Genetic Algorithms Are NOT Function Optimizers

Kenneth A. De Jong
Computer Science Department
George Mason University
Fairfax, VA 22030, USA
kdejong@aic.gmu.edu

Abstract

Genetic Algorithms (GAs) have received a great deal of attention regarding their potential as optimization techniques for complex functions. The level of interest and success in this area has led to a number of improvements to GA-based function optimizers and a good deal of progress in characterizing the kinds of functions that are easy/hard for GAs to optimize. With all this activity, there has been a natural tendency to equate GAs with function optimization. However, the motivating context of Holland's initial GA work was the design and implementation of robust adaptive systems. In this paper we argue that a proper understanding of GAs in this broader adaptive systems context is a necessary prerequisite for understanding their potential application to any problem domain. We then use these insights to better understand the strengths and limitations of GAs as function optimizers.

1 INTRODUCTION

The need to solve optimization problems arises in one form or other in almost every field and, in particular, is a dominant theme in the engineering world. As a consequence, an enormous amount of effort has gone into developing both analytic and numerical optimization techniques. Although there are now many such techniques, there are still

large classes of functions which are beyond analytical methods and present significant difficulties for numerical techniques. Unfortunately, such functions are not bizarre, theoretical constructs; rather, they seem to be quite commonplace and show up as functions which are not continuous or differentiable everywhere, functions which are non-convex, multi-modal (multiple peaks), and functions which contain noise.

As a consequence, there is a continuing search for new and more robust optimization techniques capable of handling such problems. In the past decade we have seen an increasing interest in biologically motivated approaches to solving optimization problems, including neural networks (NNs), genetic algorithms (GAs), and evolution strategies (ESs). The initial success of these approaches has led to a number of improvements in the techniques and a good deal of progress in understanding the kinds of functions for which such techniques are well-suited.

In the case of GAs, there are now standard GA-based optimization packages for practical applications and, on the theoretical side, continuing efforts to better understand and characterize functions that are "GA-easy" or "GA-hard". However, with all this activity, there is a tendency to *equate* GAs with function optimization. There is a subtle but important difference between "GAs *as* function optimizers" and "GAs *are* function optimizers".

The motivating context that led to Holland's initial GA work was the design and implementation of robust *adaptive* systems. In this paper we argue that a proper understanding of GAs in this broader adaptive systems context is a necessary prerequisite for understanding their potential application to any problem domain. We then use these insights to better understand the strengths and limitations of GAs *as* function optimizers.

2 WHAT IS A GENETIC ALGORITHM?

Figure 1 provides a flow diagram of a fairly generic version of a GA. If we ask what this algorithm is intended to do, we find ourselves in an awkward "cart before the horse" situation. The algorithm in question wasn't designed to solve any particular problem (like sorting, tree traversal, or even function optimization). Rather, it is a high level simulation of a biologically motivated adaptive system, namely evolution. As such, the kinds of questions one asks have a different slant to them. We would like to know what kind of emergent behavior arises from such a simple set of rules, and how changes in the algorithm affect such behavior. As we understand better the kind of adaptive behavior exhibited by GAs, we can begin to identify potential application areas that might be able to exploit such algorithms.

Such discussions about evolutionary systems are not new, of course, and certainly predate the emergence of GAs. Although a precise and agreed upon statement of the role of evolution is difficult to find, I think it is fair to say that there is general agreement that its role is not function optimization in the usual sense of the term. Rather, one thinks of evolution in terms of a strategy to explore and adapt to complex and time-varying fitness landscapes.

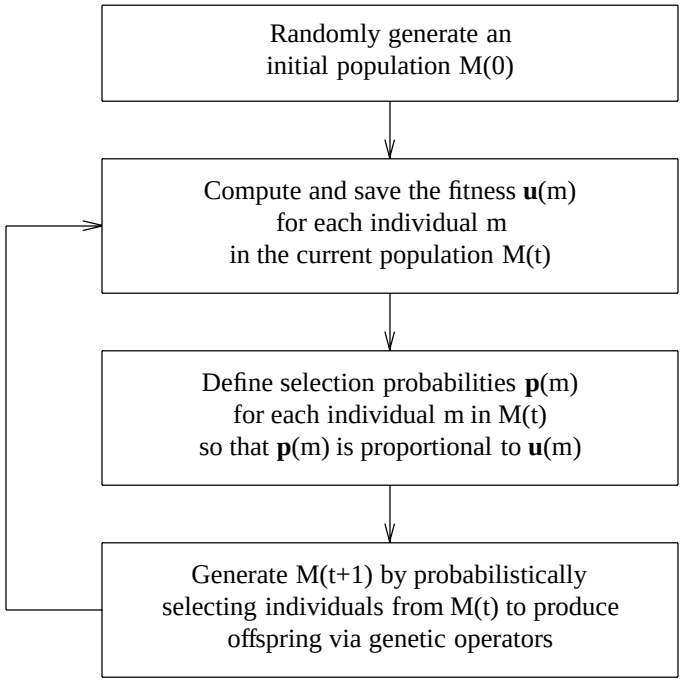

Figure 1. A Canonical Genetic Algorithm

Analyzing this evolutionary adaptive behavior is surprisingly difficult even for high level abstractions such as GAs. Holland's analysis [Holland75] in terms of near-optimal allocation of trials in the face of uncertainty is still today one of the few analytical characterizations we have of global GA behavior. Much of the early GA research was an attempt to gain additional insight via empirical studies in which the fitness landscape was defined by carefully chosen time-invariant memoryless functions whose surfaces were well understood and by observing how GA populations evolved and adapted to these landscapes (see, for example, [DeJong75] or [Bethke81]).

Several important observations came out of these early studies. First, the actual behavior exhibited by such a "simulation" varied widely as a function of many things including the population size, the genetic representation chosen, the genetic operators used, and the characteristics of the fitness landscape. Second, it became very clear that there was no universal definition of what it meant for one simulation run to exhibit "better" adaptive behavior than another. Such measures were more easily motivated by particular task domains and, as such, frequently emphasized different aspects of adaptive behavior. Finally, the robust adaptive behavior observed on these artificial fitness functions gave

rise to the belief that GAs might serve as a *key element* in the design of more robust global function optimization techniques. We explore these observations in more detail in the next few sections.

3 BEHAVIOR EXHIBITED BY GAs

Suppose we select a simple function like $f(x)=x^2$ to define an (unknown) fitness landscape over the interval [0,4] with a precision for x of 10^{-4}. Suppose further that we represent all the legal values in the interval [0,4] as binary strings of fixed length in the simplest possible manner by mapping the value 0.0 onto the string 00...00, $0.0 + 10^{-4}$ onto 00...01, and so on, resulting in an order-preserving mapping of the interval [0,4] onto the set {00...00,...,11...11}. Finally, suppose we run the simulation algorithm in Figure 1 by randomly generating an initial population of binary strings, invoking f(x) to compute their fitness, and then use fitness-proportional selection along with the standard binary string versions of mutation and crossover to produce new generations of strings. What sort of behavior do we observe? The answer, of course, is not simple and depends on many things including the choice of things to be measured. In the following subsections we illustrate 3 traditional viewpoints.

3.1 A GENOTYPIC VIEWPOINT

The traditional biological viewpoint is to consider the contents of the population as a gene pool and study the transitory and equilibrium properties of gene value (allele) proportions over time. In the case of a canonical GA, it is fairly easy to get an intuitive feeling for such dynamics. If fitness-proportional selection is the only active mechanism, an initial randomly generated population of N individuals fairly rapidly evolves to a population containing only N duplicate copies of the best individual in the initial population. Genetic operators like crossover and mutation counterbalance this selective pressure toward uniformity by providing diversity in the form of new alleles and new combinations of alleles. When they are applied at fixed rates, the result is a sequence of populations evolving to a point of dynamic equilibrium at a particular diversity level. Figure 2 illustrates how one can observe these effects by monitoring the contents of the population over time.

Note that, for the fitness landscape described above, after a relatively few number of generations almost all individuals are of the form 1..., after a few more generations, the pattern 11... dominates, and so forth. After several hundred generations we see that, although more than 50% of the gene pool has converged, the selective pressures on the remaining alleles are not sufficient to overcome the continual diversity introduced by mutation and crossover, resulting in a state of dynamic equilibrium at a moderate level of genetic diversity.

Gen 0	Gen 5	Gen 50	Gen 200	Gen 500
01111100010011	11000001011100	11111101010101	11111111010011	11111111011110
00010101011100	11100001000010	11111110011101	11110111011110	11111111110011
01001000110010	11110100010100	11111011001001	11111011010110	11111110010001
00110001110010	11110111100011	11111101000001	11111110101011	11111111011010
01010001110010	11111110100001	11111100110111	11111100001011	11111111110000
00010000010001	11111100011001	11111101000101	11111111011111	11111111111011
11001100000110	11010100000110	11111101001001	11111110100001	11111011010101
10110001110010	11111101000001	11111001011001	11111111110001	11111111110110
10100001111100	11111100111001	11010111110001	11111111111111	11111111110000
00100001101001	11110111011001	11111101011101	11110110110101	11111111000010
01010100100011	11110001111111	11101111100111	11111101110011	11111111001010
01000101100011	11110111100011	11111100110111	11111100010001	11111111111000
00010001111110	11110111100001	11111111010011	11111111011001	11111111110000
10101111100111	11110111011100	11111111010011	11101111111001	11111111010001
01111101010100	11101010100100	11110100000111	11111111100011	11111111011000
11000011100110	11010100011000	11111101011001	11111111110011	11111111010001
11001101100001	11100001110001	11111100101001	11111111111011	11111111110110
11110111011100	11110110011011	11111011100101	11111001001010	11111111110010
11001010100000	11010101000001	11111100000001	11111110111010	11111111110100
00101000101111	11010001010110	11111111100101	11111110011011	11111111110011
01000011111110	11110100010101	11111111001001	11111110110111	11111111111000
11100101000001	11110101100001	11111001010011	11111111101001	11111111101111
01001001100110	11110111100001	11111100101011	11111101010011	11111111001111
01000101100011	11111101000001	11111011011000	10111100111010	11111111011000
00111001100000	11110010011100	11110110001101	11111111010011	11111111011000
01110100100100	11111100111001	11111100111011	11111101111010	11111111010110

Figure 2: A Genotypic View a GA Population

3.2 A PHENOTYPIC VIEWPOINT

An alternative to the genotypic point of view is to focus on the "phenotypic" characteristics of the evolving populations produced by GAs. The phenotype of an individual is the physical expression of the genes from which fitness is determined. In the example introduced above, the phenotype is simply the real number x defined by a particular binary gene value. Although not true in general, in this particular case there is a one-to-one mapping between genotypes and phenotypes.

Since fitness is defined over this phenotype space, one gains additional insight into the adaptive behavior of GAs by plotting over time where the individuals in the population "live" on the fitness landscape. Figure 3 illustrates this behavior for the fitness landscape defined above. As one might expect, the population is observed to shift quite rapidly to inhabiting a very small niche at the rightmost edge of the phenotype space.

Suppose one creates more interesting landscapes with multiple peaks. Does a GA population evolve to steady states in which there are subpopulations inhabiting each of the peaks? For the typical landscapes used which are memoryless and time-invariant,

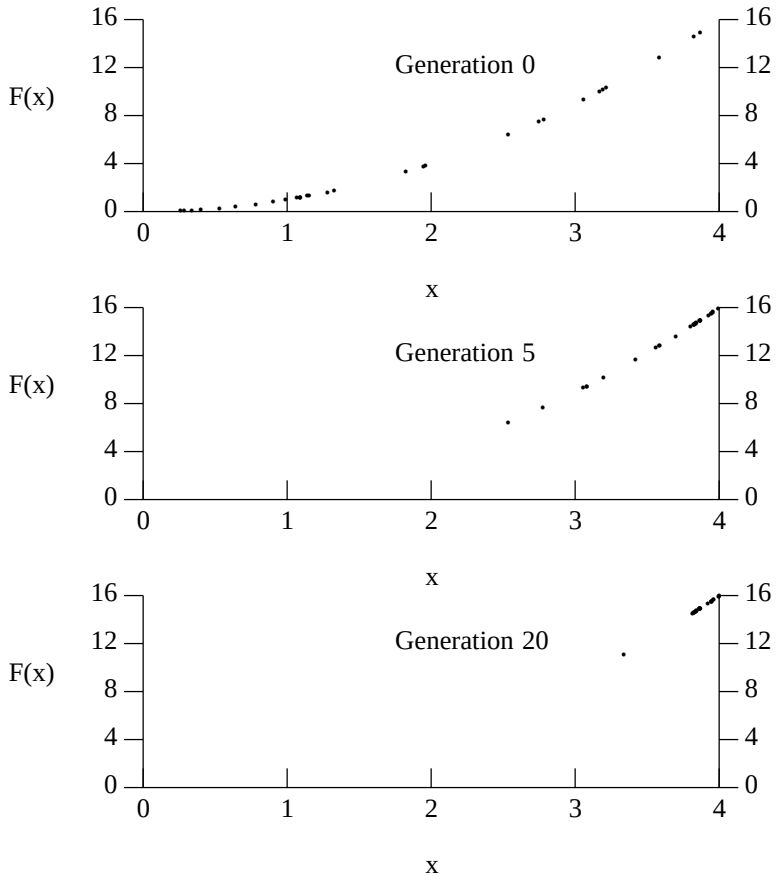

Figure 3: A Phenotypic View of a GA Population

there is no negative feedback for overcrowding. The net result for canonical GAs is a population crowding into one of the higher fitness peaks (see, for example, [DeJong75] or [Goldberg89]). Frequently, the equilibrium point involves the highest fitness peak, but not always.

3.3 AN OPTIMIZATION VIEWPOINT

There are of course other ways of monitoring the behavior of a GA. We might, for example, compute and plot the average fitness of the current population over time, or the average fitness of all of the individuals produced. Alternatively, we might plot the best individual in the current population or the best individual seen so far regardless of whether it is in the current population. In general such graphs look like the ones in Figure 4 and exhibit the properties of "optimization graphs" and "learning curves" seen in many other contexts.

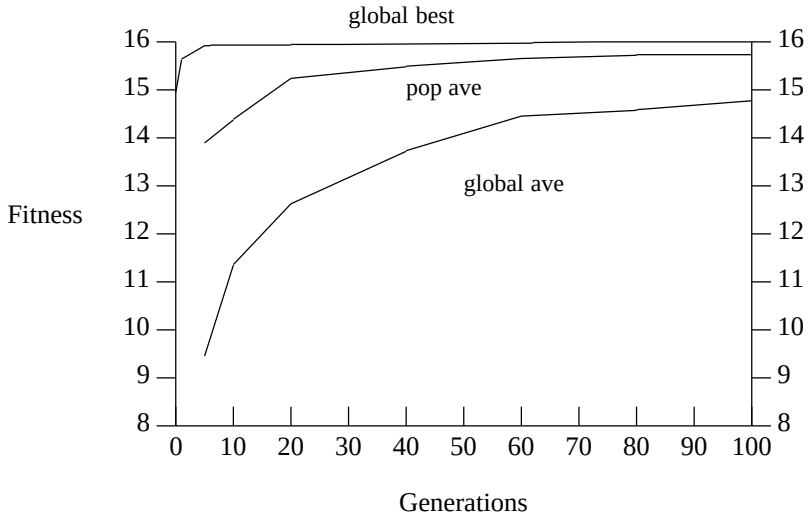

Figure 4: GA Fitness Measures

Measurements of these sorts have encouraged the view of GAs as function optimizers. However, one must be careful not to adopt too simplistic a view of their behavior. The genotypic and phenotypic viewpoints provide ample evidence that populations do not in general converge to dynamic steady states containing multiple copies of the global optimum. In fact it is quite possible for the best individual to never appear or appear and then disappear from the population forever. With finite and relatively small populations, key alleles can be lost along the way due to sampling errors and other random effects such as genetic drift. In theory, as long as mutation is active, one can prove that every point in the space has a non-zero probability of being generated. However, if the population evolves toward subspaces not containing the optimum, the likelihood of generating optimal points via mutations of the current population becomes so small (e.g., $< 10^{-10}$) that such an event is unlikely to be observed in practice.

With a canonical GA every point generated (good or bad) has a relatively short life span since the population is constantly being replaced by new generations of points. Hence, even if a global optimum is generated, it can easily disappear in a generation or two for long periods of time.

These observations suggest that a function optimization interpretation of these canonical GAs is artificial at best and raise the question as to whether there might be a more natural behavioral interpretation.

4 ANALYSIS OF GA BEHAVIOR

Holland [Holland75] has suggested that a better way to view GAs is in terms of optimizing a sequential decision process involving uncertainty in the form of lack of *a priori* knowledge, noisy feedback, and time-varying payoff function. More specifically, given a limited number of trials, how should one allocate them so as to maximize the *cumulative* payoff in the face of all of this uncertainty?

Holland argues that an optimal strategy must maintain a balance between exploitation of the best regions found so far and continued exploration for potentially better payoff areas. Too much exploitation results in local hill climbing, sensitivity to noise, and an inability to adapt to time-varying fitness functions. Too much exploration ignores the valuable feedback information available already early in the search process.

Holland has shown via his Schema Theorem that, given a number of assumptions, GAs are quite robust in producing near-optimal sequences of trials for problems with high levels of uncertainty. Regardless of whether or not one believes that real GA implementations, as opposed to mathematic models of GAs, achieve these near-optimal sequences of trials, I think that Holland's formulation in terms of trying to optimize a sequential decision process in the face of uncertainty is the best behavioral description we currently have for canonical GAs. [1]

5 GAs AS FUNCTION OPTIMIZERS

Once we view a canonical GA as attempting to maximize the cumulative payoff of a sequence of trials, we gain a better understanding of their usefulness as a more traditional function optimization technique. While maximizing cumulative payoff is of interest in some function optimization contexts involving online control of an expensive process, it is much more frequently the case that optimization techniques are compared in terms of the "bottom line" (the best value found), or in terms of the best value found as a function of the amount of effort involved (usually measured in terms of the number of trials).

The standard way of using GAs as function optimizers in this context is to keep track of the (globally) best individual produced so far regardless of whether that individual exists in the current population. That is, the population is viewed as a database (an accumulating world model) of samples from which potentially better individuals are to be generated. Hence, converged alleles represent a "focus of attention" in the sense that most new trials will share these values (i.e., live in the same subspace), while the residual diversity in other genes provides the building blocks for creating new individuals. The genetic operators in conjunction with selection use this focussed diversity to generate new individuals with potentially higher fitness.

If we apply GAs in this way as function optimizers, how well do they perform relative to other techniques? An accumulating body of experience has yielded performances

[1] Recent work by Vose [Vose92] and Whitley [Whitley92] presented elsewhere in this book provide the beginnings of additional behavior insights.

ranging from spectacular to lousy. As a consequence, considerable effort has gone into studying and improving GA-based function optimizers. We summarize a few of these efforts in the following subsections.

5.1 SCALING ISSUES

From a function optimization point of view, GAs frequently don't exhibit a "killer instinct" in the sense that, although they rapidly locate the region in which a global optimum exists, they don't locate the optimum with similar speed. If one looks at this behavior from a "maximizing cumulative returns" point of view, it makes perfect sense. If the range of payoff values is, for example, [0,100], the population evolves quite rapidly to one in which most individuals are in the range [99,100], and selective differential between 99.988 and 100.000 provides little incentive for the GA to prefer one over the other. However, if such differentials are important to the goal of function optimization, then the GA must be provided with a payoff incentive to exploit such differentials.

A typical solution involves providing feedback to the GA in the form of a dynamically scaled fitness function in order to maintain sufficient selective pressure between competing individuals in the current population (see, for example, [DeJong75] or [Goldberg89]). Unfortunately, there is currently no theory to suggest how to achieve an optimal scaling strategy.

5.2 RANKING APPROACHES

An alternative approach is to change the canonical GA itself, substituting rank-proportional selection for the traditional fitness-proportional selection. By keeping the population sorted by fitness and performing selection of the basis of rank, a constant selection differential is maintained between the best and worst individuals in the population (see, for example, [Whitley89]). The effect is to slow down initial convergence to promising subspaces, but to increase the killer instinct in the final stages.

Such approaches can frequently improve the performance of GA-based function optimizers, but depend on appropriate choices of rank-based selection functions (linear, quadratic, exponential, etc.). Unfortunately, currently theory is not strong enough to indicate how general such observations are or to provide guidance as to when to use a particular ranking scheme.

5.3 ELITIST STRATEGIES

Frequently, performance improvements can be obtained by singling out the best and/or worst individuals in the current population for special treatment. Examples of such tactics include always keeping the best individual found so far in the population (it only gets replaced by globally better individuals), or conversely, systematically replacing the worst members of the population with newly generated individuals (e.g., [DeJong75] or [Whitley89]).

The effect is to shift the balance toward more exploitation and less exploration which, for some classes of functions, works quite well. However, if no *a priori* information is available about the functions to be optimized, this can result in suboptimal hill-climbing behavior on multi-peaked functions.

5.4 REPRESENTATIONAL ISSUES

The binary representation chosen in the earlier example is just one of many ways to represent the space to be searched. It was noted very early that the choice of representation can itself affect the performance of a GA-based function optimizer. Suppose, for example the global optimum just happened to be represented by the string 0111111, but from a payoff point of view the subspace represented by 1... was slightly better. In such circumstances, a GA-based optimizer can get caught on a "Hamming cliff" by evolving a population dominated by individuals of the form 1000... which are very close to the optimum in the phenotype space, but far apart in terms of Hamming distance.

Such observations led to alternative representations such as gray codes in an attempt to avoid such problems. Unfortunately, without knowledge about the function to be optimized, it is always possible to have picked a poor representation. Adaptive representation strategies have been suggested as an alternative, but are difficult to efficiently implement.

In the case in which the function arguments are real-valued, one might view a floating point representation as more natural. Again, we have little in the way of theoretical guidance as to whether this will help or hinder the performance of a GA-based function optimizer.

5.5 DOMAIN-SPECIFIC ALTERATIONS

A canonical GA achieves its robustness in part by making no strong assumptions about the properties of the fitness landscape to be explored. If we then compare the performance of a GA-based optimizer with another optimizer on a class of functions for which that optimizer was specifically designed, the GA-based optimizer is frequently outperformed. There are two typical responses to such comparisons. One is to illustrate the robustness of the GA-based optimizer and the brittleness of the specialized one by making minor modifications to the test suite (e.g., add some noise and/or additional peaks).

The other response is to build a specialized GA-based optimizer which takes advantage of this *a priori* knowledge in the form of specialized representations, specialized genetic operators, etc. (e.g., traveling salesperson problems or continuous real-valued functions). This can lead to significant improvements in performance on the selected problems class, but the development process can be time consuming since current theory does not provide strong guidance for making such alterations.

6 SOME FUNDAMENTAL MISCONCEPTIONS

The previous sections have attempted to provide evidence for the following strong claims. First, the canonical GA given in Figure 1 is not a function optimizer in the traditional sense. Rather, it is better viewed as solving sequential decision processes. Second, by modifying a canonical GA in a variety of ways (such as those noted in the previous section) one can build very powerful and effective GA-based optimizers. Third, the strongest theoretical results currently available are for canonical GAs.

Fourth, and finally, there is a tendency to assume that the theoretical and behavioral characteristics of canonical GAs are identical to those of GA-based function optimizers. This is simply not true in general. As we have seen, an effective strategy for solving sequential decision problems does not necessarily also result in an effective function optimization strategy. A GA that satisfies the Schema Theorem does not necessarily make it a good optimizer. Conversely, when we add scaling, introduce ranking and/or elitist strategies, change the way crossover and mutation work, etc., we have significantly changed the behavior of a canonical GA so that there is no *a priori* reason to believe that it continues to solve sequential decision problems effectively. In fact, many of the strong sampling biases introduced into GA-based optimizers are known to negatively affect cumulative (online) payoff.

This last point leads to some fairly fundamental misconceptions about the meaning and implications of some of the current work in the field regarding understanding what makes problems hard (easy) for GAs to solve. [2] Although not clearly stated, the motivating context for characterizing GA-hard and GA-easy problems is function optimization. So to be precise we should be talking about classes of functions which are hard (easy) for GA-based optimizers to solve. The important point here is that GA-hard optimization problems may or may not represent hard sequential decision problems. In fact the notion of GA hard (easy) is much less well defined for canonical GAs solving sequential decision problems. Let me illustrate these points with two examples.

6.1 DECEPTIVE PROBLEMS

Since canonical GAs use sampled averages to bias the direction in which future individuals are generated, it is easy to construct fitness landscapes which "deceive" a GA-based optimizer by hiding the global optimum in a part of the landscape with low average payoff, thus making it highly unlikely to be found (see, for example, [Goldberg89] or [Whitley91]). What is important to understand here is that there is no strong sense of deception from a canonical GA's point of view. It is attempting to maximize *cumulative* payoff from arbitrary landscapes, deceptive or otherwise. In general, this is achieved by not investing too much effort in finding cleverly hidden peaks (the risk/reward ratio is too high).

On the other hand, since most interesting optimization problems are NP-hard, we might reasonably drop our demand (expectation) that a GA-based optimizer find the global

[2] See [Grefenstette92] elsewhere in this book for a related discussion of misconceptions.

optimum and think more in terms of using it as a heuristic that finds good values quickly. Notice that in this case deception becomes less of a concern since GA-based optimizers frequently turn out to be excellent heuristics for most deceptive problems.

This is not to say that the work on deception is irrelevant or incorrect. Rather, it is a step in a direction badly needed: theoretical results for GA-based optimizers (as opposed to canonical GAs). Without such work we are left in the uncomfortable position of assuming "what's good for the goose (canonical GAs) is good for the gander (GA-based optimizers)".

6.2 BUILDING BLOCK HYPOTHESES

The theoretical analysis that we have for canonical GAs suggests a view in which selection, mutation, and crossover work synergistically to construct new individuals from an evolving set of useful building blocks represented by the genetic makeup of the individuals in the population. It is easy to show that deception, finite populations and sampling errors can result in the loss of important low level building blocks which can, in turn, negatively affect future "construction projects".

Suppose, however, we select landscapes which have all the right properties with respect to encouraging the formation of useful building blocks. Are we correct in assuming that these landscapes are necessarily GA-easy from a function optimization point of view in that the global optimum is easily found? In general, the answer is no. Rather than view such results as "anomalous" (e.g., [Forrest91]), we should expect this once we understand that canonical GAs are not optimizers. Moreover, the addition of scaling, elitism, and other optimization-motivated features to a canonical GA can result in improved optimization behavior, but can also sufficiently distort and bias the sampling so as to make some of these "easy" landscapes appear hard.

7 SUMMARY AND CONCLUSIONS

The intent of this paper is not to question the usefulness of GAs as function optimization techniques. In fact, the intent is quite the opposite. GA-based function optimizers have already demonstrated their usefulness over a wide range of difficult problems. At the same time, there are obvious opportunities for improvements and extensions to new function classes. In order to achieve this we need to understand better how GA-based optimizers work.

The concern expressed here is that there is a tendency to think of GAs as function optimizers, and thus blur the distinction between a canonical GA and a GA which has been modified for use as an optimizer. By blurring that distinction, it is easy to mistakenly assume that what we know and understand about canonical GAs carries over to the GA-based optimizers. It is certainly the case that both canonical GAs and GA-based optimizers share many properties. However, it is also quite clear that they are quite different in some important respects. In order to improve our GA-based optimizers we must continue to clarify both the similarities and the differences.

REFERENCES

Bethke, Albert D. (1981). *Genetic Algorithms as Function Optimizers*, Doctoral Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor.

De Jong, Kenneth A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor.

Forrest, S. and Mitchell, M. (1991). The Performance of Genetic Algorithms on Walsh Polynomials: Some Anomalous Results and their Explanation. *Proc. 4th Intl. Conf. on Genetic Algorithms*, La Jolla, CA: Morgan Kaufmann.

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

Grefenstette, J. (1992). Deception Considered Harmful. In *Foundations of Genetic Algorithms 2*,, D. Whitley (ed.), Vail, CO: Morgan Kaufmann.

Holland, John H. (1975). *Adaptation in Natural and Artificial Systems*, The University of Michigan Press.

Vose, M. (1992). Modeling Simple Genetic Algorithms. In *Foundations of Genetic Algorithms 2*,, D. Whitley (ed.), Vail, CO: Morgan Kaufmann.

Whitley, D. (1989). The Genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, *Proc. 3rd Intl. Conf. on Genetic Algorithms*, Fairfax, VA: Morgan Kaufmann.

Whitley, D. (1991). Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*, G. Rawlins (ed.), Bloomington, IN: Morgan Kaufmann.

Whitley, D. (1992). An Executable Model of a Simple Genetic Algorithm. In *Foundations of Genetic Algorithms 2*,, D. Whitley (ed.), Vail, CO: Morgan Kaufmann.