Forrest, S. (1990). Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks. In S. Forrest (Ed.), *Emergent Computation: Proceedings of the Ninth Annual International Conference of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*, 1–11, Amsterdam. North-Holland.

Forrest, S., Smith, R. E., & Perelson, A. (1992, October). *Maintaining diversity with a genetic algorithm.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison–Wesley.

Goldberg, D. E., Horn, J., & Deb, K. (1992, October). *What makes a problem hard for a classifier system?* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

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

Montanari, D. (1992, October). *Classifier systems with a constant-profile bucket brigade.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Narendra, K. S., & Thatachar, M. A. L. (1989). *Learning automata: An introduction.* Englewood Cliffs, New Jersey: Prentice-Hall.

Riolo (1992, October). *The discovery and use of forward models by adaptive classifier systems.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Riolo, R. L. (1987). Bucket brigade performance: I. Long sequences of classifiers. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 184–195.

Smith, R. E. (1991). *Default hierarchy formation and memory exploitation in learning classifier systems* (TCGA Report No. 91003). Tuscaloosa: University of Alabama. (Ph.D dissertation, University Microfilms No. 91-30,265).

Smith, R. E. (1992, October). *Is a classifier system a type of neural network?* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Smith, R. E. (1992, October). *Memory exploitation in learning classifier systems.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Smith, S. F., & Greene, D. P. (1992, October). *Cooperative diversity using coverage as a constraint.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Sutton, R. (1991). Reinforcement learning architectures for animats. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, 188–296, Cambridge, MA. MIT Press.

Valenzuela-Rendón, M. (1992, October). *Reinforcement learning in the fuzzy classifier system.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

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

Wilson, S. W. (1989). Hierarchical credit allocation in a classifier system. In M. S. Elzas, T. I. Oren, & B. P. Zeigler (Eds.), *Modelling and simulation methodology* (351–357). New York: North–Holland.

Wilson, S. W. (1992, October). *Classifier system mapping of real vectors.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Wilson, S. W. (1992, October). *Toward a GA solution to the discovery problem.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Wilson, S. W., & Goldberg, D. E. (1989). A critical review of classifier systems. *Proceedings of the Third International Conference on Genetic Algorithms*, 244–255.

symbol association task (Davis, 1992), and in prediction (Riolo, 1992). Some architectural details of exploiting memory were also suggested, including the use of a long term internal message stack, and the use of an incrementally altered memory register (Wilson, 1992b).

Another interesting area of research suggested at the workshop was the examination of "environment theory" that outlines the amount of internal message processing needed for a given environment. With environments clearly parameterized, the internal memory processing capabilities of LCSs and other learning systems can be carefully examined (Smith, 1991).

It is particularly revealing to examine internal memory processing in an LCS, since this processing is localized into easily examined messages passed between classifiers. Moreover, it represents an aspect of learning problems where strong cooperation is virtually required. However, the use of internal messages is certainly not a defining issue for the LCS approach. There are far too many LCSs that focus only on stimulus-response behavior with no internal messages. However, the exploitation of internal messages has been a persistent goal of LCS research.

4 Final Comments, Conclusions and Criteria

In closing, one conclusion that can be drawn from the workshop's discussions is that cooperation (in conjunction with the GA) is the central defining feature of the LCS approach. A suggested (albeit slightly paraphrased) definition from the workshop is that

an LCS equals a combination of a GA and cooperativity.

This definition seems to clarify the primary conceptual details of the LCS approach. However, the other issues addressed at the workshop: representation, credit assignment, and internal message processing, are key details that one must consider when implementing the cooperative approach. The effectiveness of these details for any given implementation must be measured against some criteria. The workshop participants suggested the following criteria:

Modeling Criteria Is modeling of a cognitive, biological, sociological or other learning or adaptive system taking place?

Performance Criteria Does performance compare to that of other machine learning techniques?

Analysis Criteria Is mathematical analysis possible?

Environment-Theoretical Criteria How does the performance vary with formal characteristics of learning environments?

Scalabilty Criteria How does performance vary with problem size?

"Real World" Criteria Is there a practical and realizable application in the "real world"?

On-line, Real-time Criteria Is the system performance applicable to real-time problems?

It is hoped that the workshop papers, discussions, and the key concepts digested in this report will begin the development of framing concepts to help researchers to address these criteria in future LCS research.

References

Booker, L. B. (1982). Intelligent behavior as an adaptation to the task environment (Doctoral dissertation, Technical Report No. 243. Ann Arbor: University of Michigan, Logic of Computers Group). *Dissertations Abstracts International, 43(2)*, 469B. (University Microfilms No. 8214966)

Booker, L. B. (1992, October). *Viewing classifier systems as an integrated architecture.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Davis, L. (1992, October). *Covering and memory in classifier systems.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Desjarlais, L., & Forrest, S. (1992, October). *Linked learning in classifier systems: A control architecture for mobile robots.* Paper presented at The First International Conference on Learning Classifier Systems, Houston, Texas.

Workshop discussions indicated that when strong cooperation is needed, it may be useful to supplement GA discovery with some type of automatic *encapsulation*. Encapsulation is a process by which the classifier system operates on progressively larger "chunks" of the learning problem as its fundamental elements. Some suggested encapsulation methods as an attempt to create cooperation by explicitly linking classifiers into structures that are processed either by the GA, the conflict resolution scheme, or the credit assignment scheme as a unit (e.g. "corporate classifiers" (Smith, 1992b; Wilson & Goldberg, 1989)). These approaches amount to an incremental movement from the Michigan-style (cooperative) approach towards a Pitt-style (optimization) approach. One can also imagine adaptive techniques for continuously moving between these approaches. Other suggested encapsulation techniques include adaptive sensors that combine environmental features into chunks, possible techniques to adaptively select "grandfather" classifiers that are isolated from the competitive processes, progressively embedded process primitives, literally subdividing the learning task into well-defined sub-tasks (e.g. the hierarchical classifier systems (Wilson, 1989)), and incorporating sets of LCSs in a subsumption architecture (Desjarlais & Forrest, 1992).

3.3 Representation

Another important issue for the LCS approach is addressed by question 3: syntax and representation. Workshop papers included a variety of representations, including the traditional {1, 0, #} alphabet (Riolo, 1992), classifiers representing hyperplanes of the environmental reward function (Booker, 1992), classifiers that represent fuzzy rules in the style of fuzzy logic controllers (Valenzuela-Rendón, 1992), classifiers that define a CMAC-like mapping (Wilson, 1992a), and classifiers that define connectivity in a neural network (Smith, 1992a). Other representation-related techniques in the LCS approach include partial matching (Booker, 1982; Forrest et al., 1992) and the use of general logical predicates (Wilson, 1992a). Given this distribution of representations, it seems that **representation does not define the LCS approach.** Representation is certainly important in its relationship to the demands of any given learning task. The representation must be capable of effective expression of desired behavior. In addition, *representation is intimately bound to cooperation and related discovery issues.* Thus, although no representation defines the LCS approach *per se*,representation is extremely important to the LCS approach.

3.4 Credit Assignment

An issue that has received a great deal of attention in the LCS literature is that addressed by question 4: credit assignment (Montanari, 1992). Two varieties of credit assignment problems are faced in the LCS approach:

Structural Credit Assignment where one must assign global performance credit to a group of local structures (e.g. classifiers).

Temporal Credit Assignment where one must assign current performance credit to a sequence of classifiers that have acted over the system's past.

In these context of LCSs, these questions are ususally addressed in terms of *reinforcement learning* (learning based on rewards and punishments) as opposed to *supervised learning* (learning with an informative teacher). However, workshop discussions pointed out that LCS methods could employ either supervised or reinforcement learning strategies.

Although important, these credit assignment issues are not particularly defining for the LCS approach. Many other learning techniques face similar problems. The techniques developed in these areas are either similar to LCS techniques (e.g. learning automata algorithms (Narendra & Thatachar, 1989), Q-learning (Watkins, 1989)) or could be easily adapted to LCSs (e.g. DYNA (Sutton, 1991)). Once again, however, these issues have an important relationship to cooperativity and discovery. Where many learning schemes have fixed, non-cooperative structures that are tuned by a credit assignment algorithm, the LCS approach suggests an adaptive set of cooperative structures to allow for more parsimonious and adaptive solutions. The implication of this approach to various credit assignment schemes is an important direction for LCS research.

3.5 Internal Processing

The final workshop question (5) may represent the most challenging possibility for the LCS approach: adaptive computation with internal messages. Certainly, if learning systems are ever to deserve the label "intelligent," they will have to actively use memory. In the LCS approach, this memory is provided through internal message passing. Several workshop papers included the use of memory, in a simple delayed response problem (Smith, 1992a), in a

When cooperation exists in an LCS, *emergent computation* is also possible. Forrest (1990) offers *superposition* as a characteristic feature of emergent computation: superposed effects are not emergent computation. For instance, effects of the simple, two-rule default hierarchy listed above can be accounted for by superposition: its effect is a combination of effects of each of the two rules in isolation. Contrast this to a *rule chain*. In a rule chain one rule's action can set up a situation for another rule. Clearly, this effect can lead to strong cooperation: both the system's performance and fitness values are dependent on the combination of rules in the chain. However, it may not be possible to account for the effects of a rule chain by superposition of the isolated effects of the rules in the chain (Smith, 1992b). Therefore, these effects could be called emergent computation. Strong and weak cooperation, and related emergent computation, are important categories of LCS performance.

To underscore the nature of the LCS's cooperative approach, Valenzuela-Rendón suggests a thought experiment. Consider a GA optimization problem where a fitness value for any given solution $\vec{x}$ is given by a known function $f(\vec{x})$. The traditional simple GA approach to this problem (a population of length-ℓ strings representing complete specifications of $\vec{x}$ solutions) will be referred to as "the optimization approach". Now imagine breaking each population member into a set of n literal substrings of length ℓ/n. Each substring is "marked" as to its correspondence to the parameters of $\vec{x}$. Further, imagine each of these substrings is treated as a separate population member. This will be called the "cooperative approach". In this approach, each population member must be selected and grouped with other members to form a complete set of parameters for the fitness evaluation function $f(x)$. Thus, population members must cooperate to form a high-fitness solution. Under the details of some particular conflict resolution scheme, the cooperation can be either weak (where substrings do not effect each other's fitness values) or strong (where substrings effect each other's fitness values). Effects can also be emergent (where substrings combine to produce nonlinear effects on the fitness function that cannot be accounted for by superposition). Although this thought experiment does not include the details of conflict resolution, credit allocation, and representation necessary in a machine learning problem, it does illuminate details of the cooperative dilemma. Extensions and examinations of this experiment may also tie the concepts of cooperation in the LCS approach to the concepts of *deception* (Goldberg, 1989) in GA optimization.

There are two key differences between the optimization approach and the cooperative approach cited in this experiment. The first of these is the *timing of substring evaluation*. The traditional optimization approach evaluates an entire population of complete solutions at each GA generation. After evaluation and selection in this approach, substrings are placed in a variety of contexts through genetic recombination. This variety of contexts is necessary to allow for good estimates of expected substring fitness values.

The cooperative approach evaluates substrings in a variety of contexts before applying the GA. These contexts are determined by selecting substring combinations and performing any necessary conflict resolution. For a particular class of problems, it may be possible to generate these contexts in an intelligent fashion that quickly yields a good estimate of substring utility. If effective estimates can be evaluated for a population of substrings with only a few fitness evaluations, there may be an execution time advantage of this approach over the optimization approach. However, in the cooperative approach one must consider whether the local search operators (GA, credit assignment, and otherwise) lead to a global optimization strategy. Also, one must consider the possible *parasites* that can emerge to exploit the local operation of the system while yielding no net gain of performance at the global level (Smith, 1992b).

The other key difference between the optimization approach and the cooperative approach is the underlying basis for search. In optimization the basis is complete solutions. Thus, the GA searches for similarities (schemata) between high-fitness solutions. The search basis in the cooperative approach is the substrings themselves. Thus, the schema basis is the similarity between these partial solutions. If, in a particular problem, cooperative partial solutions can jointly benefit by exploring their similarities, this cooperative approach may have another advantage.

3.2 Discovery

Cooperative effects have important implications for the topic considered in question 2: classifier discovery and maintenance with the GA. In both weak and strong forms of cooperation, the GA must be capable of maintaining a diverse population at steady state. Workshop papers by Forrest et al. (1992) and by Goldberg, Horn, and Deb (1992) consider maintaining diversity under the action of the GA. The GA also is impacted by the cooperative approach through performance feedback: does a given set of mechanisms in a cooperative approach provide fitness values that allow for correct selection and application of genetic operators? In one paper presented at the workshop, this was demonstrated not to be the case for certain mechanisms (Smith, 1992b). The implications of the cooperative approach also effect other LCS discovery operators, including "covering" (Davis, 1992). This observation implies that examination and development of "cooperative" GA mechanisms are key to development of the LCS approach.

The fundamental nature of these early questions indicates a lack of a definite framework for LCSs. This is 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.

In a paper presented at the 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 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 the workshop's discussions was to clarify these defining, conceptual issues.

3 Developing Concepts for the LCS Approach

The workshop's conclusions on defining LCS 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:

1. What methods can create *cooperation* in an LCS, despite the competitive nature of LCS and GA mechanics?
2. What *discovery* mechanisms can help the GA to find and maintain useful classifier sets?
3. What is an appropriate syntax or *representation* in an LCS?
4. How can one insure effective *credit assignment* in an LCS?
5. How can the LCS create computationally useful *internal message processing* in an LCS?

3.1 Cooperation

The discussions indicated that the central, overarching LCS research issue is that of question 1: cooperation. Each of the other issues involves questions of cooperation. In fact, as the introductory paragraph indicates, cooperation distinguishes GA-based optimizers from LCSs.

The need for cooperation in an LCS can emerge from several sources, including:

- The need to *cover* a set of possible inputs (as in pattern classification problems (Forrest, Smith, & Perelson, 1992; Smith & Greene, 1992)).
- The need to perform a *tiling* that yields an effective input–output mapping (Wilson, 1992a; Valenzuela-Rendón, 1992).
- The need to form a parsimonious rule set through *default hierarchies* (Holland, Holyoak, Nisbett, & Thagard, 1986).
- The need to form sequences of actions (*rule chains*) (Riolo, 1987).

Cooperation in the LCS approach can be divided into two categories suggested by Wilson (1992a).

Weak cooperation exists when classifiers can combine to effect overall system performance, but do not effect one another's fitness values. Consider a classifier that effectively responds to a set of messages. Consider another classifier that effectively responds to a different (non-overlapping) set of messages. The combination of the two rules effectively respond to the union of these two sets. In this combination, the two rules effect the overall system's performance. However, they do not effect one another's fitness values.

Strong Cooperation exists when classifiers can combine to effect overall system performance, and also effect one another's fitness values as well. Consider a similar case to that listed above, but with two classifiers that match overlapping sets of messages. If one of these classifiers (the *exception*) can shield the other (the *default*) from potential errors, the two can function as a simple *default hierarchy*. In this case, the default's fitness is effected by the existence of the exception.

A Report on The First International Workshop on Learning Classifier Systems
NASA Johnson Space Center
Houston, Texas
October 6-9, 1992

R. E. Smith

November 11, 1992

1 Purpose of This Report

When Holland first introduced the use of artificial, genetics-based adaptive systems in his seminal 1975 book, he recognized the potential of these systems for evolving populations of structures that *jointly* act to perform useful computation. Holland's suggestions on genetic adaptive plans led to what are now known as *genetic algorithms* (GAs). His suggestions on the use of these plans for evolving interacting populations led to what are now known as *learning classifier systems* (LCSs). Genetic algorithms have had a great measure of success in search and optimization problems where the end goal is improvement in a single, autonomous structure. Discovering sets of interacting structures in the LCS involves many complex issues that are not involved in more traditional optimization and search problems. However, the rapid rise of interest in GAs for optimization has often delegated discussion of LCS issues to the back burner.

On October 6-9, 1992 at Johnson Space Center in Houston, Texas, a group of researchers met at **The First International Workshop on Learning Classifier Systems**. The meeting was an effort to have a small, informed group focus on the issues involved in LCSs[1]. The meeting was self-organized: a "starter" group was selected by Dr. Manuel Valenzuela-Rendón, and myself, and that group recommended other invitees. The meeting consisted of presentations from extended abstracts, and "brainstorming"–style discussion sessions. This report is an attempt to digest and relate the results of the workshop. Its intent is threefold: to serve as a record of the workshop, to provide a unifying focus for a workshop-related publication, and to begin to frame the concerns and direction of future LCS research.

2 Fundemental Questions

The workshop participants all share a long-term interest in LCSs. Despite the considerable experience of this group, the earliest discussions revealed that the participants had fundamental questions about LCSs. For instance:

- What features distinguish LCSs from other systems?
- What niche in machine learning are LCSs best able to fill?
- What niche in machine learning is LCSs intended 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?

[1] Please note that I can claim to be "author" of this report only in the strictest sense: I wrote it down. However, the information presented here is the work of all the conference attendees.