What Makes a Problem Hard for a Classifier System?

David E. Goldberg, Jeffrey Horn, and Kalyanmoy Deb
Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

IlliGAL Report No. 92007
May 1992

Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building
104 South Mathews Avenue
Urbana, IL 61801

What Makes a Problem Hard for a Classifier System?

David E. Goldberg, Jeffrey Horn, and Kalyanmoy Deb
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

1 Introduction

The first author once remarked to a colleague that "classifier systems are a quagmire—a glorious, wondrous, and inviting quagmire, but a quagmire nonetheless." To this piece of overblown commentary the colleague replied, "GAs are a quagmire." And of course he was right. But GAs are less swampy then they once were, and with the six pieces of the GA puzzle falling into place (Goldberg, Deb, & Clark, 1991), not only do we better understand 1) the things GAs process (building blocks), 2) how to get enough of them, 3) how to make them grow, 4) how to permit their exchange, 5) how to make good decisions among them, and 6) what problems are tractable with respect to them (with respect to building blocks), but we will soon be able to fit these pieces together to form an integrated, if somewhat coarse, theory of simple GA convergence. This theory should delimit quantitatively the bounds of selectorecombinative GAs and it should give estimates of GA time complexity similar to those of computational learning theory.

These efforts will have important ramifications for classifier systems and other genetics-based machine learning efforts, both directly and indirectly, and with advances in reinforcement learning (Sutton, 1992) and a better understanding of the ecological opportunities and pitfalls (Holland, 1992) of classifier systems as a result of what might be termed their *on-the-fly connectionism*, there is reason to believe that the classifier-system swamp can be drained, eventually leaving us with working versions of these systems that were so boldly outlined so many years ago (Holland, 1971, 1975; Holland & Reitman, 1978).

To get to that point, however, previously learned lessons must be transferred and understood in the context of classifier systems, and one of the key lessons of the past decade of GA research has been to understand what makes a problem difficult for a GA and take measures that insure the solution of problems of bounded difficulty. Approaches based on such *bounded pessimism* remain controversial, but certainly they have been an aid to demarking the boundary between problems that these algorithms are better-than-lucky in solving and those where a GA's primary hope is prayer and subsequent divine intervention. In this paper, we will take some steps toward understanding what makes a problem hard for a classifier system. Specifically, we will connect recent results in genetic optimization of *massively multimodal* and *deceptive* functions to the problems in classifier system learning. The connection will not be wholly satisfying, but it will help illuminate three facets of what makes a problem hard for a classifier.

We start by listing the assumptions to be made about the classifier system. Although important aspects of classifier system operation will be ignored, enough remains to make this discussion germane to real classifier system design. Thereafter, a recent study in massively multimodal function optimization is briefly reviewed. The connection between this work and the classifier problem is then drawn.

2 Classifier System Assumptions

We imagine an idealized classifier system as follows:

1. The rule and message (performance) system is set up for stimulus-response boolean function learning.
2. The bidding structure is ideal.

3. Default hierarchy formation is perfect.

4. Context sensitivity is negligibly small.

The conditions are not unlike those used elsewhere in an earlier attempt to identify what makes a problem difficult for a classifier system (Valenzuela-Rendon, 1989). Each of these assumptions is discussed in somewhat more detail.

A stimulus-response system is imagined much like that used in Wilson's (1987) Boole system. A binary message of fixed length is posted, and one or more classifiers with conditions drawn from the usual ternary alphabet, $\{0, 1, \#\}$, and a single-bit action are permitted to match messages. The winning classifier (or set of classifiers) is allowed to post its action, this action is either reinforced or not, and the process continues. The restriction to binary classification task can easily be lifted, but getting beyond stimulus-response in a pure classifier system has proved to be nontrivial. The addition of internal states and the confrontation of even the simplest sequential learning tasks presents difficult challenges to self-organizing classifier systems (Smith, 1991). The issues dealt with here, while not orthogonal to those posed by the challenge of sequential learning, may be examined in relative isolation from that consideration.

By an ideal bidding structure we mean that the strength-bid adjustment mechanisms adapt quickly and correctly to elevate those rules that are responsible for correct classification and lower those rules responsible for incorrect classification. The assumption of perfect default hierarchy formation is a related notion whereby it is assumed that exception rules are prioritized over default rules in a stable manner such that for a given level of correct classification, high priority rules are given precedence over lower priority rules, allowing the exceptions to cover the mistakes of the lower priority rules. Variable-separation schemes (Smith, 1991; Smith & Goldberg, 1990) give performance closer to this idealization than the usual fixed-separation scheme.

The least acceptable of the assumptions is the last one. In much classifier-system learning—especially in those systems where default hierarchies or chaining are encouraged—the presence or absence of a rule can directly affect the strength of another. Here we imagine that these interactions are sufficiently small to be ignored. This will permit us to make a reasonably clean connection to function optimization. As the weakest link in the chain, context sensitivity is an important issue that must receive greater attention, and perhaps a boundedly pessimistic approach will help sort it out, but we do not consider it further herein.

In the next section, we summarize a recent study of genetic optimization in functions that are both deceptive and massively multimodal.

3 Massively Multimodal Function Optimization

Elsewhere (Goldberg, Deb, & Horn, 1992) the construction and optimization of functions that are both massively multimodal and deceptive has been considered. In particular, a 30-bit function with 32 global optima and over five million local optima was solved to global optimality such that the final populations contained instances of each of the 32 global optima. In this section, we briefly review that work.

The 30-bit test function was constructed from *bipolar deceptive* functions (figure 1). A bipolar deceptive function over strings of length $2l$ has two relatively isolated global optima located maximally distant from one another (global complements of each other). A plethora of local optima is then located halfway between the globals. Straightforward counting shows that there are $\binom{2\ell}{\ell}$ local optima for a total of $\binom{2\ell}{\ell} + 2$ optima per subfunction. In the particular problem with six-bit subfunctions ($2l = 6$), there are $\binom{6}{3} + 2 = 22$ optima per subfunction. With five subfunctions added together independently in the test function, there are $22^5 = 5.15(10^6)$ optima among the $2^{30} = 1.07(10^9)$ alternatives. Of these optima $2^5 = 32$ are global.

Using simple one-point crossover, sharing, and fitness scaling, a GA was able to find all globals (tight linkage was assumed). For a representative run, the average, maximum, and minimum number of copies of global solutions are shown in figure 2. After generation 20, all global solutions are always represented in the population, and at steady state (achieved at roughly generation 50), no fewer than 30 copies of each of the 32 global points is maintained in the population.

These results bode well for the maintenance of an appropriate rule set in a classifier system, a topic taken up in the next section.

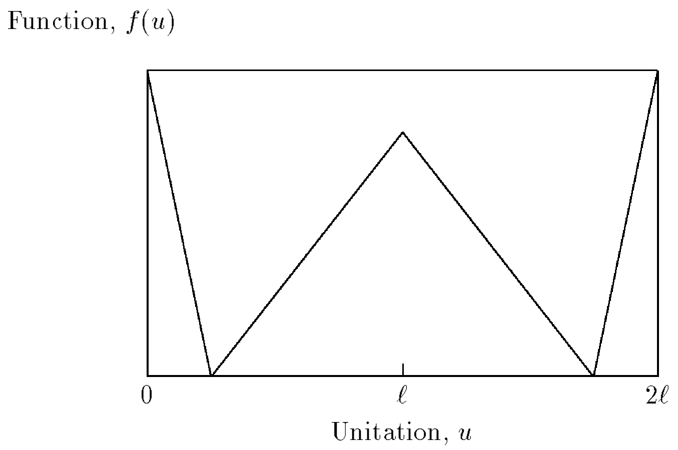

Figure 1: Schematic of a bipolar deceptive function.

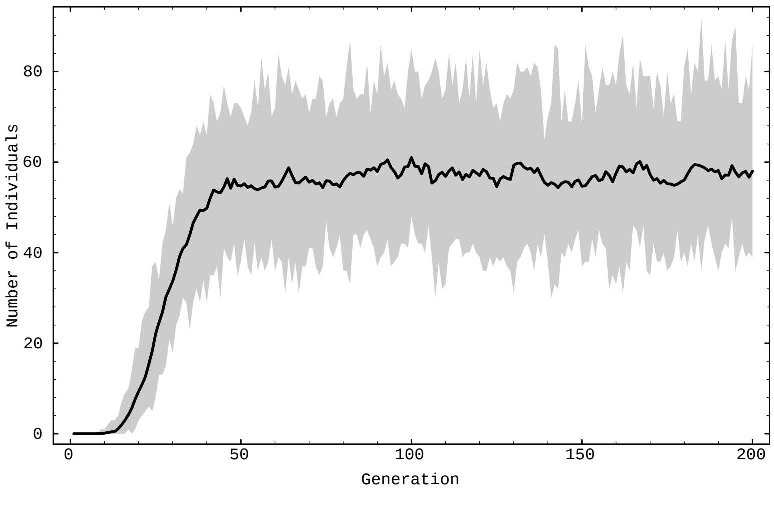

Figure 2: A graph of the range of subpopulation sizes (shaded zone) among the 32 global optima in the 30-bit, five subfunction problem for each of 200 generations demonstrates that a simple GA is able to stably find all 32 points of the global set. Also plotted as a dark solid line is the mean subpopulation size among the global points.

4 The Classifier System Connection

It is nice that a GA can optimize functions that are both massively multimodal and deceptive, but what does this have to do with classifier systems? Classifier systems are closer to niched genetic optimization than simple genetic optimization, because in classifier systems a fairly large number of rules is usually required to solve problems. Of course this is not news, and in this section, we start by briefly reviewing some of the efforts to use and analyze various niching schemes in classifier systems. Thereafter, we examine the mapping between niched function optimization and classifiers more carefully, and we consider three facets of classifier-system-difficult problems.

4.1 Review of niching

It has long been recognized that classifier discovery mechanisms had better be able to create populations of rules that cover the possible inputs properly. Going back to Holland and Reitman (1978), crowding was incorporated in the CS-1 system, while in Booker (1982) sharing was suggested and used. Goldberg (1983) used crowding to learn a set of control rules. More recently, Smith and Valenzuela-Rendon (1989) showed analytically that genotypic fitness sharing (as opposed to Booker's use of phenotypic sharing) can allow the LCS to maintain enough rules to cover a boolean function. In these and other studies genetic algorithms have been modified in manners that permit stable formation of niches and species. In this way, searching for a co-adapted set of rules is closer to the multimodal function optimization problem solved by a niched GA than it is to the problem of finding any global optimum that is solved by simple GAs.

4.2 Mapping classifiers to binary function optimization

The intuitive mapping of multimodal function optimization to classifier systems should be clear. If we imagine a set of rules that form a functional default hierarchy, and if these rules are present at some time during a run, under the ideal bidding and perfect default hierarchy formation assumed in section 2, these rules will each achieve maximum payoff. Other rules that are error prone will receive something less than maximum payoff, and during the selectogenetic cycle, the GA must maintain the well co-adapted set of rules that has been discovered to solve the problem. This requirement to maintain a set of diverse rules stably is much like the requirement in multimodal function optimization to maintain one or more representatives of every element, or a representative set of elements, of the global set.

Of course, there are differences between the two situations. We have already discussed how we have dispensed with the complication of context sensitivity, and most classifier systems do not use binary structures. In a moment, we discuss context sensitivity in somewhat more detail. Here we imagine an idealized classifier system that maps to binary structures directly; we also consider how to map the usual ternary conditions to binary structures.

One way to map classifier systems to binary structures is to imagine what we will call a f-# classifier. Here, the population is imagined to contain a number of length-ℓ structures drawn from the 2-alphabet $\{\mathrm{f}, \#\}$, where the "f" suggests expansion over the usual 2-alphabet $\{0, 1\}$ and the # is the usual don't-care character. For example, the structure f## expands to the set of rules $\{0\#\# : 0, 1\#\# : 0, 0\#\# : 1, 1\#\# : 1\}$. Note how the enumeration of the action is implied. To make the mapping complete we would have to imagine how the structures would receive payment, because each f-# classifier expands to a set of the usual classifiers, but since the goal here is to show some correspondence we postpone that activity and consider another means of mapping classifiers to binary structures.

Another way to connect classifiers to binary structures is to perform a mapping from a single ternary position to two binary positions as follows: $0 \rightarrow 00$, $1 \rightarrow 01$, $\# \rightarrow 10$ or 11. Such mappings have been used elsewhere (Goldberg, 1983) to permit fast assembler-level matching, but here our goal is simply to show the relevance of binary function optimization to classifier system study. Although it would also be possible to consider deceptive functions over a ternary alphabet directly, we postpone this exercise to learn the lessons of our optimization-classifier connection.

4.3 Three concerns

Having considered two ways to connect function optimization to classifier learning, we are in a better position to identify the lessons of genetic optimization to identifying classifier hard problems. Three difficulties carry over immediately:

1. deception;
2. multimodality;
3. lack of separation.

Deception has been discussed for some time and at some length, and functions that are average-sense misleading to classifier systems should be among the most difficult. As we've seen, massive multimodality can overwhelm a classifier system's niching algorithm, making it difficult to find a good default hierarchy stably. We expect these two difficulties to come together in a definition of *generalized deception*, a definition that will unify the notions of simple deception and bipolar deception, thereby allowing us to imagine functions with an arbitrary placement of the global set and determine where the worst place to put local optima might be.

Lack of separation between global solutions can be a problem to a niched GA, because for a given population size, there is a lower limit to the distance between global solutions that the GA can discriminate. In other words, below some critical distance, two different solutions are considered to be in the same niche. In the case of a classifier system, if both of these rules are required to solve a particular problem, over time the GA will be unable to maintain both rules in the population stably. Elsewhere (Goldberg, 1989), sexual differentiation has been suggested as one way to maintain marginally different genotypes in a population stably, and this may prove useful to overcome difficulty in solving classifier problems that lack separation.

5 Conclusions

This extended abstract has considered the ramifications for classifier system learning of some recent results in genetic optimization of functions that are both deceptive and massively multimodal. This connection suggests that classifier problems whose global optima are improperly predicted by low-order schemata, problems that have many locally optimal rules, and problems that have insufficient separation between rules in a default hierarchy should be difficult for classifier systems. Of course, these facets of classifier-hardness are unlikely to be the only things that can cause a classifier system grief, and context sensitivity has already been mentioned as an area of concern. Nonetheless, identifying these important aspects of what it means for a problem to be classifier-system difficult is an important step toward assembling a more rigorous theory of classifier system convergence.

Acknowledgments

The authors acknowledge the support provided by the US Army under Contract DASG60-90-C-0153 and by the National Science Foundation under Grant ECS-9022007.

References

Booker, L. B. (1982). Intelligent behavior as an adaptation to the task environment. (Doctoral dissertation, University of Michigan).

Deb, K. (1989). *Genetic algorithms in multimodal function optimization* (MS Thesis and TCGA Report No. 89002). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

Deb, K., Horn, J., & Goldberg, D. E. (1992). *Multimodal deceptive functions.* (IlliGAL Report No. 92003). Urbana: University of Illinois, Illinois Genetic Algorithms Laboratory.

Deb, K., & Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. *Proceedings of the Third International Conference on Genetic Algorithms*, 42–50.

Goldberg, D. E. (1983). *Computer-aided gas pipeline operation using genetic algorithms and rule learning* (Doctoral dissertation, University of Michigan).

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

Goldberg, D. E., Deb, K., & Clark, J. H. (1991). *Genetic algorithms, noise, and the sizing of populations* (IlliGAL Report No. 91010). Urbana: University of Illinois, Illinois Genetic Algorithms Laboratory.

Goldberg, D. E., Deb, K., & Horn, J (1992). *Massive multimodality, deception, and genetic algorithms* (IlliGAL Report No. 92005). Urbana: University of Illinois, Illinois Genetic Algorithms Laboratory.

Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*, 41–49.

Holland, J. H. (1971). Processing and processors for schemata. *Proceedings of the Symposium on Associative Information Techniques*, 127–146.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI: University of Michigan Press.

Holland, J. H. (1992). *Adaptation in natural and artificial systems.* Cambridge, MA: MIT Press.

Holland, J. H. & Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. In *Pattern-Directed Inference Systems.* Academic Press, Inc., 313–329.

Smith, R. E. (1991). *Default hierarchy formation and memory exploitation in learning classifier systems* (Doctoral Dissertation and TCGA Report No. 91003). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

Smith, R. E., & Goldberg, D. E. (1990). Variable default hierarchy separation in a classifier system. *Foundations of Genetic Algorithms*, 148–167.

Smith, R. E., & Valenzuela-Rendon, M. (1989). A study of rule set development in a learning classifier system. *Proceedings of the Third International Conference on Genetic Algorithms*, 340-346..

Sutton, R. S. (Ed.) (1992). Special issue on reinforcement learning. *Machine Learning, 8*(3/4), 225–395.

Valenzuela-Rendon, M. (1989). *Two analysis tools to describe the operation of classifier systems* (Doctoral Dissertation and TCGA Report No. 89005). Tuscaloosa: University of Alabama, The Clearinghouse for Genetic Algorithms.

Wilson, S. W. (1986). *Classifier learning of a boolean function.* (Research Memo RIS No. 27r). Cambridge: The Rowland Institute for Science.