Hyperplane Ranking in Simple Genetic Algorithms

D. Whitley, K. Mathias, and L. Pyeatt
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
whitley,mathiask,pyeatt@cs.colostate.edu

Abstract

We examine the role of hyperplane ranking during genetic search by developing a metric for measuring the degree of ranking that exists with respect to static hyperplane averages taken directly from the function, as well as the dynamic ranking of hyperplanes during genetic search. The metric applied to static rankings subsumes the concept of deception but the metric provides a more precise characterization of a function. We show that the degree of dynamic ranking induced by a simple genetic algorithm is highly correlated with the degree of static ranking that is inherent in the function, especially during the initial generations of search.

1 Introduction

Holland (1975) originally explained the computational power of genetic algorithms by developing a theory of how genetic algorithms process hyperplanes represented by schemata. Certain aspects of the theory regarding how hyperplanes are processed by genetic algorithms have been under attack for the last few years. There are serious questions about whether genetic algorithms allocate exponential trials to the observed best hyperplanes (e.g. Grefenstette and Baker 1989; Grefenstette 1991). There are also questions about whether static observations about functions (such as whether a function is deceptive or not) can be used to make inferences about the dynamical behavior of a genetic algorithm during search (Grefenstette, 1993).

The criticisms of the various theories related to hyperplane sampling take two general forms. First, counterexamples to some of the basic schema processing hypotheses have been developed. Specifically, the 2-armed bandit analogy suggests that genetic algorithms allocate exponential trials to the observed best hyperplane partitions; there are certainly counterexamples to show this is not *always* true (e.g. Grefenstette and Baker 1989; Grefenstette 1991). Another conjecture is that a simple genetic algorithm *ranks* hyperplanes according to fitness. This implies that hyperplanes with a higher average static fitness are allocated a higher proportional representation in the population over time. Counterexamples also prove that the genetic algorithm does not necessarily *always* rank hyperplanes according to fitness.

The second form of criticism of the schema processing hypothesis is that very little has actually been proven aside from the schema theorem itself. New theoretical models of how the genetic algorithm behaves as a dynamical system shed little or no light on schema processing. For example, the k-armed bandit analogy assumes the arms of the bandit are independent variables, but hyperplane representations are not independent. To what degree, then, does the k-armed bandit analogy apply to genetic algorithms and competing sets of hyperplanes?

This paper takes an empirical approach to studying some of the principles that motivated Holland's original theory about schema processing and hyperplane ranking. A metric is developed that measures the degree to which the proportional representation of hyperplanes in a population corresponds to the static average fitness of strings contained in those hyperplane subpartitions. The results suggest that the degree to which the genetic algorithm ranks hyperplanes depends very much on the degree to which those hyperplanes are ranked statically when all points in the search space are evaluated. Thus, there appear to be special conditions under which a simple genetic algorithm does rank many competing hyperplanes in an appropriate fashion, although it does not and can not consistently rank all hyperplanes for most functions.

1.1 Does the Genetic Algorithm Rank All Schemata?

Let μ_ξ be the expected average evaluation of all strings in hyperplane ξ. Let $P(\xi,t)$ represent the proportion of a population that samples hyperplane ξ at time t during the execution of a simple genetic algorithm. Also, $f(\xi,t)$ represents the average fitness of the strings in a population that sample ξ at time t during genetic search. Holland suggests that the average fitness of strings sampled from hyperplane ξ at time t will come to estimate μ_ξ and that each hyperplane ξ will come to be represented in the population in accordance with the *observed* fitness of that hyperplane. More precisely, Holland states,

> The discussion of "intrinsic parallelism" in chapter 4 would imply here that *each* [hyperplane] ξ represented in [the population] $\beta(t)$ should increase (or decrease) at a rate proportional to *its* observed "usefulness" .... If this could be done consistently then each ξ would be automatically and properly ranked within $\beta(t)$ as t increases (1975:88).

If we equate "usefulness" with $f(\xi,t)$ and the increase (or decrease) of ξ with $P(\xi,t)$, then the preceding discussion of intrinsic parallelism might be rephrased in the form of the following conjecture.

For any hyperplanes ξ_x and ξ_y, $P(\xi_x,t) > P(\xi_y,t)$ if and only if $\mu_{\xi_x} > \mu_{\xi_y}$.

Such a statement is clearly not true in general for any pair of hyperplanes.

Observation: If $\xi_x \subseteq \xi_y$ then $P(\xi_y,t) \geq P(\xi_x,t)$; if ξ_x is a proper subset of ξ_y and ξ_{y-x} represents all strings in hyperplane ξ_y that are not in hyperplane ξ_x then $P(\xi_y) > P(\xi_x,t)$ provided that $P(\xi_{y-x}) \neq 0$.

Proof: Divide the strings in ξ_y into ξ_x and ξ_{y-x}. If $\xi_x \subset \xi_y$, then by definition $P(\xi_y,t) = P(\xi_x,t) + P(\xi_{y-x},t)$ and $P(\xi_y,t) \geq P(\xi_x,t)$; if $P(\xi_{y-x}) \neq 0$, then $P(\xi_y)$ is guaranteed to have positive representation in the population and $P(\xi_y) > P(\xi_x,t)$.

In simple terms, the proportional representation of a lower order hyperplane subpartition must be greater than or equal to the proportional representation of any higher order hyperplane subpartition that is fully contained within the lower order hyperplane, regardless of their average fitness values.

Therefore, we explore a more restricted notion of ranking. We specifically look at ranking among the schemata in a hyperplane *partition*; this has also been referred to as a set of *primary hyperplane competitors* (Whitley 1991). The term *partition* will be used here, but the meanings are identical. To define a partition of order-k, pick k bit positions in a string of length l. Place the symbol * in the $(l-k)$ positions which have not been selected. Next enumerate all possible binary values over the k positions. This divides the entire search space into equal size subsets corresponding to hyperplane subpartitions. For example, the set of schemata $\{00**, 01**, 10**, 11**\}$ constitutes an order-2 partition with bit positions 3 and 4 selected. In this paper bit positions are numbered from *right to left*, starting with 1. This has the representational advantage that a string with a single bit at position i has a numeric value of 2^{i-1}.

We now consider ranking as it applies to a single partition. From a dynamic point of view, one might argue that a partition denoted by H, is ranked if

$$\forall(\xi_x,\xi_y)\epsilon H \left\{ P(\xi_x,t) > P(\xi_y,t) \quad IFF \quad \mu_{\xi_x} > \mu_{\xi_y} \right\}.$$

This definition of hyperplane ranking also fails to characterize what actually happens during genetic search. For example, assume the schemata $000**$ and $*111*$ have the highest static average fitness within their respective hyperplane competitions. Clearly, both 000** and $*111*$ cannot continually increase their representation in the population as measured by $P(\xi,t)$ since the decision whether to select for the 0 or 1 bit in positions 3 and 4 must eventually be resolved. If $*111*$ increases its representation in the population, hyperplane 000** will be represented less than other hyperplanes such as 011** even though $\mu_{\xi_{000**}} > \mu_{\xi_{011**}}$. Thus, one cannot predict in general how a genetic algorithm will dynamically rank a partition of competing schemata without considering what is happening in other overlapping partitions.

One critical problem that arises when trying to relate measurements based on dynamical population behavior (e.g. $P(\xi,t)$) to static hyperplane averages (e.g. μ_ξ) is that *inconsistent* bit patterns may exist over different sets of hyperplane partitions when ranked according to static hyperplane fitness averages. On the other hand, the degree to which inconsistency can exist in the population representation is very much constrained. This implies that there can be relationships between hyperplanes with respect to measurements based on μ that cannot be captured in the relative population representation of hyperplanes as measured by $P(\xi,t)$.

1.2 Consistent Static Rankings

Some functions have what we will term a *consistent static ranking* over all hyperplanes. For example, let the evaluation function be a count of the number of 1 bits in any given string. The static rankings over any two partitions of the search space will be consistent for this function. Thus, for a four bit problem we say that the static ranking $(\mu_{11**} > \{\mu_{10**}, \mu_{01**}\} > \mu_{00**})$ is consistent with the static ranking $(\mu_{*11*} > \{\mu_{*10*}, \mu_{*01*}\} > \mu_{*00*})$. The rankings are consistent in this case because for both sets of rankings, schemata with a greater number of 1 bits are ranked higher than schemata with fewer 1 bits. Assuming the global optimum of a function is the string of all 1 bits, this counting principle provides an informal definition of a *consistent static ranking*. A more formal definition is offered in Section 2 and another quantitative characterization is given in Section 2.1.

If a function has a consistent static ranking then this has implications for measurements based on $P(\xi, t)$ when that function is processed by a simple genetic algorithm. The orderings $P(11**, t) > P(00**, t)$ and $P(*11*, t) > P(*00*, t)$ can both be true for a single population. This bit consistency also implies that populations exist where the dynamic rankings over hyperplanes induced by $P(\xi_x, t)$ can be ranked in the same order as the static hyperplane rankings induced by μ. For such functions, the representation of hyperplanes in the population could potentially reflect the relative average hyperplane fitnesses.

Few functions will have a consistent static ranking since this implies that all partitions of the search space have consistent rankings. We prove that a *consistent static ranking* holds for linear functions if and only if the set of strings have a consistent static ranking.

Most functions will not have a consistent static ranking, but may display some *degree of consistency* with respect to the static ranking of hyperplane partitions. We therefore generalize the notion of consistency and propose a metric, ϕ, which can be used to measure the relative *degree of consistency* within a single partition. A formal definition of ϕ is given in Section 2.1. The ϕ metric measures the degree of consistency between some target bit pattern and an arbitrary rank ordering of the schemata contained in a partition. A consistency measure independent of a target bit pattern would be preferred, but at this point we have not found an obvious way to define such a measurement. Measuring the ranking of the schemata in a partition with respect to a target bit pattern also has the advantage that each partition has its own unique value for ϕ independent of all other partitions.

The sum of the ϕ metric over all partitions H for some arbitrary ranking of the schemata in those hyperplanes is denoted $\sum \phi_H$. Applying $\sum \phi_H$ to rankings based on μ with respect to the same target bit pattern yields a measure of the degree of static consistency for that function with respect to the target bit pattern. We will refer to $\sum \phi_H$ measured with respect to rankings based on μ (with N as the target bit pattern representing the global optimum) as the *static ranking* of a function.

Because the ϕ metric relates an arbitrary ranking of schemata in a partition to a target bit pattern, it can also be used to measure rankings based on dynamic relationships between schemata, such as $P(\xi, t)$. The definition of ϕ does not change; rather, the measurement is taken with respect to a different set of rankings. We will refer to $\sum \phi_H$ measured with respect to rankings based on $P(\xi, t)$ (with N as the target bit pattern representing the global optimum) as the *dynamic ranking* of a function at time t.

Empirical data based on correlations between the static ranking and dynamic ranking of random functions indicates that static ranking and dynamic ranking are very highly correlated after the first generation of genetic search. The data also suggests that the degree to which the genetic algorithm is able to dynamically rank hyperplanes during the first few generations of genetic search very much depends on the degree of consistency that is inherent in the static ranking of the function based on the average hyperplane fitnesses. Note that the dynamical rankings change as a function of generation t while the static rankings remain constant. Our results also show that the correlation between the static ranking of a function and the dynamic ranking of a function decreases as search progresses; this appears to be exactly the opposite of what Holland predicted.

2 Linear Functions with Consistent Static Ranking

To simplify our discussion of consistency, we first assume that the function is remapped using bitwise addition mod 2 (i.e., bitwise exclusive-or) so that the global optimum is the string composed of all 1 bits. We will assume there is a single global optimum. We denote the string of all 0 bits as string 0, and denote the string of all 1 bits as string N, where $N = 2^l - 1$. Bitwise exclusive-or between all strings in the space and the complement of the global optimum will move the global optimum to string N. This has various advantages: it does not change the actual hyperplane partitioning of points in the search space and the co-

efficients for any linear function are positive. Before discussing functions with a consistent static ranking we first show that all linear functions are still linear after the global optimum is remapped to string N. To do this we first develop a simple algorithm to determine the set of weight coefficients for any binary function.

Let $c = E(0)$, where $E(X)$ is the evaluation of string X. Let $w_L, w_{L-1}, \ldots, w_2, w_1$ be the weight coefficients of the linear function, which can also be expressed as a vector α such that $E(X) = \alpha^T X + c$ (c.f. Liepins and Vose, 1990; Vose and Wright, 1994). For any linear function coefficient $w_1 = E(1) - c$ and in general $w_i = E(2^{i-1}) - c$. Let c^0 represent the coefficient c when the global optimum is at 0 and let c^N represent the coefficient c when the global maximum is at N.

Lemma 1. Any linear function expressed in the form $E(X) = c + w_1 b_1 + w_2 b_2 + \ldots + w_L b_L$ can be remapped as a linear function with its global maximum at string N as follows: $c^N = c - \sum_{(w_i<0)} |w_i|$ and $w_i^N = |w_i|$ where w_i^N are the remapped weights and $c^N = E(0)$ when the global optimum is at N.

Proof: Let $\oplus$ denote exclusive-or. The following establishes a general transformation for linear functions.[1]

$$\alpha^T(X \oplus Y) + c = c + \sum_i \alpha_i (x_i \oplus y_i)$$

$$= c + \sum_{y_i=0} \alpha_i x_i + \sum_{y_i=1} \alpha_i (1 - x_i)$$

$$= \{c + \sum_{y_i=1} \alpha_i\} + \sum_{y_i=0} \alpha_i x_i - \sum_{y_i=1} \alpha_i x_i$$

$$= \{c + \sum_{y_i=1} \alpha_i\} + \sum \beta_i x_i$$

where $\beta_i = \alpha_i$ if $y_i = 0$, otherwise $\beta_i = -\alpha_i$

Let K be the current global optimum and Y be the complement of K. For all strings X, Y remaps the space; Y also remaps string K to string N. Since Y has 1 bits in the locations that correspond to $w_i < 0$,

$$c^N = c - \sum_{(w_i<0)} |w_i| = \{c + \sum_{y_i=1} \alpha_i\}$$

If $y_i = 0$, α_i is positive and β_i is positive. If $y_i = 1$, α_i is negative and β_i is positive. Since Y is fixed for all X, $w_i^N = |w_i| = \beta_i$. QED

[1] Our thanks to Michael Vose for suggesting this general transformation which improved our original proof.

Definition Let q_i be the number of 1 bits in the schema representing hyperplane ξ_i. A partition has a *consistent static ranking* with respect to a function F with a global optimum at N if $\mu_{\xi_x} > \mu_{\xi_y}$ when $q_x > q_y$. If all partitions of a function have a *consistent static ranking* with respect to a function F with a global optimum at N then function F is also said to have a *consistent static ranking*. Note that strings can be ranked as a special case of the hyperplanes. For any string X, $q_i = 1^T X$.

Theorem 1: A linear function has a consistent static ranking if and only if all strings in the search space have a consistent static ranking.

Proof: If a linear function has a consistent static ranking, then by definition all partitions have a consistent static ranking, including the set of all strings. Thus, if a linear function has a consistent static ranking, the partition of all strings has a consistent static ranking.

We now show that if the strings have a consistent static ranking and the function is linear, then all partitions have a consistent static ranking. Assume that a linear function F has been remapped so that the global optimum is at string N. Consider schema ξ of order-k. Let i index the fixed bits in schema ξ and j index * positions in the schema ξ. Since F is linear, for each string $X \epsilon \xi$, $E(X) = \sum_i w_i b_i + c + \sum_j w_j b_j$. Therefore

$$\mu_\xi = \frac{\sum_{(X \epsilon \xi)} ((\sum_i w_i b_i) + c) + \sum_{(X \epsilon \xi)} \sum_j w_j b_j}{2^k}$$

Since i indexes those bits which do not vary in schema ξ, it follows that $\sum_i w_i b_i$ defined with respect to ξ can be computed by converting ξ into a string. Let Y_ξ be the string produced when the * positions in schema ξ are replaced by 0. It follows that $\alpha^T Y_\xi + c = \sum_i w_i b_i + c$ when computed with respect to ξ and that

$$\mu_\xi = \frac{\sum_{(X \epsilon \xi)} (\alpha^T Y_\xi + c) + \sum_{(X \epsilon \xi)} \sum_j w_j b_j}{2^k}$$

$$= (\alpha^T Y_\xi + c) \left\{ \frac{\sum_{(X \epsilon \xi)} 1}{2^k} \right\} + \frac{\sum_{(X \epsilon \xi)} \sum_j w_j b_j}{2^k}.$$

which reduces to

$$\mu_\xi = \alpha^T Y_\xi + c + \frac{\sum_{(X \epsilon \xi)} \sum_j w_j b_j}{2^k}.$$

Let ξ_q and ξ_z be any two hyperplanes that are members of a partition. Note that

$$\mu_{\xi_q} = \alpha^T Y_{\xi_q} + c + \frac{\sum_{(X \epsilon \xi_q)} \sum_j w_j b_j}{2^k}$$

$$\text{and} \quad \mu_{\xi_z} = \alpha^T Y_{\xi_z} + c + \frac{\sum_{(X \epsilon \xi_z)} \sum_j w_j b_j}{2^k},$$

$$\text{therefore} \quad \frac{\sum_{(X \epsilon \xi_q)} \sum_j w_j b_j}{2^k} = \frac{\sum_{(X \epsilon \xi_z)} \sum_j w_j b_j}{2^k},$$

since we are enumerating all possible bit values over exactly the same positions in both cases. It follows that

$$\mu_{\xi_q} > \mu_{\xi_z} \quad IFF \quad (\alpha^T Y_{\xi_q} + c) > (\alpha^T Y_{\xi_z} + c).$$

Thus, for any linear function, if all strings in the space have a consistent static ranking, then Y_{ξ_q} and Y_{ξ_z} have a consistent static ranking.

QED

Not all linear functions have a consistent static ranking, however. It is possible for a small number of linear weight coefficients to contribute more to the evaluation of some string X compared to a larger number of weight coefficients for some string Y. Thus, 110000 may have a higher evaluation than 001111 and analogously, hyperplane 100**** may have a higher average fitness than 011****.

2.1 Partially Ranked Functions

Next we develop a metric for partially ranked functions which can provide information about the *static ranking* of a function. As already noted, the metric, ϕ, is general since it measures the degree of consistency between an arbitrary ranking of hyperplanes in a partition and some target bit string. The ordering of the hyperplanes can thus be static (based on μ_ξ) or dynamic (based on $P(\xi, t)$). The *dynamic ranking* can be measured during the execution of an actual genetic algorithm, or using a model of the dynamical behavior of a genetic algorithm. We use a model of the simple genetic algorithm in this paper (Vose and Liepins, 1991; Whitley, 1993).

We assume that the *static ranking* and *dynamic ranking* by default refer to rankings where the target bit pattern is the global optimum and the global optimum has been mapped to string N. However, one can also calculate ϕ with respect to strings other than the global optimum. For "fully deceptive functions" for example, $\sum \phi_H$ will be low with respect to the global optimum, but would be high if measured with respect to the complement of the global optimum as the target bit pattern.

For each hyperplane $\xi \epsilon H$, construct the string Y_ξ and compute $1^T Y_\xi$. Recall Y_ξ is constructed by replacing the * symbol with 0 in the schema representing ξ and that $1^T Y_\xi$ measures the number of bits shared by N and the schema representing hyperplane ξ. As noted in the previous section, a function has a "consistent static ranking" if sorting all of the hyperplanes in each partition with respect to μ_ξ also results in an ordering of the hyperplanes with respect to $1^T Y_\xi$. (Note that ties in the rankings with respect to $1^T Y_\xi$ do not cause a problem.)

We will define ϕ with respect to a ranking function R. R could be a ranking of all partitions based on μ or $P(\xi, t)$. We now measure the *degree of consistency* by defining a metric which measures the degree to which the values $1^T Y_\xi$ are sorted when the corresponding hyperplanes are sorted with respect to R.

First, consider a partition denoted H and a schema representing a hyperplane denoted ξ such that $\xi \epsilon H$. Assume ξ is of order k. Sort the members of H with respect to R. Let i be the index of the rank of $\xi \epsilon H$ after the hyperplanes are sorted. The degree of ranking for H is then defined as follows:

$$\phi_{H,R} = \sum_{i=1}^{2^k} \sum_{j=i}^{2^k} Threshold(1^T Y_{\xi_i} - 1^T Y_{\xi_j})$$

$$\text{where} \quad Threshold(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{otherwise} \end{cases}$$

Note that $\phi_{H,R}$ has its maximum possible value when all the values for $1^T Y_\xi$ are also ranked.

$$Max(\phi_{H,R}) = \sum_{q=1}^{k} \binom{k}{q} \sum_{i=0}^{q-1} \binom{k}{i} q - i$$

We divide each count $\phi_{H,R}$ by $Max(\phi_{H,R})$ so that the resulting measure of hyperplane ranking in H is between 0 and 1. If a function has a *consistent static ranking* then $\phi_{H,R}/Max(\phi_{H,R}) = 1$ for all partitions H when R is based on μ.

Table 1 computes values for ϕ for different rankings for the same hyperplane partition. If these rankings are based on μ, then each rank corresponds to a different function. If the rankings are based on $P(\xi, t)$, then the rankings could be the population representation for H for the same function at different generations.

2.2 ϕ and Deception

Assume R is based on μ so that we are measuring the degree of static ranking. When $\phi_{H,R}/Max(\phi_{H,R}) = 1$ for all partitions then the function is nondeceptive, but not all nondeceptive problems have a static ranking of $\phi_{H,R}/Max(\phi_{H,R}) = 1$ for all partitions. "Deception"

$\xi \epsilon H_p^1$	$1^T Y_\xi$	$\xi \epsilon H_p^2$	$1^T Y_\xi$	$\xi \epsilon H_p^3$	$1^T Y_\xi$	$\xi \epsilon H_p^4$	$1^T Y_\xi$
111***	3	111***	3	110***	2	000***	0
110***	2	100***	1	000***	0	001***	1
101***	2	010***	1	100***	1	010***	1
011***	2	101***	2	011***	2	100***	1
001***	1	001***	1	001***	1	011***	2
010***	1	011***	2	101***	2	101***	2
100***	1	000***	0	111***	3	110***	2
000***	0	110***	2	010***	1	111***	3
	$\phi_{H,R_1} = 30$		$\phi_{H,R_2} = 20$		$\phi_{H,R_3} = 10$		$\phi_{H,R_4} = 0$

Table 1: Computations of ϕ with respect to various rankings of H.

is defined with respect to only the top ranked hyperplane without considering the consistency of the ranking over all hyperplanes as represented by schemata. Assuming R is based on μ then R_1 and R_2 in Table 1 would correspond to different functions; neither of these function have "deception" in these partitions, but they have different ϕ values. Thus, deception is a less precise characterization of a function compared to its static ranking. During the computation of $Max(\phi_{H,R})$, when $q = k$ we are counting the contribution to $\phi_{H,R}$ made by the top ranked hyperplane assuming that the schemata in the partition display no deception. Since deception in a partition is defined with respect to only the top ranked hyperplane in that partition deception is independent of how the other hyperplanes are ranked with respect to each other.

3 The Relation Between Static and Dynamic Ranking

We collected statistics based on a model of the genetic algorithm assuming an infinite population size, a crossover rate of 0.6 and a mutation rate of 0. All strings have equal representation in the initial population at generation 0. The recombination operator was 1-point crossover. We generated 100 random 6 bit functions in order to evaluate the relationship between the static ranking of hyperplanes for each function and the dynamic ranking of the hyperplanes during genetic search for that same function. Functions were generated by randomly assigning an integer value between 0 and 256 to each of the 64 strings composing the search space for each function. We also wished to include a few nondeceptive functions in the comparison of the static and dynamic rankings. We generated 100,000,000 random functions; only 14 of these functions were nondeceptive. These were also included in our set of test functions.

We then used these 114 functions to compare the static ranking to the dynamic ranking when a model of the genetic algorithm was executed on each individual function. The results for generations 1, 2, 5, 10, 15 and 20 are shown in Figure 1.

For each function we computed the sum of $\phi_{H,R}/Max(\phi_{H,R})$ over all hyperplane partitions except the largest and smallest partitions. We will denote this sum simply as $\sum \phi$. The number of possible hyperplane partitions is given by $\sum_{i=1}^{l} \binom{l}{i} = 2^l - 1$. The set of all strings is not counted as a partition. Also, ϕ was not calculated for the partition where each hyperplane is composed of one string. Thus, ϕ was measured over $2^l - 2$ partitions and therefore $\sum \phi$ ranges from 0 to $2^l - 2$.

Each point on the graph represents the dynamic ranking as measured by $\sum \phi$ plotted against the static ranking measured using $\sum \phi$ for one of the 114 functions. This range is 0 to 62 for a 6 bit function, which accounts for the scale found on the graphs in Figure 1.

Figure 1 shows that the degree of dynamic ranking during the first few generations is highly correlated with the degree of static ranking which is inherent in the functions. However, the correlation does not improve with time. Instead, the degree of dynamic ranking increases if the genetic algorithm converges toward the global optimum. The degree of dynamic ranking with respect to the global optimum decreases if the genetic algorithm allocates increasing numbers of trials to hyperplanes with bit patterns that are inconsistent with the global optimum. The degree to which a point moves up or down indicates the number of 1 bits (i.e. bits shared with the global optimum) in the solution toward which the population is converging. [2]

[2] Since our model contained no mutation, it indeed makes sense to talk about the simple genetic algorithm converging to a single point.

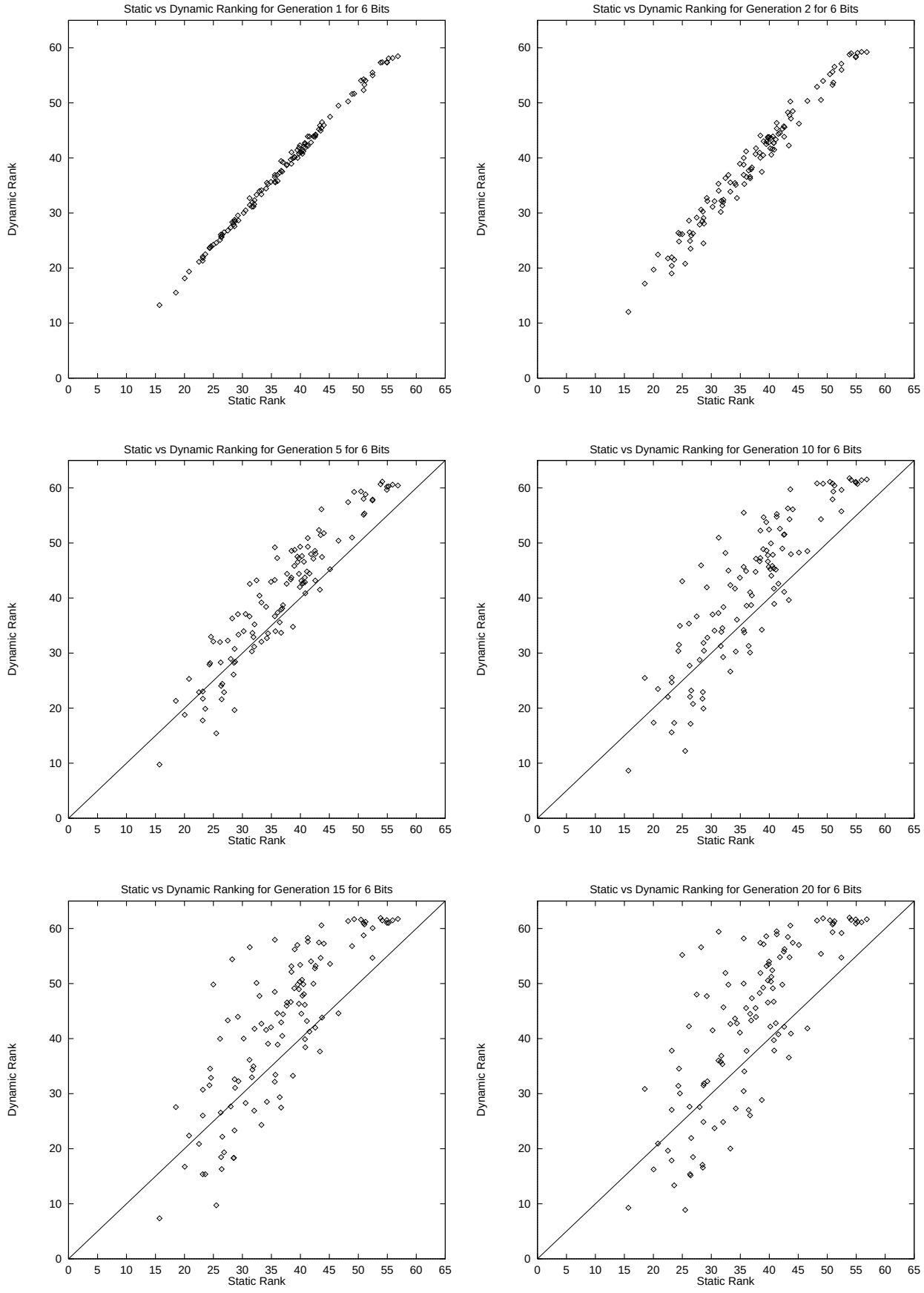

Figure 1: Static Ranking versus Dynamic Ranking for Generations 1, 2, 5, 10, 15 and 20.

It should also be noted that this correlation is entirely emergent during the first generation of the simple genetic algorithm. It has nothing to do with the ranking of the population during generation 0 (before search begins). While we do not present a graph for generation 0, the distribution would not change for the static rankings (since these are fixed) and the dynamic ranking would be identical for all functions because the model has an identical representation for all hyperplanes over all functions. Thus, the distribution would appear as a straight horizontal line across the graph. In the first generation this jumps from a flat line to the distribution shown–which implies that the dynamic ranking as measured by $\sum \phi$ that emerges during generation 1 is very consistent with the static ranking of the function as measured by $\sum \phi$. It is important that during only the first generation of the simple genetic algorithm is the dynamic ranking based on selection applied to a uniform sample over the entire function.

All linear and nondeceptive functions are found in the extreme upper right hand corner of the graphs. Vose and Wright (1994) have recently shown that a simple genetic algorithm applied to a linear function will always converge to the global optimum assuming that the population is sufficiently large. We have shown that for certain linear functions all hyperplane partitions are statically ranked. However, all linear functions will be characterized by a high degree of static ranking. More work is needed to extend these kinds of results to other functions characterized by a high degree of static ranking.

4 Conclusions and Future Work

The ϕ metric has a number of useful properties. The ϕ metric can yield different consistency measures for hyperplane partitions that are classified as deceptive (or nondeceptive). Thus, this metric is both a less rigid and a more sensitive measurement than measurements based on deception. Also, we empirically show that there is indeed a very strong correlation between the static ranking of a function measured by ϕ and the dynamic ranking induced by a genetic algorithm after the first generation of genetic search. In addition, the ϕ metric need not be defined with respect to the global optimum. One can determine the degree of static ranking or dynamic ranking that exists with respect to any point in the search space.

An interesting study, for example, would be to determine to what point the simple genetic algorithm eventually converged for each of the 114 functions, and then to repeat the comparison of the static and dynamic rankings with respect to these points. In this case the dynamic ranking would increase over time for all functions. However, what would happen to the correlation with the static rankings during the first few generations? Also, what would the correlations between static and dynamic ranking look like when the measurements are taken with respect to an actual genetic algorithm as opposed to a model?

The ϕ metric is only one way to measure the consistency of the hyperplane rankings of a function. Better metrics could be developed and might yield more insight into the processing of schemata by genetic algorithms.

Acknowledgements

This work was supported in part by NSF grant IRI-9312748 and IRI-9503366. Part of this work was also done while the first author was visiting the Santa Fe Institute; their support is gratefully acknowledged.

References

[Grefenstette, 1993] Grefenstette, J. (1993). Deception Considered Harmful. In Whitley, L. D., editor, *FOGA - 2*, pages 75–91. Morgan Kaufmann.

[Greffenstette, 1991] Greffenstette, J. (1991). Conditions for Implicit Parallelism. In Rawlins, G., editor, *FOGA -1*, pages 252–261. Morgan Kaufmann.

[Greffenstette and Baker, 1989] Greffenstette, J. and Baker, J. (1989). How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. In Schaffer, J. D., editor, *Proc. of the 3rd Int'l. Conf. on GAs*, pages 20–27. Morgan Kauffman.

[Holland, 1975] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

[Liepins and Vose, 1990] Liepins, G. and Vose, M. (1990). Representation Issues in Genetic Algorithms. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:101–115.

[Vose and Liepins, 1991] Vose, M. and Liepins, G. (1991). Punctuated Equilibria in Genetic Search. *Complex Systems*, 5:31.44.

[Vose and Wright, 1994] Vose, M. and Wright, A. (1994). Simple Genetic Algorithms with Linear Fitness. *Evolutionary Computation*.

[Whitley, 1991] Whitley, L. D. (1991). Fundamental Principles of Deception in Genetic Search. In Rawlins, G., editor, *FOGA -1*, pages 221–241. Morgan Kaufmann.

[Whitley, 1993] Whitley, L. D. (1993). An Executable Model of the Simple Genetic Algorithm. In Whitley, L. D., editor, *FOGA - 2*, pages 45–62. Morgan Kaufmann.