The Fighter Aircraft LCS: A Case of Different LCS Goals and Techniques

R. E. Smith
The Intelligent Computing Systems Centre
The University of The West of England
Bristol, UK
email: robert.smith@uwe.ac.uk

B. A. Dike
The Boeing Company
St. Louis, Missouri
email: bruce.a.dike@boeing.com

B. Ravichandran
A. El-Fallah
R. K. Mehra
Scientific Systems
Woburn, MA
Email: ravi, adel, rkm@ssci.com

Abstract

There are a number of common difficulties and open issues that pertain to the "traditional" LCS model. Many of these topics were central at The First International Workshop on Learning Classifier Systems (Houston, Texas, 1992). Since the first workshop, several significant, theoretically-supported advances in LCS practice have addressed these issues. However, a system employed by the authors to acquire novel fighter aircraft maneuvers from combat simulation is more akin to the traditional LCS model than to more recent systems.. Given the difficulties often experienced in LCS research on simple problems, one must ask how a relatively primitive LCS has had consistent success in the complex domain of fighter aircraft maneuvering?

This paper overviews the troublesome issues discussed at the first workshop, and recent advances. It then presents the fighter aircraft LCS, in greater detail than in previous publications. Positive results from the system are discussed. The paper then focuses on the primary reasons the fighter aircraft LCS has avoided the difficulties of the traditional LCS. The authors believe the system's success has three primary origins: differences in credit assignment, differences in action encoding, and (possibly most importantly) a difference in system goals. In the fighter aircraft system, the goal has been simply the discovery of *innovative, novel* tactics, rather than online *control*.

The paper concludes by discussing the most salient features of the fighter aircraft learning system, and how those features may be profitably combined with other LCS developments.

1 QUESTIONS FROM THE FIRST LCS WORKSHOP

At The First International Workshop on Learning Classifier Systems, several open-ended brainstorming sessions were conducted. The attendees were a self-selected group of international experts in the field. Key questions from an early brainstorming session included:

- What features distinguish LCSs from other systems?
- What niche in machine learning are LCSs *intended* to fill? *Best able* to fill?
- Is the best role for LCSs *descriptive* (scientific modeling) or *prescriptive* (problem solving)?
- How does one scale LCSs up to real-world problems?
- How does one evaluate the results of LCS experiments?

It is interesting that, despite the gathered expertise, the attendees of the first workshop had fundamental questions about what an LCS was, and what an LCS was for! Workshop attendees felt that this was a serious concern for LCS research, since lack of defining issues restricts communication and focus in the LCS research community. Therefore, providing, clarifying, and classifying the framing issues became a primary concern in workshop discussions.

2 ISSUES AND PROBLEMS WITH LCSS

In a paper presented at the first LCS workshop, Booker (1992) points out an important part of clarifying the details and concerns of LCS research. I paraphrase his comments here. The LCS is usually described as a *method*: a set of algorithmic details that define how a way to solve a class of problems. However, in many ways the LCS is more of an *approach*: a set of conceptual details that define a certain direction for developing methods. Therefore, the defining issues for the LCS are not necessarily algorithmic, but conceptual. The central problem addressed by workshop's discussions was to clarify these defining, conceptual issues. The workshops conclusions on these defining issues emerged from a "reverse engineering" process: discussion focused on central technical issues that arise in LCS research, and outlining these issues helped to categorize key LCS concepts. Technical questions included:

- What methods can create cooperation in an LCS, despite the competitive nature of LCS and GA mechanics (e.g., covering a set of possible inputs;

performing a tiling that yields an effective input output mapping; forming a set of structures that combine to form a action; forming a parsimonious rule set through default hierarchies; forming sequences of actions through rule chains; etc.)?

- How can one insure effective credit assignment in an LCS?
- How can the LCS create computationally useful internal message processing in an LCS (to allow LCSs to deal with *non-Markovian* environments)?
- What is an appropriate syntax or representation in an LCS?

Behind these questions lay a number of typical difficulties that occur in practice with LCSs. These difficulties include:

- Classifier populations can sometimes be "taken over" by rules that have high strength (i.e., high expected payoff), despite the fact that they are supplanting other (lower strength) rules upon which overall system performance depends.
- This problem can be coupled to difficulties in appropriate credit assignment. LCSs often have both complex *structural* credit assignment problems (i.e., balancing credit in default hierarchies) and *temporal* credit assignment problems (i.e., how should credit be propagated through rule chains).
- As a result of structural credit assignment problems (and possible LCS syntactical difficulties) over-general rules often take over populations inappropriately.
- Attempts at the use of internal messages for non-Markovian problems seems to simply aggravate these problems, and lead to problems of *parasite rules.* There are rules that benefit from indirect payoff through temporal credit assignment, yet cause the eventual degradation of system performance.

A common outcome of all of these problems is a "rise and crash" behavior in LCSs. LCSs will often evolve useful, high performance rule sets, then lose key rules through inappropriate takeover of other rules. Sometimes the system will re-evolve useful rule sets, and repeat the pattern.

3 RECENT ADVANCES

The observed problems above have led to a number of modifications to the traditional LCS. Many of these are best illustrated by Wilson's XCS (1995). In XCS, the goal of genetic learning is more strictly clarified by a framework of reinforcement learning (Sutton & Barto, 1998). Specifically, consider the mapping from state/action pairs to payoff, $(s,a) \rightarrow P$ (where payoff P includes both reward r and appropriate temporal credit assignment). XCS focuses genetic learning on acquiring accurate generalizations over this "payoff landscape". In XCS this is accomplished by coupling genetic learning to Q-learning, a standard reinforcement learning technique (Watkins, 1989). Given an accurate, compressed representation of the payoff landscape, and the (approximate) assurance of correct Q values, one can follow a greedy strategy with respect to these values, and be (approximately) assured of Bellman-optimal reinforcement learning control.

Given this clarification of the goal of the "LCS approach" within XCS, an appropriate set of techniques are adopted. Chief amongst these are:

- the use of an *accuracy* measure for classifier fitness, rather than the traditional *strength* measure. While strength is an estimator of the expected payoff for a classifier, accuracy is related to the variance of classifier's payoff. Low variance (high accuracy) classifiers are considered to be good generalizations over the payoff landscape.
- The use of a non-panmictic GA (that is, a GA that *does not* act across the entire classifier population). In early versions of XCS, the GA was applied only to classifiers in the *match set* (those that match the current environmental message). In later versions the GA was applied only to classifiers in the action set (those that match the current environmental message *and* suggest the same action).

In XCS, the effects of these two techniques act in balance to form maximally general, but simultaneously maximally accurate classifiers.

There are several relevant observations to made from XCS in light of the outcomes of the first workshop.

In XCS (and other, more recent LCS efforts) the nature and goals of the LCS approach are clarified in the context of reinforcement learning control. This addresses most of the questions discussed in Section 1.

In relationship to the questions in Section 2, issues are clarified as well. Appropriate credit assignment is related to the reinforcement learning control framework. The nature of appropriate cooperation of classifiers is also clarified in two senses: cooperation to form accurate generalizations over the payoff landscape, cooperation through time, via the Bellman optimality principles of reinforcement learning techniques (i.e., Q-learning). The other LCS questions in Section 2 (non-Markovian environments and alternate classifier syntax) are also being addressed within the framework of XCS (Wilson, 1999).

XCS clarifies many of the questions associated with the LCS in terms of reinforcement learning control. However, one should also consider how these questions can be clarified in other contexts. The following sections consider another successful LCS, and it's implications for future LCS research.

4 ANOTHER LCS

The authors have extensive, ongoing experience with using LCSs for acquiring novel fighter aircraft maneuvers. This is a successful, ongoing project, satisfying real-world goals for industry, NASA, and The United States Air Force (Smith & Dike, 1995; Smith et al., in press). Despite the extensive questions about the traditional LCS model expressed at the first workshop, this project has used a system that is quite similar to the traditional model. It is important to note that this system is an outgrowth of work that began in 1992, and many of its features were designed incrementally, based on experience, for a particular task. The system is likely to benefit from many of the advances in LCS theory and practice discussed in Section 3 (this is a part of our ongoing work). However, given the current system's success, the goals and techniques of this system merit further consideration.

By way of introduction, consider the basic problem of one-versus-one fighter aircraft combat. Two aircraft start their engagement at some initial configuration and velocity in space. In our simulations, an engagement lasts for a pre-specified amount of time (typically 30 seconds). An engagement is divided into discrete time instants (in our simulation, 1/10th second instants). At each instant, each "aircraft" must observe the state of the environment, and make a decision as its own action for that instant. A score for a given aircraft can be calculated at the end of an engagement by comparing that aircraft's probability of damaging its opponent to its own probability of being damaged.

Given this basic outline of the problem at hand, we will introduce details of the fighter aircraft LCS.

4.1 DETECTORS AND CLASSIFIER CONDITIONS

The conditions of the classifier are from the traditional {1, 0, #} alphabet, defined over the encoding of the environmental state shown in Table .

	Bins (represented in binary)							
3 bits:								
Own Aspect Angle	< 45	45 to 90	90 to .35	135 to 180	180 to 215	215 to 270	270 to 315	315 to 360
Opp. Aspect Angle	< 45	45 to 90	90 to 135	135 to 180	180 to 215	215 to 270	270 to 315	315 to 360
2 bits:								
Range (1000 feet)	< 1	1 to 4.5	4.5 to 7.5	> 7.5				
Speed (100 knots)	< 2	2 to 3.5	3.5 to 4.8	> 4.8				
Delta Speed (100 knots)	< -.5	-.5 to .5	.5 to 1	> 1				
Alt. (1000 feet)	< 10	10 to 20	20 to 30	> 30				
Delta Alt. (1000 feet)	< -2	-2 to 2	2 to 4	> 4				
Climb Angle	< -30	-30 to 30	30 to 60	> 60				
Opp. Climb Angle	< -30	-30 to 30	30 to 60	> 60				
Total: 20 bits								

Table 1: Encoding of Environmental State in the Fighter Aircraft LCS. "Opp." Means opponent, and aspect angles are in degrees.

Note that although this encoding is *crisp*, its coarseness, and the open-ended bins sometimes used in the encoding, have a linguistic character that is similar to that of a *fuzzy* encoding. That is, concepts like "high, low, and medium" are embodied in the encoding.

4.2 MATCH-AND-ACT

In our system if no classifiers are matched by the current message, a default action for straight, level flight is used. There is no "cover" operator.

4.3 GA ACTIVATION

- The GA acts at the end of each 30 second engagement.
- The GA is panmictic (it acts over the entire population).
- In some of our experiments, the *entire* classifier list is replaced each time the GA is applied. This has been surprisingly successful, despite the expected disruption of the classifier list. In recent experiments, we have used a generation gap of 0.5 (replacing half of the classifier population with the GA). This is still a substantially larger portion than are replaced in many LCSs.
- A new, GA-created classifier is assigned a fitness that is the average of the fitness values of its "parent" classifiers.
- The GA used employed tournament selection, with a tournament size ranging from 2 to 8.
- Typical GA parameters are a crossover probability of 0.95, and a mutation rate of 0.02 per bit position. When a condition bit is selected for mutation, it is set to one of the three possible character values is selected (1, 0, or #) with equal probability. Note that this actually yields a effective mutation probability of (0.02)(2/3)=0.0133. Children rules replaced randomly selected rules in the population.

4.4 CONFLICT RESOLUTION

The matching rule with the highest fitness/strength is selected to act *deterministically*.

4.5 CREDIT ALLOCATION

- At the end of an engagement, the score for the complete engagement is calculated.
- This score is *assigned* as the fitness for *every* classifier that acted during the engagement (and to any duplicates of these classifiers).
- Note that this score *replaces* the score given by averaging the parent scores when the rule was generated by the GA.

4.6 EFFECTORS AND CLASSIFIER ACTIONS

- The classifier actions directly fire effectors (there are no internal messages).
- Actions are from the traditional {1,0,#} syntax, and correspond to the coarse encoding shown in Table 1.

	Action Values (suggested to AASPEM)							
3 bits:								
RBA	0	30	45	90	180	-30	-45	-90
AOA	0	10	20	30	40	50	60	70
2 bits:								
Speed (100 Knots)	1	2	3.5	4.8				
Total: 8 bits								

Table 1 Action Encoding in the Fighter aircraft LCS. RBA is Relative Bank Angle (in degrees), and AOA is Angle of Attack (in degrees).

5 RESULTS

Results from the system outlined in Section 4 are extensively documented elsewhere (Smith & Dike, 1995; Smith et al., in press). The system has found a variety of novel fighter aircraft maneuvers that were positively evaluated by actual fighter test pilots. Moreover, the system discovers maneuvers from a variety of well recognized fighter aircraft strategies (Shaw, 1988). In initial experiments, only one aircraft learned under the action of the LCS, while the other aircraft flew a simple, fixed, but reactive, fighter aircraft logic. In more recent work, both aircraft are learning under the action of LCS. These efforts are showing useful, coevolved strategies. Typical results are shown in Figure 1 and Figure 2.

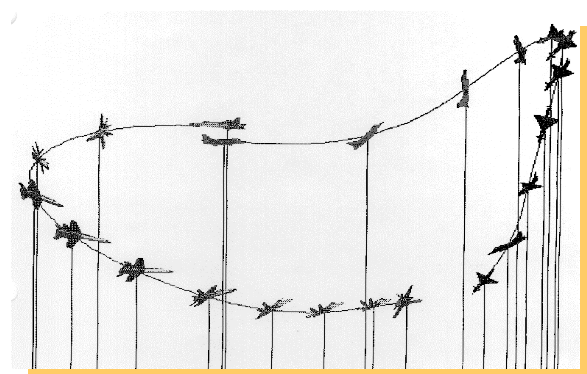

Figure 1 A maneuver evolved by the LCS. The aircraft on the left is following a fixed, but reactive strategy, while the aircraft on the right is following a strategy evolved by the LCS. (Smith & Dike, 1995)

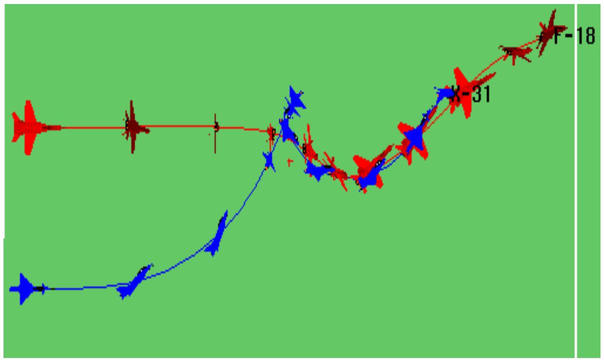

Figure 2 A coevolved result, where both aircraft are executing maneuvers that were discovered by the separate LCSs. Both the evasion and the attack strategies are adapted to one another (Smith et al., in press).

6 THIS SYSTEM'S GOAL: MACHINE INNOVATION

As was indicated in Section 1, the first workshop indicated some indecision about the exact goals of the LCS approach. Section 2 shows that many of the goals and techniques of the LCS have been clarified and formalized through active research in reinforcement learning. However, Section 4 shows a successful LCS that has much in common with the traditional form. At first glance, this LCS seems to have less of a firm, reinforcement learning based foundation than many modern LCSs . Therefore, one must ask, why the LCS from Section 4, which is applied to a very complex problem, has not suffered from the typical problems of traditional LCSs, discussed in Section 2.

We believe much of the explanation lies in the goal at which the system is directed. From the outset of the fighter aircraft LCS project, the goal has not been to directly *control* fighter aircraft. Instead, the focus has been on the discovery of novel maneuvers for their own sake. The utility of this goal is revealed when one considers the complexity of the aircraft combat task (Shaw, 1988).

In a general sense, one can understand the aircraft's *aerodynamic* characteristics before an actual prototype is generated. However, given the complexity of the aircraft combat task, one cannot map this directly to which maneuvers are advantageous. Therefore, test pilots, and extensive combat trials (mock or real), are generally needed to discover the innovative maneuvers that will convert raw aerodynamics into combat success.

Therefore, the goal of or system simply the discovery of novel maneuvers, not optimality. There is, in general, no such thing as an optimal maneuver, and discovery of such a maneuver is not the system's goal. Discovering many *successful* combat maneuvers in this complex task has quantifiable advantages, including:

- Feedback to aircraft designers with regard to the combat advantages of various aerodynamic characteristics.
- Feedback to fighter pilots on the use of various aircraft characteristics.
- Feedback to customers on the advantages of a given aircraft's characteristics.

These advantages have real-world value. Moreover, they are advantages that could transfer to any number of other tasks where the discovery of novel strategies is difficult, and where new strategies have intrinsic values.

7 IMPLICATIONS OF THIS GOAL

In general, the goal of novelty discovery shifts the perception of what techniques are advantageous. For instance, the pursuit of Q-values that yield a Bellman optimal control strategy seems less critical in this setting. While it is true that one wants to pursue high utility maneuvers, it is not expected, that an "optimal" maneuver can be found. Moreover, enumerating all possible high-utility maneuvers is probably not possible. Finding classifier-based generalizations over the *entire* payoff landscape is probably not possible as well.

Most importantly, maintaining any *consistent* control strategy is not of great importance in this task. The frequent tendency of LCSs to have periodic failures noted in Section 2 is not a particular concern. Consider the typical, but previously unpublished result from the fighter aircraft system shown in Figure 3.

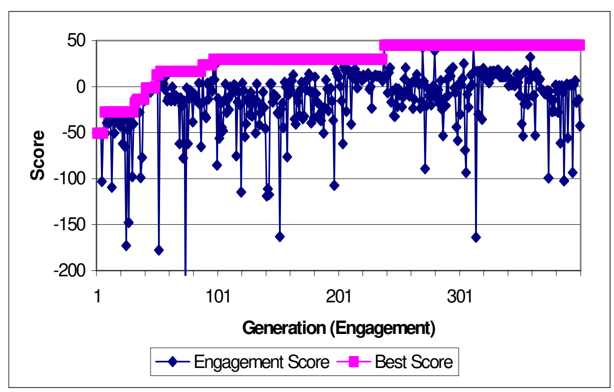

Figure 3 Engagement Score for the LCS aircraft versus Generation Number. In this typical run, only one aircraft is using the LCS, while the other uses standard combat maneuver logic, as in the run shown in Figure 1.

This figure shows that the fighter aircraft LCS frequently loses rules and suffers temporary losses of performance. However, the system moves to improved performance (and new maneuvers). Thus, it accomplishes the goal of *discovery,* but not necessarily the goal of control.

8 TECHNIQUES USED IN THIS SYSTEM

Like any LCS, the fighter aircraft LCS involves a complex interplay of elements. However, the authors believe there are two differences in technique that can be directly related to the particular goals of this system. The first involves conflict resolution and credit assignment, and the second involves effectors and action encoding.

8.1 EPOCHAL CREDIT ALLOCATION

Note that the system employs an *epochal* credit assignment scheme. This scheme is similar to the *profit sharing plan* (Grefenstette, 1988) although it is applied in a "Michigan-style" LCS (Holland & Reitman, 1978). The scheme also employs deterministic conflict resolution, selecting the highest fitness matching classifier at every time step. At first, this may seem very disconnected from the reinforcement literature. However, it can be related to the Monte-Carlo reinforcement learning techniques discussed by Sutton and Barto (1998).

Consider the following notation. Assume the strengths in the aircraft system are analogous to Q-values, environmental messages are analogous to states s, and action messages are analogous to actions a. Given this notation, consider the following Monte Carlo reinforcement learning method, paraphrased from Sutton and Barto (1998):

1) For all states s and all actions a
    a) Initialize $Q(s,a)$ to arbitrary values
    b) Initialize *Returns(s,a)* to an empty list
2) Repeat:
    a) Generate a complete engagement, *deterministically selecting* the maximum $Q(s,a)$ action for each state s visited.
    b) For each state/action pair s,a that occurred in the engagement
        i) Append the engagement score to the list *Returns(s,a)*
        ii) Set $Q(s,a)$ to average(*Returns(s,a)*)

Note that deterministic action selection is required in this algorithm. As Sutton and Barto indicate, this method is expected to converge to an optimal policy, but *only if* "exploring starts" for each engagement are employed. That is, each engagement must start at a randomly selected state, to insure that the complete state-action space is explored.

Clearly, the fighter combat LCS does not employ exploring starts to insure adequate exploration. However, we surmise that the randomness interjected by the GA, and the complexity of the fighter aircraft world, are sufficient to provide for exploration. Also note that returns are not being averaged in the fighter aircraft LCS, although such averaging is necessary in step 2)b)ii) above. However, the GA itself carries out an implicit schema average over generational time.

Therefore, although the credit allocation scheme used in the fighter combat LCS is not strictly similar to Bellman optimal reinforcement learning techniques, it does bear a relationship to these techniques. Note that this is as much as can be said for the Q-learning based techniques used in other LCSs. The authors are not arguing that the scheme used in the fighter aircraft LCS is superior to other techniques, simply that the scheme we have employed is not without foundation for an epochal reinforcement learning task.

Although the fighter combat LCS scheme is unlikely to have provable convergence to Bellman optimal values, recall that such convergence is not necessary for the goals of this problem. What is necessary is good evaluation of classifier schema, so ongoing exploration of the maneuver space can go on. Our empirical results indicate that this is the case in the fighter aircraft LCS.

8.2 ACTION ENCODING

A second important aspect of the fighter combat LCS lies in the environmental effector's interpretation of classifier actions. In effect, the classifiers only *suggest* the actions to the environment, since many actions they may suggest may be aerodynamically impossible. However, because of the nature of the simulation environment (The Air-to-Air System Performance Evaluation Model, AASPEM, one of the standard models of fighter aircraft combat), the simulated aircraft responds with a "nearest feasible response" (in an aerodynamic sense) to the suggested action. This is an artifact of the AASPEM simulation environment that we have exploited in our system.

This means that a single, generalized classifier can have a variety of effects in different contexts. For instance, consider a generalized classifier that suggests Relative Bank Angle = 45degrees, AOA = 20 degrees, and Speed = 350 knots. This rule can have a variety of effects, depending on the (generalized) condition in which it is activated. If the aircraft was at a speed much lower than 350 knots, the rule is likely to result in a fast, but incremental, change *towards* 350 knots. If the aircraft was at a speed nearer to 350 knots, the rule may result in exactly the suggest value being achieved. The same holds for the other suggested action values. Moreover, the aerodynamic coupling between what values can be achieved is automatically handled by the simulation.

The authors believe this has several effects:

- It builds on the linguistic character of the classifier conditions, by giving the actions appropriate linguistic character (i.e., suggesting things like "speed up and bank steeply").

- It "smoothes" the effect of a classifier become less of more general due to the action of the GA, since the meaning of an action in a generalized context is likely to be similar to that in a subsumed, more specific context.
- It "smoothes" the effect of rule deletion, since one rule is likely to have a persistent effect, even if the context of the rule set in which it exists changes. We believe this helps the system avoid the "rise and crash" behavior often observed in traditional LCSs. Moreover, it helps to continue discovery of new maneuvers after rule deletions, which is consistent with the goals discussed in the previous section.

9 OBSERVATIONS AND SUGGESTIONS FOR FUTURE LCS EFFORTS

There are at least three features of the current fighter aircraft LCS that deserve consideration in the design of other LCSs.

One should consider the fighter aircraft LCS credit allocation scheme. Given the efficacy and popularity of Q-learning and related algorithms (e.g., SARSA, which is essentially the implicit bucket brigade), it is easy to overlook that there are well founded techniques that do not follow their form. In particular, the Monte-Carlo RL methods are just as firmly related to reinforcement learning theory. In fact, Sutton and Barto (1998) indicate that such methods may be more appropriate than Q-learning and related methods, when one faces a non-Markovian task.

However, such methods are only appropriate for tasks with clearly defined episodes. The fighter aircraft task is clearly episodic, since engagement take place in a pre-defined time window. When one directs an LCS at an episodic task, epochal credit assignment schemes, like that used in the fighter aircraft LCS, may be of higher utility than methods based on Q-learning, SARSA, or the bucket brigade.

Like the Q-learning based methods used in LCSs, epochal schemes can only approximate the associated reinforcement learning techniques, since such techniques are typically based on table lookup. Given that the focus of the LCS is on finding generalized rules, associated reinforcement learning techniques, be they epochal or not, must be adapted to evolving, generalized rule sets. XCS (Wilson, 1995) seems to have found an appropriate adaptation of Q-learning for an LCS. Finding an appropriate adaptation of well-founded, epochal reinforcement learning schemes to LCS use is an area worthy of future research.

One should also consider the "smoothed" interpretation of actions in the fighter combat LCS. The authors strongly suspect that this is a reason for the LCSs success on this task. Many of the action representations used in LCS research are "brittle". That is, rule deletion, and other forms of classifier set disruption, can easily cause less-than-graceful failure. In the fighter aircraft task, the actions of rules are implicitly interpreted in a linguistic, aerodynamically appropriate manner. This makes each classifier less sensitive changes in the composition of the classifier set, which are likely to occur, given the action of the GA.

Finally, one should consider the goal of the LCS approach in the fighter aircraft LCS. Since there is a real, quantifiable value to the discovery of innovative, high utility fighter combat maneuvers, one can concentrate on the exploration and synthesis aspects of the LCS, without particular consider for the long term stability of any given rule set. While it is true that some stability is necessary for appropriate rule evaluation, and subsequent improvement of rule sets, occasional rule loss can be tolerated in this task.

Quantifying the purpose of the LCS approach was identified as a primary need of the LCS research community at the first workshop. Reinforcement learning control (that is, obtaining optimal strategies in reinforcement learning problems) is certainly a well-quantified goal, for which the LCS approach is useful. However, one should not overlook the utility of the LCS approach for generating novel, innovated approaches to problems. In many domains (like the fighter aircraft task), such open-ended discovery can have a real world, hard cash value. The applicability of the LCS approach to such tasks deserves further consideration.

ACKNOWLEDGMENTS

The authors gratefully acknowledge that this work is sponsored by The United States Air Force (Air Force F33657-97-C-2035 and Air Force F33657-98-C-2045). The authors also gratefully acknowledge the support provided by NASA for the early phases of this project, under grant NAS2-13994.

REFERENCES

Booker, L. B. (1992) Viewing Classifier Systems as an Integrated Architecture. Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas, October 1.

Goldberg, D. E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Grefenstette, J. J. (1988) Credit assignment in rule discovery systems based on genetic algorithms. Machine Learning 3. pp. 225-246.

Holland, J. H. (1992) Adaptation in Natural and Artificial Systems MIT Press.

Holland, J. H., Holyoak, K. J., Nisbett, R. E. and Thagard, P. R. (1986) Induction: Processes of inference, learning, and discovery. MIT Press, Cambridge, MA.

Holland, J. H. and Reitman, J. S. (1978) Cognitive systems based on adaptive algorithms. In Waterman, D. A. and Hayes-Roth, F., " Pattern directed inference systems". Academic Press, NY.

Shaw, R. L. (1998) Fighter Combat : Tactics and Maneuvering. United States Naval Institute Press.

Smith, R. E. and Dike B. A. (1995) Learning novel fighter combat maneuver rules via genetic algorithms. International Journal of Expert Systems, 8(3) (1995) 247-276.

Smith, R. E., Dike, B. A. and Stegmann, S. A. (1994) Inheritance in Genetic Algorithms, in: Proceedings of the ACM 1995 Symposium on Applied Computing. ACM Press. pp. 345-350.

Smith , R. E., Dike, B. A., Mehra, R. K., Ravichandran , B. and El-Fallah, A. (in press). Classifier Systems In Combat: Two-Sided Learning of Maneuvers For Advanced Fighter Aircraft. Computer Methods in Applied Mechanics and Engineering, Elsevier.

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

Watkins, J. C. H. (1989). Learning with delayed rewards. Unpublished doctoral dissertation. King's College, London.

Wilson, S. W. (1994) ZCS: A zeroth-level classifier system, Evolutionary Computation 2(1). pp. 1-18.

Wilson, S. W. (1995). Classifier fitness based on accuracy. Evolutionary Computation, 3(1), 149-176.

Wilson, S. W. (1999) State of XCS Classifier System Research. Technical Report Number 99.1.1 (Unpublished), Prediction Dynamics, Concord, MA.