., (Eds.) (2000). *Learning Classifier Systems: An Introduction to Contemporary Research, LNAI 1813*. Springer-Verlag, Berlin Heidelberg.

Riolo, R. L. (1991). Lookahead planning and latent learning in a classifier system. In Meyer, J.-A. and Wilson, S. W., (Eds.), *Proceedings of the first international conference on simulation of adaptive behavior*, pages 316–326, Cambridge, MA. MIT Press.

Rosen, R. (1985). *Anticipatory Systems*. Pergamon, Oxford, UK.

Stolzmann, W. (1997). *Antizipative Classifier Systems [Anticipatory Classifier Systems]*. Shaker Verlag, Aachen, Germany.

Stolzmann, W. (2000). An introduction to Anticipatory Classifier Systems. In (Lanzi et al., 2000).

Stolzmann, W. and Butz, M. (2000). Latent learning and action-planning in robots with Anticipatory Classifier Systems. In (Lanzi et al., 2000).

Sutton, R. S. (1991). Reinforcement learning architectures for animats. In Meyer, J.-A. and Wilson, S. W., (Eds.), *Proceedings of the first international conference on simulation of adaptive behavior*, pages 288–296, Cambridge, MA. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tolman, E. C. (1932). *Purposive Behavior in animals and men*. Appleton, New York.

Widrow, B. and Hoff, M. (1960). Adaptive switching circuits. *Western Electronic Show and Convention*, 4:96–104.

Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175.