WL-TM-96-1133

CONVERGENCE BEHAVIOR OF TEMPORAL DIFFERENCE LEARNING

RAJ P. MALHOTRA

MAY 1996

FINAL REPORT

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7623

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
MAY 1996

3. REPORT TYPE AND DATES COVERED
FINAL

4. TITLE AND SUBTITLE
CONVERGENCE BEHAVIOR OF TEMPORAL DIFFERENCE LEARNING

5. FUNDING NUMBERS
C
PE 61102
PR 2304
TA ES
WU 01

6. AUTHOR(S)
RAJ P. MALHOTRA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7623

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7623

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TM-96-1133

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

TEMPORAL DIFFERENCE LEARNING IS AN IMPORTANT CLASS OF INCREMENTAL LEARNING PROCEDURES WHICH LEARM TO PREDICT OUTCOMES OF SEQUENTIAL PROCESSES THROUGH EXPERIENCE. ALTHOUGH THESE ALGORITHMS HAVE BEEN USED IN A VARIETY OF NOTORIOUS INTELLIGENT SYSTEMS SUCH AS SAMUEL'S CHECKER-PLAYER AND TESAURO'S BACKGAMMON PROGRAM, THEIR CONVERGENCE PROPERTIES REMAIN POORLY UNDERSTOOD. THIS PAPER PROVIDES A BRIEF SUMMARY OF THE THEORETICAL BASIS FOR THESE ALGORITHMS AND DOCUMENTS OBSERVED CONVERGENCE PERFORMANCE IN A VARIETY OF EXPERIMENTS. THE IMPLICATIONS OF THESE RESULTS ARE ALSO BRIEFLY DISCUSSED.

14. SUBJECT TERMS

15. NUMBER OF PAGES
11

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
SAR

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. ***(If known)***

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1993-0-336-043

Convergence Behavior of Temporal Difference Learning

Raj P. Malhotra

The University of Dayton, Electrical Engineering Department

Abstract--*Temporal Difference Learning is an important class of incremental learning procedures which learn to predict outcomes of sequential processes through experience. Although these algorithms have been used in a variety of notorious intelligent systems such as Samuel's checker-player and Tesauro's Backgammon program, their convergence properties remain poorly understood. This paper provides a brief summary of the theoretical basis for these algorithms and documents observed convergence performance in a variety of experiments. The implications of these results are also briefly discussed.*

INTRODUCTION

Problems involving sequential processes are encountered in a wide variety of interesting engineering situations. In heuristic search applications one attempts to learn an evaluation function which predicts the utility of searching in a certain region of the search space. Game-like decision processes involve learning to predict outcomes based upon current state. Difficult modeling tasks and pattern recognition problems often involve learning process complexities through observing sequential behavior. The common thread in these involves *learning to predict* process outcomes and sequential behavior. Often, the goal of prediction is not an end in itself, but rather a prerequisite for effective control of some complex, sequential system.

For problems in which apriori knowledge is limited, unreliable, or unavailable, Reinforcement Learning methods ([Klopf],[Sutton],[Watkins]) have received increasing attention as viable control mechanisms which can achieve near-optimal performance. Mathematically, these algorithmic methods may be viewed as a form of Iterative, Stochastic Dynamic Programming in which one attempts to approximate value iteration (or policy iteration) as process characteristics are learned through experience. The hallmark of reinforcement learning methods is their treatment of actual, experienced state transitions and rewards as unbiased estimators of the statistically true values. This provides for incremental calculations which have been shown to theoretically converge in the limit to optimal predictions. An important question which has to do with the nature of the convergence of these algorithms (e.g., convergence requirements and convergence behavior as a function of some appropriate parameter space).

Temporal Difference Learning [Sutton] has received increased attention in recent years. as a type of reinforcement learning which learns to predict outcomes of *markov processes* with delayed and accrued rewards. The significance of this method is that it can learn to address the difficult temporal credit assignment problem in which one must assign credit (or blame) to decisions when the feedback is noisy and delayed. This is believed to describe many real-world situations in which more conventional, supervised learning methods are less efficient (or not viable) because of the reliance on the availability of accurate feedback at each decision point. In the absence of such feedback, the task of calculating optimal actions at each decision point becomes formidable--a requirement for farsighted. predictive considerations emerges. Whereas conventional prediction-learning methods assign credit (value of decisions) by means of the difference between predicted and actual outcomes, Temporal Difference Learning assigns credit by means of the difference between temporally successive predictions. This allows for more efficient learning.

Although Temporal Difference Learning has been used in several famous machine intelligence systems ([Samuel],[Tesauro]), their convergence properties remain poorly understood. In particular, proofs abound which show that Temporal Difference Learning converges to the optimal value function, but only given infinite training time. Of course, in practice one never has infinite training time. Therefore, a more cogent question for the practitioner is "what is the convergence behavior of temporal difference learning when training times are finite?" That is the question to which this paper is dedicated.

We include several considerations in the term "convergence behavior": These include performance metrics such as the rate of the probabilistic convergence of predictions and the root mean-squared error of predictions at deterministic stopping times versus algorithmic, *synthesis parameters* involving learning update rate and an important *eligibility horizon* effecting the spread of credit or blame over the sequence of states. The interactions of these factors with *system parameters* including differing amounts of process noise, state transition and cost/reward variances will be briefly highlighted through simulation results. The intent of this presentation is to document general empirical results and to motivated further (more rigorous) studies. Before this is undertaken, however, a brief theoretical foundation shall be provided.

THEORETICAL FOUNDATIONS

The original motivation for the development of temporal difference learning methods involved the *temporal credit assignment* problem. There, one experiences a series of state transitions which terminate when an *absorbing state* is reached. There is a cost, z, associated with the terminal state; all other states (and transitions) are nominally assumed to have zero cost. The goal of the Temporal Difference Learner is to derive a sequence of predictions $\{P_1,P_2,...,P_n\}$ of the final outcome based upon a sequence of experienced states,$\{x_1,x_2,...\}$ (as shown in figure 1).

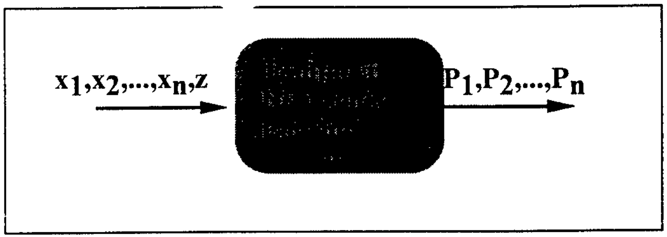

Figure 1 - Conceptual depiction of TDL

The sequence of states is assumed to be a markov process in which state transition probabilities are dependent only upon the most recent state: $p(x_{t+1}|x_t,x_{t-1},x_{t-2},...,x_1) = p(x_{t+1}|x_t)$. The learning agent is parameterized by a vector of weights, w, which control the sequence of outcome predictions based upon state trajectories. The goal is to find the appropriate weight settings so that the prediction values are optimized. The optimal prediction values are known to satisfy the Bellman Equation:

$$P^*_w(x) = \min/\max[R(x) + \gamma \sum_{x'} p(x'|x)P^*_w(x) \qquad \textbf{(1)}$$

where R(x) is the incremental transition cost to leave state x, x' is the state transitioned to, P^*_W is the weight-parameterized optimal prediction value, and γ is a discount factor in (0,1). The learning procedure is expressed as a rule for updating w:

$$w(n+1) = w(n) + \sum_{t=1}^{m}\Delta w(t) \qquad \textbf{(2)}$$

In practice the change in weights,Δw, may be accrued over a complete sequence, until a termination is reached. A typical supervised learning method would associate each state with the final outcome and perform gradient-descent based upon each pair separately:

$$\Delta w(t)= \alpha(z-P_t)\ \{\nabla_w P_t\} \qquad \textbf{(3)}$$

where α is an update rate effecting the rate of convergence and the gradient, $\nabla_w P_t$, is a vector of partial derivatives of P_t with respect to w. The difficulty with this approach is that the error terms, $(z-P_t)$, can only be computed at the end of each sequence, when an outcome is achieved. This is computationally expensive. The temporal difference method is based upon equating the final prediction errors to a string of intermediate prediction errors which can be computed incrementally:

$$z - P_t = \sum_{k=t}^{m} P_{k+1} - P_k \leftrightarrow \quad E(z) \approx P_{k+1} \qquad \textbf{(4)}$$

A key point is that the individual predictions serve as unbiased estimates of the final outcome, z. By substituting equation (4) into equations (3) and (2) we derive a new weight update equation:

$$
\begin{aligned}
w \leftarrow w &+ \sum_{t=1}^{m} \alpha\left(z - P_t\right)\nabla_w P_t \\
= \ w &+ \sum_{t=1}^{m} \alpha\left(\sum_{k=t}^{m}\left(P_{k+1} - P_k\right)\right)\nabla_w P_t \qquad \textbf{(5)} \\
= \ w &+ \sum_{t=1}^{m} \alpha(P_{t+1} - P_t)\sum_{k=1}^{t}\nabla_w P_k
\end{aligned}
$$

where m represents the number of state transitions experienced in a particular Observation-Outcome Sequence (OOS)--this will in general be a random variable. Note the last summation term in (5); this is an *eligibility* term which assigns credit over a string of past states. Finally, note that (5) can be computed incrementally, after each state transition. Weight updates may either be performed after the OOS has completed (*batch* mode) or alternatively, after each individual state transition (*on-line* learning).

The TD(λ) family of learning procedures modifies equation (5) by the inclusion of an exponential weighting factor on the eligibility, λ ε [0,1]. This results in a new weight update procedure in which the incremental weight changes are given by

$$
\Delta w(t) = \alpha\left(P_{t+1} - P_t\right)\sum_{k=1}^{t}\lambda^{t-k}\nabla_w P_t \qquad \textbf{(6)}
$$

Equation (6) may be thought of as an error term for temporal credit assignment coupled with a gradient-descent mechanism for structural credit assignment. The last summation term can be thought of as a state eligibility factor, e_t, given by

$$
\begin{aligned}
e_{t+1} &= \sum_{k=1}^{t+1}\lambda^{t+1-k}\nabla_w P_k = \nabla_w P_{t+1} + \sum_{k=1}^{t}\lambda^{t+1-k}\nabla_w P_k \\
&= \nabla_w P_{t+1} + \lambda e_t \qquad \textbf{(7)}
\end{aligned}
$$

Note that this eligibility factor may also be computed recursively.

Using the equations given above, we can now derive the TD(λ) learning algorithm:

1. Initialize parameters; x, w, λ, α
2. Observe a state transition
3. Compute new prediction, $P_t=f(x_t,w_t)$
4. Compute the new eligibility, e_{t+1}
5. Compute incremental weight update via (6)
6. If x_t is not an absorbing state, Go To step 2

The incremental weight updates computed in step 5 may either be immediately applied to the weights (in the case of on-line learning) or they may be summed and applied to the weights after the completion of an OOS (in the case of batch mode learning).

As mentioned in several recent papers ([Bertsekas], [Jaakkola]), the convergence behavior of TD(λ) remains poorly understood. One way to begin to study the convergence of the predictions (or equivalently, the weights) to the optimal prediction, P*, satisfying equation (1), is to examine factors effecting the convergence through repeated statistical experiments. For a complete and rigorous analysis (a considerable undertaking), one would need to evaluate the performance of TD(λ) over the complete *parameter space* which includes[1]:

- **Synthesis Parameters:** exponential eligibility factor, λ; learning update rate, α; function approximation method for $P_t=f(x_t,w_t)$, and the internal state representation comprising x_t
- **System Parameters:** characteristics of state transition and cost/reward probability density functions (mean, variance, time-dependency, uniformity over the state space, etc.); dimensionality/cardinality of the "true" state space; nature of the markov processes (topology, number of absorbing states, etc.); initial state, etc..

There are several alternatives for studying the convergence behavior of temporal difference learning. In the remainder of this paper we shall document the results of high-level experiments which produce several interesting insights and conjectures.

EXPERIMENTAL RESULTS

In order to establish a top-level assessment of the significance of various factors upon the convergence of TD(λ), several experiments were conducted. These involved learning to predict the outcomes of a markov

[1] these lists are not exhaustive; the determination of the proper parameter space to study convergence over is worthy of extensive research in itself.

process with 9 possible states, $x^1, x^2, ..., x^9$. The state space may be considered as a string in which the end states (x^1, x^9) represent two different outcomes (possibly winning and losing a game). As indicated in figure 2 below, the simulation nominally begins in state 5 and progresses according to the state transition probabilities until a terminal state is reached.

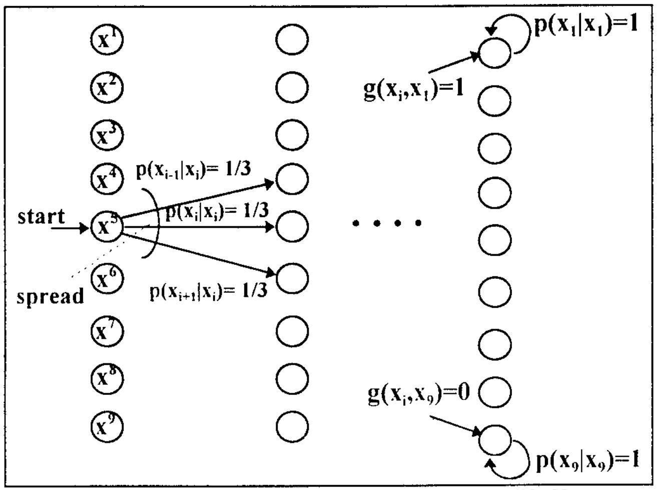

Figure 2 - 9-state, modified Random Walk

This markov process is based upon the simpler random walk process described and studied in [Sutton]. The only important costs associated with the process are those given by the transition cost function going into the terminal states. The concept of "spread" , as illustrated in figure 2, is related to state transition variance.

Experiment 1

The first set of experiments involved learning to predict outcomes for the 9-state modified random walk with a variety of settings for the system parameters, λ and α. Specifically, λ was set at λ=0, λ=.3, λ=.8, and λ=1 while α was allowed to vary over α = {0,.1,.2,.3,.4,.5,.6}. The metric used to judge convergence was root-mean-squared error (RMSE) taken after all learning was completed.

In figure 3 below we can view the average RMSE taken over 10 sets of simulations in which the temporal difference learner operated over 2000 state transitions. Note that the use of a fixed number of state transitions, rather than a fixed number of sequences (OOSs) allows for a more even comparison (since OOSs will have varying lengths). It is particularly interesting to note the erratic behavior of the TD(λ=1) curve. This curve indicates that for the 9-state, modified random walk example, TD(λ=1) is much more sensitive to variations in α than are the other λ curves.

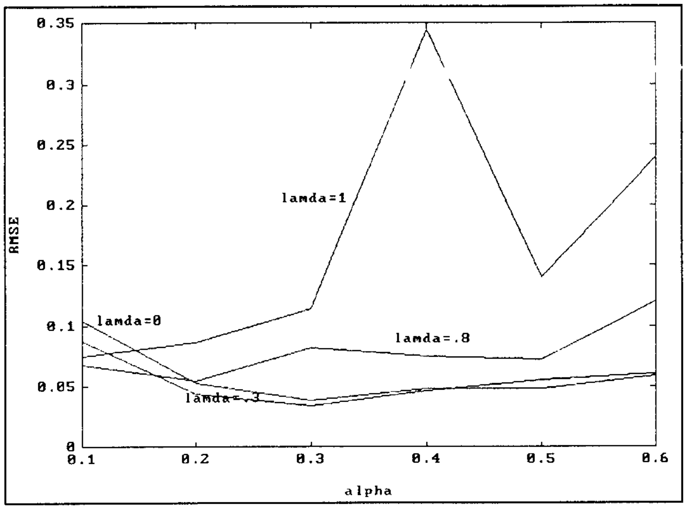

Figure 3 - Results for Experiment 1

Experiment 2

A more interesting experiment entails observing changes in convergence behavior for cases where there is more (or less) uncertainty in state transitions. In particular we will examine cases where the spread (as indicated in figure 2; spread is related to state transition variance) is set to 3, 5, and 7. For each of these values the probability density is evenly spread among the possible transitions (as indicated in figure 2 for the spread=3 case). In figure 4 we plot the RMSE values averaged over all λ (λ=0,.3,.8,1).

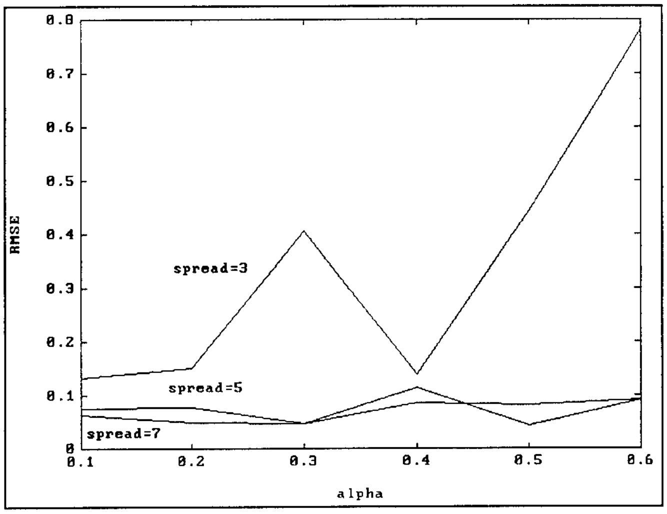

Figure 4 - RMSE for various "spreads"

It is interesting to note the improvement in going from a spread of 3 to a spread of 5. The RMSE values do not improve significantly when going to spread=7. This seems to indicate that an increased variance in state transition offers some benefit; the process becomes more volatile, and singular state transitions contain potentially more information. However, this benefit wanes as more volatility is introduced.

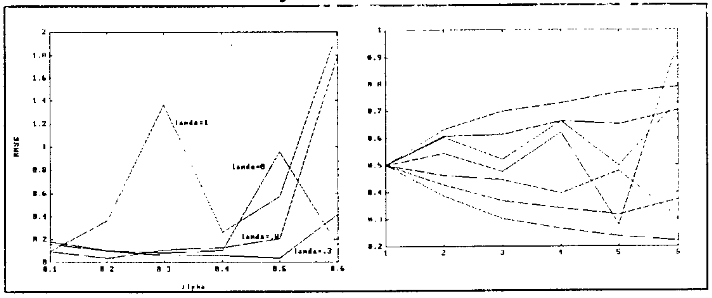

Fig. 5a - RMSE, weight record curves for spread=3

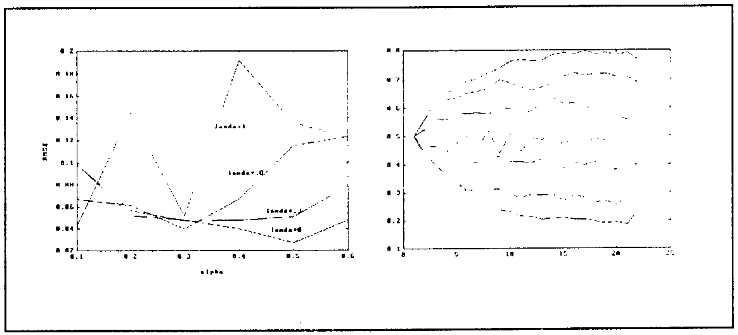

Fig. 5b - RMSE, weight record curves for spread=5

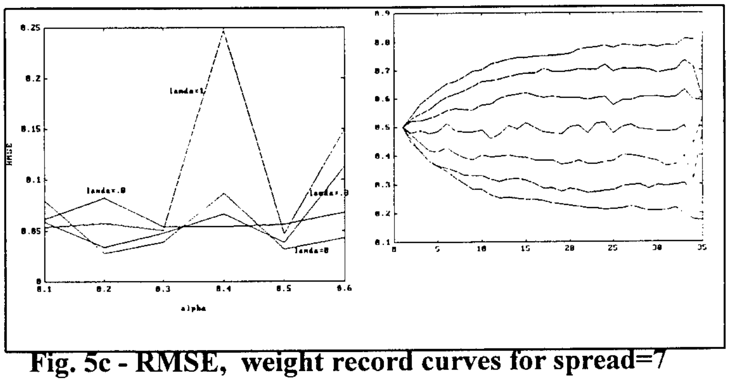

Fig. 5c - RMSE, weight record curves for spread=7

Experiment 3

A third experiment was run in which the learner was allowed to partially converge for a given state transition probability set. Then the transition probabilities were changed and it was observed how well each of the learners could recover for different setting on α and λ. The result was somewhat surprising and almost counter-intuitive. As shown in figure 6 learning with large λ values was superior after the sudden change in transition probabilities. These seems counter-intuitive, as one would expect that updating weights over a shorter horizon would be superior. This is an interesting matter for further study.

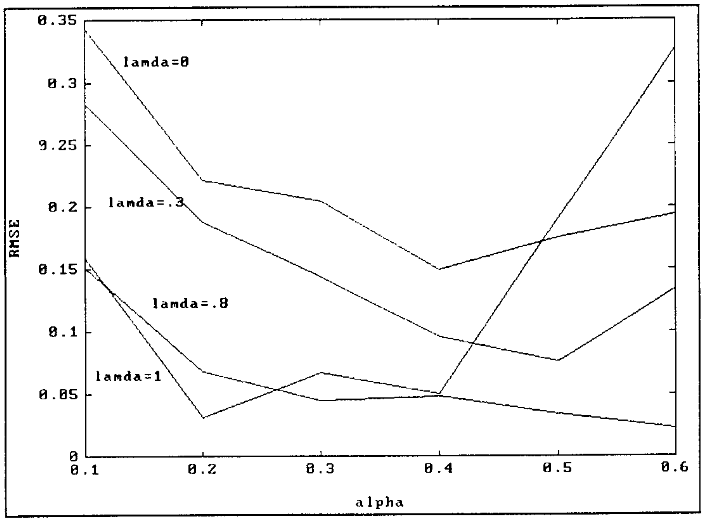

Figure 6 - Results from Experiment 3

Experiment 4

A final set of experiments was performed in which the learner was allowed to start in different states. Not surprisingly, convergence was sensitive to the starting state. In figure 7 below, the weights for lower states (physically, on the top of the graph), x_1-x_3, learn less rapidly than those for the higher-numbered states. This is because the process is started in state x_7 and given the nature of the state transition probabilities (uniformly distributed, see fig. 2), the lower-numbered states will occur less frequently.

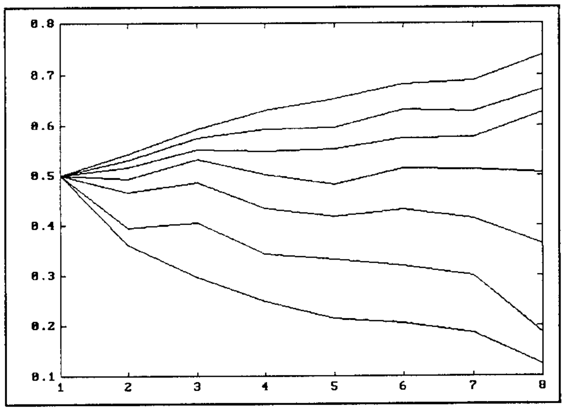

Figure 7 - Weight changes (exp. 4)

As shown in figure 8, the RMSE values are drastically different for this experiment than for some of the earlier ones. This would indicate that there is also some interaction between λ and the starting state.

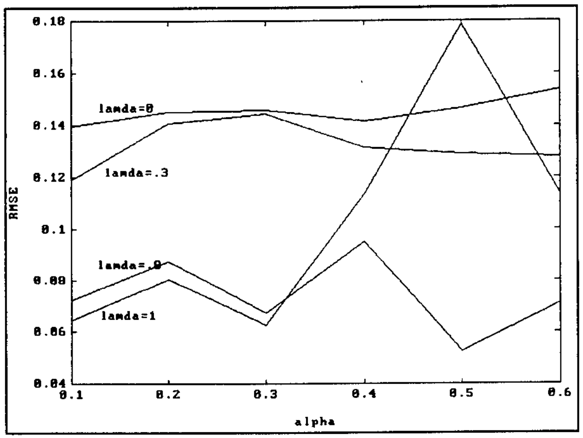

Figure 8 - RMSE for experiment 4

CONCLUSION

In this paper we have briefly discussed some preliminary considerations for what may be a fruitful new thrust in the study of reinforcement learning algorithms. We have argued that convergence properties are not well understood and that they warrant further investigations. Several simple experiments were performed and documented in order to get a first look at the behavior of these important learning algorithms.

REFERENCES

[SUTTON] R.S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, pp. 9-44, 1988

[WATKINS] Watkins, C.J.C.H., "Learning from Delayed Rewards", Ph.D. Thesis, Cambridge University, England, 1989

[BERTSEKAS] Bertsekas, Dimitri P., "A Counterexample to Temporal Difference Learning", *Neural Computation,* vol. 7, pp 270-279 1995

[JAAKKOLA] Jaakkola, T., M. Jordan, S. Singh, "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms", *Neural Computation*, vol. 6, pp1185-1201, 1994

[TESAURO] Tesauro, G., "Practical Issues in Temporal Difference Learning", *Machine Learning,* vol.8, pp257-277, 1992

[KLOPF] Klopf, A. H., "A Neuronal Model of Classical Conditioning", *Psychobiology,* vol.16, pp85-125 1988