Deception Considered Harmful *

John J. Grefenstette
Navy Center for Applied Research in Artificial Intelligence
Code 5514
Naval Research Laboratory
Washington, DC 20375-5000
E-mail: GREF@AIC.NRL.NAVY.MIL

Abstract

A central problem in the theory of genetic algorithms is the characterization of problems that are difficult for GAs to optimize. Many attempts to characterize such problems focus on the notion of *Deception*, defined in terms of the static average fitness of competing schemas. This article examines the Static Building Block Hypothesis (SBBH), the underlying assumption used to define Deception. Exploiting contradictions between the SBBH and the Schema Theorem, we show that Deception is neither necessary nor sufficient for problems to be difficult for GAs. This article argues that the characterization of hard problems must take into account the basic features of genetic algorithms, especially their dynamic, biased sampling strategy.

Keywords: Deception, building block hypothesis

1 INTRODUCTION

Since Holland's early work on the analysis of genetic algorithms (GAs), the usual approach has been to focus on the allocation of search effort to subspaces described by schemas representing hyperplanes of the search space. The Schema Theorem (Holland, 1975) provides a description for the growth rate of schemas that depends on the observed relative fitness of the schemas represented in the population. Bethke (1981) initiated work on the formal characterization of problems that might be difficult for GAs to solve,

* With apologies to Edsger Dijkstra (Dijkstra, 1968).

and presented an analysis of problems in terms of the Walsh transformation of the fitness function. Goldberg (1987) introduced the notion of *Deception* in GAs, and defined a Minimal Deceptive Problem (MDP). Subsequently, Deception has come to be widely regarded as a central feature in the design of problems that are difficult for GAs (Das and Whitley, 1991; Homaifar, Qi and Fost, 1991). Goldberg and his colleagues (Goldberg, Korb, and Deb, 1989) have defined messy GAs (mGAs) specifically to handle Deceptive problems, and consider the use of Deceptive functions as test functions to be "critical to understanding the convergence of mGAs, traditional GAs, or any other similarity-based search technique". The literature on Deception in GAs is growing rapidly (Battle & Vose, 1991; Davidor, 1990; Deb and Goldberg, 1992; Goldberg 1989a, 1989b, 1989c, 1991, 1992; Goldberg, Deb and Korb, 1990, 1991; Goldberg, Deb and Clark, 1992; Liepins & Vose, 1990, 1991; Mason, 1991; Whitley, 1991, 1992), so this is clearly a topic that deserves careful scrutiny.

In previous papers (Grefenstette and Baker, 1989; Grefenstette, 1991) we have raised some questions about this approach to the analysis of GAs, and others have begun to express similar concerns (Forrest and Mitchell, 1993). This paper will try to clarify and expand on the argument that the current definitions of Deception appear to be based on faulty assumptions about the dynamics of GAs. This paper only addresses definitions of Deception that are based on the static analysis of hyperplanes. By static analysis, we mean the analysis based on the average fitness of hyperplanes, when the average is taken over the entire search space. Our fundamental point is that the dynamic behavior of genetic algorithms cannot be predicted in general on the basis of the static analysis of hyperplanes. This paper does *not* argue against the notion that there are classes of problems that are "deceptive", in the sense that properties of the response surface lead the GA away from the optimal regions of the search space. It is clearly important to try to characterize such problems, but any useful characterization will have to be based on correct assumptions about genetic algorithms.

The remainder of the paper is organized as follows: Section 2 discusses the Static Building Block Hypothesis (SBBH) that appears to underlie much of the work on static hyperplane analysis. Section 3 shows that some functions that are highly Deceptive according to the SBBH are, in fact, very easy for GAs to optimize. Section 4 shows that some functions that have no Deception and therefore should be easy, according to the SBBH, are nearly impossible for GAs to optimize. These counterexamples show that Deception is neither necessary nor sufficient to make a problem difficult for GAs. More importantly, the analysis of these results highlights the shortcomings of the static analysis of hyperplanes. As an aside to our main point, Section 5 shows that, even if the SBBH were true, many Deceptive functions could be easily solved by simple changes to the basic GA. Section 6 summarizes the paper.

2 THE STATIC BUILDING BLOCK HYPOTHESIS

The fundamental description of the dynamics of genetic algorithms is the Schema Theorem (Holland, 1975), which describes the relationship between the expected growth in a hyperplane from one generation to the next as a function of the hyperplane's observed relative fitness:

Schema Theorem.

$$M(H,t+1) \geq M(H,t)\left(\frac{f(H,t)}{\bar{f}(t)}\right)(1-p_d(H,t))$$

where $M(H,t)$ is the expected number of samples of a hyperplane H in the population at time t, $f(H,t)/\bar{f}(t)$ is the *observed* relative fitness of H at time t, and $p_d(H,t)$ is the probability that H will be disrupted by the genetic operators such as crossover and mutation. The Schema Theorem is only directly applicable to a single generational cycle, but one can get an intuitive feel for the dynamics of GAs by considering what happens to a hyperplane that consistently has an observed fitness that is higher than the population average. This case is often described as follows:

> "The usual interpretation of [the Schema Theorem] is that subspaces with higher than average payoffs will be allocated exponentially more trials over time, while those subspaces with below average payoffs will be allocated exponentially fewer trials. This assumes that ... the effects of crossover and mutation are not disruptive." (Spears and De Jong, 1991).

Strictly speaking, the word *observed* should be inserted before each occurrence of *payoff*, but the intended meaning is usually clear and the shorthand phrase is often more convenient to use. Many even more reckless statements appear in the introductory sections of many articles on genetic algorithms, for example:

> "[The Schema Theorem] says that the number of samples allocated to an above-average hyperplane *H* grows exponentially over time." (Grefenstette, 1990)

Sometimes, as in the case above, such generalizations are qualified at some later points in the article. Using the same form of shorthand, the overall dynamics of GAs are often expressed in the form of the so-called "Building Block Hypothesis":

> "[A] genetic algorithm seek[s] near optimal performance through the juxtaposition of short, low-order, high performance schemata, or building blocks." (Goldberg 1989a, p. 41)

The Building Block Hypothesis is a rough but serviceable first explanation of how GAs operate, and has been used by most of us when explaining GAs to newcomers. There is usually little harm in informal statements about the dynamics of GAs, but they can lead to rather more serious misinterpretations if they are taken as the basis for an operational theory for GAs. This paper will focus on the implications of one operational version of the Building Block Hypothesis, which we call the:

> *Static Building Block Hypothesis* (SBBH): Given any short, low-order hyperplane partition, a GA is expected to converge to the hyperplane with the best static average fitness (the "expected winner").

For example, consider the following 2nd-order hyperplane partition.

		$f(H)$
H_a:	00######	1
H_b:	01######	3
H_c:	10######	10
H_d:	11######	7

Here, and throughout this paper, the value $f(H)$ refers to the static average fitness for schema H, that is, the mean fitness value of every point described by that schema. This *static* average is independent of whatever points happen to be in the population at any time. According to the SBBH, the "expected winner" of the above schema competition is the hyperplane H_c.

The SBBH implies that functions for which the low-order schemas associated with the optimum have higher static average fitness than the competing schemas in their partitions ought to be easy for GAs. For example, suppose the global optimum is 00...0, and that

$$
\begin{aligned}
f(0\#...\#) &> f(1\#...\#) \\
f(00\#...\#) &> f(01\#...\#) \\
f(00\#...\#) &> f(10\#...\#) \\
f(00\#...\#) &> f(11\#...\#)
\end{aligned}
$$

and so on for every hyperplane partition of the search space. According to the SBBH, a GA should find f simple to optimize. In fact, such functions are commonly called "GA-easy" (Wilson, 1991).[1]

Conversely, the SBBH implies that functions for which which the low-order schemas associated with the optimum have lower static average fitness than the competing schemas in their partitions ought to be difficult for GAs. For example, suppose the global optimum is 00...0, and that

---

[1] We are not asserting that Wilson subscribes to the SBBH.

$$
\begin{aligned}
f(0\#...\#) &< f(1\#...\#) \\
f(00\#...\#) &< f(01\#...\#) \\
f(00\#...\#) &< f(10\#...\#) \\
f(00\#...\#) &< f(11\#...\#)
\end{aligned}
$$

and so on. Then this function would be called *Deceptive* (Goldberg, 1987).

The SBBH appears to underlie much of the recent published work in GA theory, especially work on Deception. The SBBH itself is rarely cited as an assumption in the Deception literature. Instead, the Schema Theorem is usually invoked, along with an informal statement of the building block hypothesis. For example,

> "Let's construct the simplest problem that should cause a GA to diverge from the global optimum ... To do this, we want to violate the building block hypothesis in the extreme." (Goldberg introducing the MDP, 1989a)

or

> "It follows from the Schema Theorem that the number of instances of a schema is expected to increase in the next generation if it is of above average utility and is not disrupted by crossover. Therefore, such schemata indicate the area within the search space that the GA explores, and hence it is important that, at some stage, these schema contain the object of search. Problems for which this is not true are called *deceptive*." (Liepins and Vose, 1991)

or

> "It has been shown that genetic algorithms work well when building blocks, short, low-order ... schemata with above average fitness values, combine to form optimal or near-optimal solutions." (Homaifar et. al, 1991)

When such statements are followed by the analysis of static hyperplane averages, as they are in all the above cases, it seems to be fair to ascribe the SBBH as an implicit assumption.

While the SBBH is an appealing intuitive explanation for how GAs work, it has never been explicitly proven, and it does not follow from the Schema Theorem. The Schema Theorem describes the expected growth of a hyperplane for a single generation based on its *observed* average fitness, that is, based on the average fitness of current samples of the hyperplane in the population. Over a period of generations, the observed average fitness of a hyperplane does not necessarily reflect the static average fitness of the hyperplane. The SBBH arises when we ignore the crucial distinction between observed average fitness and static average fitness. In particular, we get the SBBH if we drop the time index t from the term $f(H,t)/\overline{f}(t)$ in the Schema Theorem. The SBBH, then, is merely an approximation to the dynamics described by the Schema Theorem.

There are many ways in which such an approximation might be expected to diverge from the dynamics described by the full Schema Theorem. We will examine just two ways in which the SBBH fails to account for the true dynamics of GAs:

1. Failure to account for collateral convergence.
2. Failure to account for fitness variance within schemas.

The effect of collateral convergence is that, once the population begins to converge, even a little, it is no longer possible to estimate the static average fitness of schemas using the information present in the current population. The effect of fitness variance within schemas is that, in populations of realistic size, the observed fitness of a schema may be arbitrarily far from the static average fitness, even in the initial population. In the next two sections, the reasons listed above are used to show why the Static Building Block Hypothesis, and the accompanying notion of Deception, cannot be used to predict how difficult a function may or may not be for a GA to optimize.

3 COLLATERAL CONVERGENCE

The primary reason that the SBBH is a poor model of GAs is that, except possibly for the very first generation, the population contains only a biased sample of representatives from each schema. This is a normal feature of all GAs, and is true regardless of population size, but it can yield results that are exactly the opposite of what one might expect from the SBBH. A simple thought experiment will bring the point home. Consider the following two first-order schema competitions:

		$f(H)$
H_A:	0#...###...	9
H_B:	1#...###...	1
H_C:	##...#0#...	6
H_D:	##...#1#...	4

Assume that the initial population is selected using a uniform distribution, and that the population is sufficiently large that, for each schema, the observed fitness at time $t = 0$ approximates the schema's static average fitness. Then, the Schema Theorem predicts that the expected allocation of trials in the second generation is as follows:

		Pop % in H at time t =1
H_A:	0#...###...	90
H_B:	1#...###...	10
H_C:	##...#0#...	60
H_D:	##...#1#...	40

Note that the Schema Theorem predicts that the population will begin to converge more rapidly with respect to the first competition than with respect to the second competition. Now, can we predict how these two first-order competitions will proceed at time t = 1? The answer is no, since we no longer have sufficient information to estimate what the *observed fitnesses* of the competing schema might be. The *collateral convergence* of the population[2] toward H_A will bias the samples taken from H_C and H_D at time t = 1. For example, we would expect that 90% the observed representatives of H_C would have a 0 in the first position. The static average fitness does not indicate what the observed fitness of this highly biased sample might be. Taking this collateral convergence into account, we cannot predict, on the basis of the static average fitness of the low-order hyperplanes, whether H_C will continue to grow more rapidly than H_D at time t = 1. We have even less reason to predict whether the GA will ultimately converge to H_C or H_D, and yet this is what the SBBH claims to do.

To see how collateral convergence can affect the study of Deception, consider the following problem:

$$\max \quad f(x_1, x_2), \text{ where } 0 \le x_i \le 1, \text{ and}$$

$$f(x_1, x_2) = \begin{cases} x_1^2 + 10x_2^2 & \text{if } x_2 < 0.995 \\ 2(1-x_1)^2 + 10x_2^2 & \text{if } x_2 \ge 0.995 \end{cases}$$

Assume that x_1 and x_2 are represented in binary notation on a chromosome of length 20, using 10 bits for each argument. The optimal solution is $(x_1, x_2) = (0, 1)$.

Using the terminology in (Whitley, 1991), this problem has a "fully deceptive subproblem of order-10", meaning that in every schema competition defined over the first 10 bit positions, the "expected winner" of each such competition is the schema containing the "deceptive attractor" 1111111111##########, corresponding to the hyperplane $(x_1, x_2) = (1, \#)$. Despite this high level of Deception[3] a standard GA[4] has

[2] *Collateral convergence* refers to the phenomenon that the population converges at different rates to two intersecting hyperplanes, e.g., H_A and H_C, above.

[3] This problem can be made arbitrarily more Deceptive by replicating the gene x_1 in the above example.

[4] GENESIS with population size 200, other parameters set to default values.

no trouble finding the global optimum within a few thousand trials in 10 out of 10 runs. The success of the GA can be predicted by the Schema Theorem, which says that the second 10 bits will rapidly converge toward the value of $x_2 = 1$. Once this collateral convergence occurs, the *observed* fitness of schemas representing values of x_1 near 0 will change from low to high, and the GA will converge toward the global optimum.

Figure 1 illustrates some of the dynamics of the GA on this problem, averaged over 10 independent runs. The graph show the progress of the competition between the schemas:

$$H_0\text{: } 0\#...\#$$
$$H_1\text{: } 1\#...\#$$

by indicating the per cent of the population in H_0. The dotted line indicates the convergence of the second 10 bits (x_2). Since $f(H_1) > f(H_0)$, the SBBH predicts convergence to H_1. For $t < 15$, the *observed* fitnesses of the competing schemas agree with the SBBH, that is, $f(H_1, t) > f(H_0, t)$, and the allocation of trials to H_0 declines accordingly. By about generation 20, the value of x_2 has largely converged, and the GA now observes that $f(H_1, 20) < f(H_0, 20)$, thanks to the effects of the second term in the fitness function. In accordance with the Schema Theorem, and in contradiction to the SBBH, the allocation of trials shifts to H_0 and away from H_1.

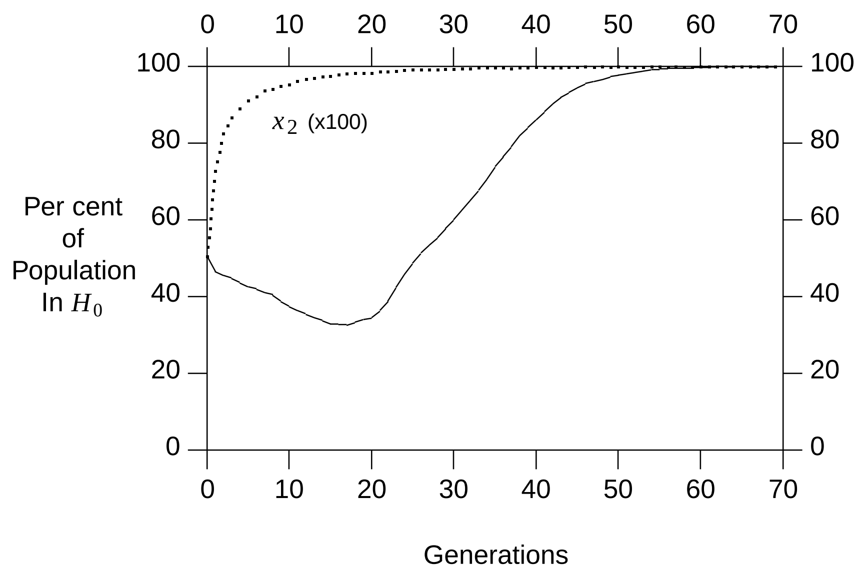

Figure 1: The Schema Theorem in Action.

As this example shows, the ultimate allocation of trials across a given hyperplane partition can be difficult to predict, precisely because the GA makes its allocation decisions based upon the current estimate of payoff associated with hyperplanes. These current estimates are highly influenced by collateral convergence. In general, there is no simple relationship between the observed payoff $f(H,t)$ at time t and the static average payoff $f(H)$. This example shows that some highly Deceptive problems are in fact easy for GAs to optimize. The underlying reason is that the SBBH simply does not model the dynamic behavior of GAs, as described by the Schema Theorem.

It may be thought that having a large population might allow us to ignore the effects of collateral convergence, and use the SBBH as a first approximation to the Schema Theorem. However, the effects of collateral convergence do not really depend on having a small population size. In fact, the thought experiment at the beginning of this section can be applied to an infinite population model. To give a practical example, we repeated the experiment shown in Figure 1, using a population of size 10,000 (instead of 200). The results are shown in Figure 2.

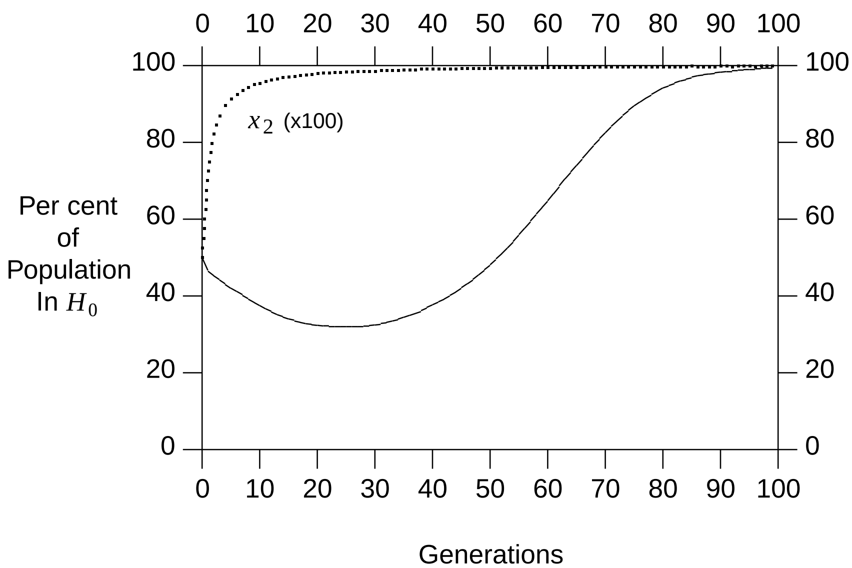

Figure 2: The Schema Theorem in Action with Population Size 10,000

As expected, the convergence is a little slower, since the exponential allocation of trials implies that the time to converge to high values of x_2 increases with the logarithm of the population size. Nevertheless, the overall performance profile is very similar to the small population case. If our analysis of collateral convergence is correct, the results predicted by the SBBH should not be expected for any size population.

4 LARGE VARIANCE WITHIN SCHEMAS

A second shortcoming of the Static Building Block Hypothesis is that it fails to account for the variance of fitness within schemas. With a limited population size and large variance within the schemas, the sampling in the initial, random population will produce errors in the estimate of each schema's static average fitness. This schema variance can lead to results that contradict the SBBH.

To illustrate this, we can define a class of problems that are "easy" in the sense implied by the SBBH -- they have no Deception -- but are, in fact, nearly impossible for GAs to optimize. Consider an L-bit space representing the interval [0, 1] in binary encoding. Let f be defined:

$$f(x) = \begin{cases} 2^{(L+1)} & \text{if } x = 0 \\ x^2 & \text{otherwise.} \end{cases}$$

For any schema H such that the optimum is in H (that is, all the defined positions of H have value 0), $f(H) > 2$, since the sum of the fitness of the points in H is at least $2^{(L+1)}$ and there are at most 2^L points in the hyperplane. For any schema H such that the optimum is not H, $f(S) \leq 1$. So in any schema partition, the schema containing the optimum has the highest static average fitness. That is, there is no Deception at any level in the function. Such functions are often called "GA-easy" (Liepins and Vose, 1991; Wilson, 1991).

Suppose we run a standard GA on f with a population of size polynomial in L. If the optimum is not in the initial population, it will probably never be found. (Of course, it might be created by a lucky crossover or a very lucky multiple mutation.) Why is this function hard for GAs? Because the schemas associated with the optimum have extremely high variance, so the *observed* average fitness for the hyperplanes never reflects their *static* average fitnesses, not even in the initial random population. Of course, this is a "needle-in-a-haystack" function, so we don't expect the GA to solve it on a regular basis. But it does satisfy the commonly used definition of "GA-easy" that follows from the SBBH, so this example provides a counterexample to the claim that only Deceptive problems are challenging for GAs (Das and Whitley, 1991).

There has been some recent work that addresses the (static) fitness variance within schemas (Goldberg and Rudnick, 1988; Rudnick and Goldberg, 1991). This is a step in the right direction, but it is unlikely that an analysis of the static fitness variance will be much more helpful than an analysis of the static fitness averages, for the same reasons as in the previous Section. As the search proceeds, the *observed* variance associated with hyperplanes in the population is unlikely to have any correlation with the *static* variance.

5 AUGMENTED GAs FOR DECEPTIVE PROBLEMS

There is a growing literature on how to make GAs more effective on Deceptive problems. For example, Liepins and Vose (1991) specify representation transformations that render Fully Deceptive problems "fully easy". Goldberg at. al (1991) define "messy GAs" in order to deal with problems of bounded Deception. In this section, we suggest that, assuming that the SBBH applies to GAs, slight changes to the basic GA would be sufficient to solve Deceptive problems.

Much of the work on Deception involves functions for which the bit-wise complement of the global optimum is the Deceptive attractor (Liepins and Vose, 1991; Whitley, 1991). In fact, arguments have been made that all Fully Deceptive problems have this feature (Whitley, 1992). For the purpose of this section, let us accept this argument and suppose that a GA would actually perform according to the SBBH on these fully Deceptive problems. That is, suppose the GA really does converge to the complement of the global optimum. Then one algorithm that finds the global optimum in all such problems is simply:

1. Run a GA until it converges to some string, *x*
2. Output either *x* or the complement of *x*, whichever is better.

At a cost of a single extra evaluation, this algorithm strictly extends the class of problems that can be optimized by GAs to include all Fully Deceptive problems.

However, perhaps the more usual cases are "partially deceptive" problems, that is, problems that have a Deceptive component and a non-Deceptive component. Such problems might be handled by the augmented GA shown in Figure 3. The lines marked with (*) represent the changes to a standard generational GA. In the augmented version, we maintain three separate populations of size *N*, called *P*, *Q* and *R*. During each generation, we update P according to the original GA, set the members of *Q* to the bitwise complement of the corresponding elements of *P*, and create *R* by crossing over randomly selected parents from *P* and *Q*.

Consider any problem for which the original GA (i.e., the one without the (*) lines) finds an acceptable solution in time *t*, using a population of size *N*. Then the augmented algorithm finds a solution that is at least as good as the original GA, using at most time 3*t* (assuming that the evaluation time dominates the other operations in the algorithms). Like the simpler variant, this augmented algorithm solves any fully Deceptive problem that has the property that the global optimum is the binary complement of the Deceptive attractor, since as soon as the population *P* produces a copy of the Deceptive attractor, the population *Q* produces a copy of the global optimum.

```
procedure Augmented GA
begin
        t = 0;
        initialize P(t);
        evaluate structures in P(t);
        while termination condition not satisfied do
        begin
                t = t + 1;
                select P(t) from P(t-1);
                recombine structures in P(t);
(*)             Q(t) = complement(P(t));
(*)             form R(t) by recombining parents from P(t) and Q(t);
                evaluate structures in P(t);
(*)             evaluate structures in Q(t);
(*)             evaluate structures in R(t);
                output best structure in P(t) ∪ Q(t) ∪ R(t);
        end
end.
```

Figure 3: An Augmented Genetic Algorithm.

By allowing recombination between randomly selected members of populations *P* and *Q*, the SBBH predicts that we could generate optimal solution to partially deceptive problems as well. For any component that is not deceptive, the elements in *P* should converge to the correct component values, according to the SBBH. The deceptive components should converge to the complement of the optimal component values. Thus the complements of the deceptive components -- the optimal component values -- will be stored in population *Q*. Performing (multi-point) crossover across populations *P* and *Q* should eventually produce the optimal components all along the chromosome in population *R*. Thus the augmented GA can produce a final answer that is never worse than the one produced by the original GA, and it eliminates concern about Deceptive problems, all at a cost of only tripling the computational time.

One might conclude from this discussion is that Deception, as currently defined, is not a serious problem for GAs, since it can be handled by small extensions to the original algorithm. Unfortunately, this discussion is purely academic, since the proposed "solutions", like the notion of Deception itself, are based on the SBBH, and are therefore unlikely to provide useful results for real GAs.

6 SUMMARY

This paper criticizes the Static Building Block Hypothesis as a description of the dynamics of GAs. The SBBH arises by ignoring an important distinction made in the Schema Theorem, between the *observed* fitness of a hyperplane and the *static* fitness of a hyperplane. We have identified two reasons that cause divergent predictions between the SBBH and the Schema Theorem:

1. Failure to account for collateral convergence.
2. Failure to account for fitness variance within schemas.

According to the SBBH, Deceptive problems ought to be difficult for GAs to solve, and "GA-easy" problems ought to be easy for GAs to solve. Taking the above differences between the Schema Theorem and the SBBH into account, it is easy to demonstrate that:

1. Some highly Deceptive problems are easy for GAs to optimize.
2. Some "GA-easy" problems with no Deception are nearly impossible for GAs to optimize.

That is, Deception is neither necessary nor sufficient for causing difficulties for GAs. Put another way, the class of Deceptive functions is neither a subset nor a superset of the class of functions that are hard for GAs to optimize, as shown in Figure 4. At the very least, the term *Deceptive* seems to be poorly chosen. More importantly, our arguments show that it is in general impossible to predict the dynamic behaviors of GAs on the basis of the static average fitness of hyperplanes.

Even though the presence or absence of Deception provides no logical implication of problem difficulty, there may be some correlation between Deception and difficulty for GAs. Although we have no way at present to measure such a correlation, it is true that *some* Deceptive problems are difficult for GAs, and *some* "GA-Easy" problems are in fact easy for GAs.[5] An interesting problem would be to characterize *which* Deceptive problems actually cause difficulties for GAs. An approach to this question might involve the design of problems that take advantage of the different rates at which collateral schema competitions converge, as predicted by the Schema Theorem. It seems doubtful that a successful approach will be based on the SBBH.

One might also argue that the work to date that is based on the SBBH has been a preliminary exploration of how to analyze simple distributions of fitness, and was always intended to be replaced by dynamic analysis. In this case, articles using static analysis should qualify their conclusions accordingly, rather than make what appear to be strong

[5] For example, the Partially Deceptive problem defined in Section 3 can be made difficult for GAs, as well as Deceptive, if we negate the coefficient on x_2, so that x_2 converges to 0 rather than to 1.

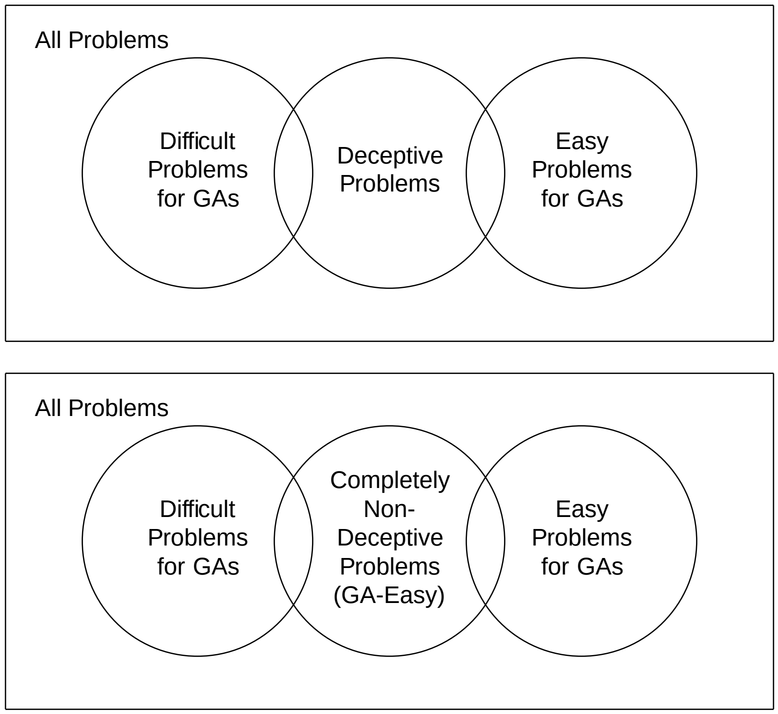

Figure 4: Deception and Problem Difficulty.

statements about GAs and Deception, for example:

> "The use of deceptive functions as test functions is critical to understanding the convergence of any GAs or other similarity-based search technique ... Since these functions are maximally misleading, if an algorithm can solve this class of problem, it can solve anything easier." (Goldberg, Deb, and Korb, 1990)

> "The only challenging problems are deceptive ... Test problems must be used that involve some degree of deception." (Whitley, 1991)

Such conclusions properly apply only to algorithms that satisfy the SBBH. It is not clear that there exist any practical algorithms in this class, but in any case the GA is not one of them.

Some other concerns about the notion of Deception have been raised by others, including Forrest and Mitchell (1993), and De Jong (1992). In particular, measuring GA performance by the ability to find the global optimum seems at odds with the emphasis in (Holland, 1975) on exploiting local information for adaptation. Both De Jong (1975) and Bethke (1981) have proposed other performance metrics that are more appropriate for measuring the effectiveness of adaptive search methods.

It goes without saying that the characterization of hard problems should remain a high priority for the GA research community. However, the characterization must take into account the basic features of the GA, especially its dynamic, biased sampling strategy. Our recommendation is that the efforts currently being expended on the static analysis of functions should be diverted to the dynamic analysis of GAs. It is gratifying to see some recent efforts in this direction (Bridges and Goldberg, 1991; Liepins and Vose, 1991; Nix and Vose, 1992). Nevertheless, it is important that articles on GA theory avoid the implicit assumption of the SBBH. The SBBH is a seductive, but inaccurate, explanation for the power of the GA that can easily mislead (deceive?) newcomers to the field, as well as potential users of the technology.

Acknowledgements

Thanks to Ken De Jong, Bill Spears, and the reviewers for helpful suggestions in improving the presentation of this paper.

REFERENCES

Battle, D. L. and M. D. Vose (1991). Isomorphisms of genetic algorithms. In *Foundations of Genetic Algorithms*, G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Bethke, A. D. (1981). Genetic algorithms as function optimizers. Doctoral dissertation, University of Michigan.

Bridges, C. L. and D. E. Goldberg (1991). The nonuniform Walsh-schema transform. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Das, R. and D. L. Whitley (1991). The only challenging problems are deceptive. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 116-173). San Mateo, CA: Morgan Kaufmann.

Davidor, Y. (1990). Epistasis variance: suitability of a representation to genetic algorithms. *Complex Systems 4(4)*, 369-384.

De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems.* Doctoral dissertation, Department of Computer and Communications Sciences, University of Michigan, Ann Arbor.

De Jong, K. A. (1992). Genetic algorithms are NOT function optimizers. In *Foundations of Genetic Algorithms 2,* D. Whitley (Ed.), San Mateo: Morgan

Kaufmann.

Deb, K. and D. E. Goldberg (1992). Analysing deception in trap functions. In *Foundations of Genetic Algorithms 2*, D. Whitley (Ed.), San Mateo: Morgan Kaufmann.

Dijkstra, E. W. (1968). Go to statement considered harmful. *CACM, 11(3),* 147-148.

Goldberg, D. E. (1987). Simple genetic algorithms and the minimal deceptive problem. In L. Davis (Ed.), *Genetic algorithms and simulated annealing* (pp. 74-88). London: Pitman.

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

Goldberg, D. E. (1989b). Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems, 3,* 129-152.

Goldberg, D. E. (1989c). Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems, 3,* 153-171.

Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems, 5,* 139-167.

Goldberg, D. E. (1992). Construction of high-order deceptive functions using low-order Walsh coefficients. *Annals of Mathematics and Artificial Intelligence, 5,* 35-48.

Goldberg, D. E., K. Deb and B. Korb (1990). Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems, 4,* 415-444.

Goldberg, D. E., K. Deb and B. Korb (1991). Don’t worry, be messy. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 24-30). San Mateo, CA: Morgan Kaufmann.

Goldberg, D. E., K. Deb and J. Clark (1992). Accounting for noise in the sizing of populations. In *Foundations of Genetic Algorithms 2,* D. Whitley (Ed.), San Mateo: Morgan Kaufmann.

Goldberg, D. E., B. Korb and K. Deb (1989). Messy genetic algorithms: Motivation, analysis and first results. *Complex Systems, 3,* 493-530.

Goldberg, D. E. and M. Rudnick (1988). Genetic algorithms and the variance of fitness. *Complex Systems, 2,* 265-278.

Grefenstette, J. J. (1990). Genetic algorithms and their applications. In **The Encyclopedia of Computer Science and Technology, Vol. 21,** A. Kent and J. G. Williams (Eds.), New York: Marcel Dekker.

Grefenstette, J. J. (1991). Conditions for implicit parallelism. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Grefenstette, J. J. and J. E. Baker (1989). How genetic algorithms work: An critical look at implicit parallelism. *Proceedings of the Third International Conference of Genetic Algorithms* (pp. 20-27). San Mateo, CA: Morgan Kaufmann.

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

Homaifar, A., X. Qi and J. Fost (1991). Analysis and design of a general GA deceptive problem. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 196-203). San Mateo, CA: Morgan Kaufmann.

Liepins, G. E. and M. D. Vose (1990). Representational issues in genetic algorithms. *J. Exp. Theor. Artificial Intelligence,* 4-30.

Liepins, G. E. and M. D. Vose (1991). Deceptiveness and genetic algorithm dynamics. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Mason, A. J. (1991). Partition coefficients, static deception and deceptive problems for non-binary alphabets. *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 210-214). San Mateo, CA: Morgan Kaufmann.

S. Forrest and M. Mitchell (1993). What makes a problem hard for a genetic algorithms? Some anomalous results and their explanation. To appear in *Machine Learning.*

Nix, A. E. and M. D. Vose (1992). Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence, 5,* 79-88.

Rudnick, M. and D. E. Goldberg (1991). Signal, noise, and genetic algorithms. IlliGAL Report No. 91004, Univ. Illinois at Urbana-Champaign. Also available as Oregon Graduate Institute Technical Report No CS/91-013.

Spears, W. M. and K. A. De Jong (1991). An analysis of multi-point crossover. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Whitley, L. D. (1991). Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms,* G. J. E. Rawlins (Ed.), San Mateo, CA: Morgan Kaufmann.

Whitley, L. D. (1992). Deception, dominance and implicit parallelism in genetic search. *Annals of Mathematics and Artificial Intelligence, 5,* 49-78.

Wilson, S. W. (1991). GA-easy does not imply steepest-ascent optimizable *Proceedings of the Fourth International Conference of Genetic Algorithms* (pp. 85-89). San Mateo, CA: Morgan Kaufmann.