A Theoretical Analysis of the HIFF Problem

Nicholas Freitag McPhee
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota
mcphee@morris.umn.edu

Ellery Fussell Crane
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota
cran0117@morris.umn.edu

ABSTRACT

We present a theoretical analysis of Watson's Hierarchical-if-and-only-if (HIFF) problem using a variety of tools. These include schema theory and course graining, the concept of effective fitness, and statistical analysis. We first review the use of Stephen's exact schema equations and schema basis to compute the changes in population distributions over time. We then use the tools described above to solve for the limit distributions of the 2 and 4-bit HIFF problems, and show that these limit distributions are essentially one-dimensional. We also show that a combination of fitness and the number of break points (a rough measure of distance in crossover space) in a string can be used to almost completely explain the limit distribution in the 4-bit HIFF problem.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*; G.3 [**Probability and Statistics**]: *Distribution functions, probabilistic algorithms, stochastic processes*

General Terms

Theory

Keywords

Evolutionary Computation, Schema Theory, Hierarchical-if-and-only-if (HIFF) Problem, Limit Distributions, Distance Metrics

1. INTRODUCTION

One of the primary goals of evolutionary computation (EC) theory is to better understand the dynamics of evolutionary systems, in the hopes that such an understanding will be of both scientific and practical value. Even in simple problems, however, these dynamics can be complex. This often makes a detailed understanding of interesting problems quite difficult.

GECCO'05, June 25–29, 2005, Washington, DC, USA.

In this study we apply schema theory and other tools to better understand the dynamics of the Hierarchical-if-and-only-if (HIFF) problem [17]. The HIFF problem is simple to define, yet has interesting and complex dynamics that have been valuable in previous studies of important concepts such as the Building Block Hypothesis [3, 2, 16] and modularity [15].

One of the earliest and best known analytical tools in genetic algorithms (GAs) is Holland's Schema Theorem [3]. This was, however, an inexact theorem, providing only a lower bound on the propagation of schemata over time. This was later extended by Stephens and Waelbroeck to an exact schema equation for GAs [12, 13]. Exact schema theory, which has been successfully applied in a number of settings [4, 13, 14, 8, 9], will serve as the primary analytical tool in the work presented here.

After providing necessary background information in Section 2 we present a theoretical analysis of the HIFF problem in Section 3. We begin by using schema theory and coarse graining to calculate population distributions and solve for limit distributions of the 2 and 4-bit HIFF problem (Sections 3.1, 3.2, and 3.3). We also show that these limit distributions are essentially one-dimensional, since we can use the proportion of a single schema in the limit distribution to compute all the other proportions in the distribution.

In Sections 3.4 and 3.5, we analyze these results using statistical methods and the concept of effective fitness. With these tools, we verify the existence of the limit distributions, and show that a combination of fitness and break points (a rough measure of distance in crossover space) almost completely explain the limit distribution in the 4-bit HIFF problem.

After discussing our ideas for future avenues of research in Section 4, we present our conclusions in Section 5.

2. BACKGROUND

In this section we review a variety of concepts and definitions used in this study. We also present the details of the problem setup used throughout the paper.

2.1 The HIFF problem

The HIFF problem is defined using a recursive fitness function, as follows: If the bit string being considered consists of all zeros or all ones, the fitness of the string is equal to the length of the string, otherwise it has a fitness of 0. This same criteria is then applied recursively on each half of the string, until it can be subdivided no further. Adding the fitnesses of all substrings together yields the fitness of

the whole. Two examples of this process are presented in Figure 1.

More formally, the HIFF fitness function [17] can be defined as:

DEFINITION 1 (THE HIFF FUNCTION).

$$f(B) = \begin{cases} 1, & \text{if } |\mathrm{B}| = 1, \\ |\mathrm{B}| + f(\mathrm{B}_L) + f(\mathrm{B}_R), & \text{if } (|\mathrm{B}| > 1) \text{ and} \\ & (\forall i\{\mathrm{b}_i = 0\} \text{ or} \\ & \forall i\{\mathrm{b}_i = 1\}), \\ f(\mathrm{B}_L) + f(\mathrm{B}_R), & \text{otherwise.} \end{cases}$$

Here B is a block of bits, $\{\mathrm{b}_1, \mathrm{b}_2, ...\mathrm{b}_n\}$, $|\mathrm{B}|$ is the size of the block (and therefore equal to n), b_i is the ith element of B, and B_L and B_R are the left and right halves of B (i.e. $\mathrm{B}_L = \{\mathrm{b}_1, ...\mathrm{b}_{n/2}\}$, $\mathrm{B}_R = \{\mathrm{b}_{n/2+1}, ...\mathrm{b}_n\}$). n must be an integer power of 2.

From Definition 1, it follows that for any value of n, there will be two global optima: the string consisting of all zeros, and the string consisting of all ones. In the 4-bit case for example, the global optima are the strings 0000 and 1111, each of which has a fitness of 12. Similarly, the lowest fitness a string can have is equal to the length of the string, and occurs when there are no matching pairs.

2.2 Problem setup

In this paper we will be dealing solely with a "standard" fixed length, binary bit string GA using crossover and no mutation. We will use one-point crossover, where all crossover points (including the point before and after the entire string) are equally likely. A generational model and fitness proportionate selection will be used throughout.

In our studies of both the 2 and 4-bit HIFF problems, we will always start with an initial distribution containing equal proportions of the least fit strings. In the 2-bit HIFF problem this means that we start with half the strings being 01 and half the strings being 10. In the 4-bit case there are four strings having minimal fitness (0101, 0110, 1001, and 1010), so our initial distribution will be 25% of each of these string and 0% of the other 12 strings.

2.3 Schema theory

In this section we will present key concepts from the exact schema theory of Stephens and Waelbroeck [12, 13]. This theory allows us to calculate the exact proportions of different strings in the population over time, but with the important assumption that we have an infinite population.

In this paper a schema (typically represented with H) is sequence of symbols from the set $\{0, 1, *\}$, where $*$ is the "don't care" or wild card symbol. A schema, then, represents the set of individuals who match the schema at all the defined positions, i.e., those positions having either a 0 or a 1. Schemata of this form allow for *coarse graining* [13], where we can treat whole sets of strings as a single entity.

Next we present a series of definitions that allow us to count instances of schemata:

DEFINITION 2 (PROPORTION IN POPULATION). *$\phi(H, t)$ is the proportion of strings in the population at time t matching schema H. For finite populations of size M, $\phi(H, t) = m(H, t)/M$, where $m(H, t)$ is the number of instances of H at time t.*

DEFINITION 3 (SELECTION PROBABILITY). *$p(H, t)$ is the probability of selecting an instance of schema H from the population at time t. This is typically a function of $\phi(H, t)$, the fitness distribution in the population, and the details of the selection operators. With fitness proportionate selection, for example, $p(H, t) = \phi(H, t) \times f(H, t)/\overline{f}(t)$, where $f(H, t)$ is the average fitness of all the instances of H in the population at time t and $\overline{f}(t)$ is the average fitness in the population at time t.*

DEFINITION 4 (TRANSMISSION PROBABILITY). *$\alpha(H, t)$ is the probability that an instance of the schema H will be constructed in the process of creating a new individual for the population at time $t + 1$ out of the population at time t. This will typically be a function of $p(K, t)$ for the various schemata K that could play a role in constructing H, and the details of the various recombination and mutation operators being used.*

We can now model the standard evolutionary algorithm as the transformation

$$\phi(H, t) \xrightarrow{\text{select}} p(H, t) \xrightarrow{\text{XO}} \alpha(H, t) \xrightarrow{\text{sample}} \phi(H, t + 1).$$

Here the arrows indicate that some new distribution (on the right hand side of the arrow) is generated by applying the specified operation to the previous distribution (on the left hand side). The process of selection can be seen as a transformation from the proportions of schemata $\phi(H, t)$ to the selection probabilities $p(H, t)$. A crucial observation is that for an *infinite* population the sampling process is exact, so $\alpha(H, t) = \phi(H, t+1)$ for $t \geq 0$. This means we can iterate these transformations to *exactly* model the behavior of an infinite population over time.

We have shown how to compute $p(H, t)$ from $\phi(h, t)$ using fitness proportionate selection, and how to compute $\phi(H, t+1)$ from $\alpha(H, t)$ (because we are using an infinite population). To complete our formalization of the transformation process, all that remains is to compute the transmission probabilities $\alpha(H, t)$ from $p(H, t)$ via modelling crossover. To formalize this process, we consider a schema $H = c_0 c_1 \ldots c_{N-1}$, where each $c_i \in \{0, 1, *\}$, and define[1]

$$\begin{aligned} l(H, i) &= c_0 c_1 \ldots c_{i-1} (*)^{N-i} \\ r(H, i) &= (*)^i c_i c_{i+1} \ldots c_{N-1} \end{aligned}$$

Here $l(H, i)$ is the schema matching the leftmost i symbols of H, and $r(H, i)$ is the schema matching the rightmost $N - i$ symbols of H. The important property of l and r is that

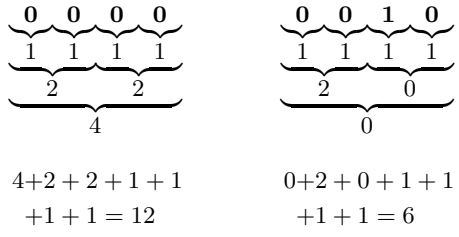

Figure 1: Examples of HIFF fitness calculation on the strings 0000 and 0010

[1]The notation $(*)^k$ here represents a string of k wildcard symbols.

if one uses one-point crossover to crossover *any* instance of $l(H,i)$ at position i with *any* instance of $r(H,i)$ at position i, the result will be an instance of H, provided $0 \leq i \leq N$. Further, these are the *only* ways to use crossover to construct instances of H, so these definitions fully characterize the mechanism for constructing instances of H.

Given these definitions, we can present the schema theorem (based on [13]):

THEOREM 1 (SCHEMA THEOREM). *For fixed length bit strings using one-point crossover and no mutation we have*

$$\alpha(H,t) = \frac{1}{N} \sum_{0 \leq i \leq N} \Big[p(l(H,i),t) \times p(r(H,i),t) \Big] \quad (1)$$

2.4 Schema and string bases

The schema calculations above can be performed in any of a number of bases [1]. A basis in this context is a set of 2^N schemata (where N is the length of the bit strings) such that knowing the proportions of those schemata allows one to reconstruct the proportions of all of the 3^N schemata. As is discussed in [1], the choice of basis can greatly simplify (or complicate) calculations in a system. In this work we used two bases, the string basis and the schema basis (both described below), each of which has advantages at different points in the schema calculations. In this section we will describe the two bases, discuss their uses in this work, and briefly discuss how to convert between these bases.

2.4.1 String basis

In the string basis we use all the strings without wildcards as the basis. It is then quite straightforward to compute the proportions of schemata with wildcards by simply adding up the proportions of the appropriate strings, e.g., $p(*1,t) = p(01,t) + p(11,t)$.

It is often necessary to work in the string basis when computing the selection probabilities $p(H,t)$ with non-trivial fitness functions. In particular, one can only compute the selection probabilities for schemata with uniform fitness, i.e., where every string matching the schema has the same fitness. In the HIFF problem, the only schemata that have this property are strings without wildcards, so we have to use the string basis to compute the selection probabilities.

2.4.2 Schema basis

In the schema basis [1] we track the proportion of all the schemata formed using the symbols $\{1, *\}$. This is a complete basis in the sense that if you know the proportions of all the schemata in this basis, you can calculate the proportions of any other string or schema using the 0 symbol as well. So, for example, $p(01,t) = p(*1,t) - p(11,t)$.

In this work we have performed the transmission probability calculations from Equation (1) in the schema basis as it greatly simplifies those computations. If we performed these calculations in the string basis, for example, terms like $l(H,i)$ would expand to potentially large sets, creating a large implicit double summation inside the explicit summation in (1). As an example, consider the case where $H = 0110$ and $i = 2$. Then

$$\begin{aligned} & p(l(0110,2),t) \\ &= p(01**,t) \\ &= p(0100,t) + p(0101,t) + p(0110,t) + p(0111,t) \end{aligned}$$

and $p(r(H,i))$ would expand to a similar summation. Working in the schema basis, however, avoids these implicit summations. If H is in the schema basis, both $l(H,i)$ and $r(H,i)$ are also members of the schema basis, so their proportions are known immediately and don't have to be computed using summations.

Note that it's not *necessary* to use the schema basis for these calculations as we can certainly manage the nested summations implied by the string basis. They do, however, greatly simplify those calculations, and we found it much easier to evaluate the schema theorem equations using that basis.

2.4.3 Converting between bases

Since we're using two bases in this work, we obviously need to convert probability distributions represented in one basis to the equivalent distribution in the other basis, and back again. Happily this basis conversion is accomplished through straightforward matrix multiplication. The construction of this transformation matrix is outside the scope of this paper; see [1] for more information, and Section 3.1 below for some examples.

2.5 Effective Fitness

When using fitness based selection, it is often assumed that fitness is the largest determinant of which individuals increase in number within the population over time. Studies have shown that this is not always the case, however, due to biases in the genetic operators [4]. Crossover, for example, can result in certain building blocks being constructed more frequently than others. This, correspondingly, makes the construction of certain individuals more likely than others, independent of fitness. This behavior, therefore, results in higher proportions of some individuals in the population than fitness alone can explain. The concept of *effective fitness* was invented to account for this phenomenon. A schema's effective fitness is the value its fitness would have to be, using selection alone, to cause the same change in proportion it experiences with its unmodified fitness in the presence of genetic operator bias.

Various concepts of effective fitness have been discussed in the literature; see [4], e.g., for a survey. For this study, we use Stephens and Waelbroeck's exact effective fitness for GAs [12, 13], but base our notation on [7]. This definition grows naturally out of the concepts presented in Section 2.3:

DEFINITION 5 (EFFECTIVE FITNESS). *The effective fitness of a schema H at time t is given by*

$$f_{\mathit{eff}}(H,t) = \frac{\alpha(H,t)}{p(H,t)} f(H,t) \quad (2)$$

The key idea in this definition is that of scaling the actual fitness $f(H,t)$ of the schema by $\alpha(H,t)/p(H,t)$. This would, for example, have the effect of raising the effective fitnesses of schemata that have a higher transmission probability (are constructed more often) than their selection probability alone would suggest. Thus schemata which are likely to be constructed (via crossover in our case) can have an effective fitness that is significantly higher than their actual fitness. Similarly, schemata for which crossover is a more destructive operator will have an effective fitness that is lower than their fitness because their transmission probability α will be less than their selection probability p. It is important to note that the effective fitness of a string or schemata

changes over time, unlike normal fitness which is static. This is due to the fact that effective fitness is directly tied to selection probability and transmission probability, both of which vary from generation to generation.

In Section 3.4, we will use the idea of effective fitness to help understand the limit distributions of both the 2-bit and 4-bit HIFF problems.

3. THEORETICAL ANALYSIS

In this section we apply schema theory, effective fitness, and statistical tools to the 2 and 4-bit versions of the HIFF problem. For the 2-bit case we are able to completely characterize the problem, including solving for all the possible limit distributions. For the 4-bit case we are able to find the limit distribution for a particular class of symmetric initial distributions, and statistically analyze the different forces acting on the limit distribution.

3.1 Computing proportions for 2-bit HIFF

We'll start by analyzing the 2-bit HIFF problem using schema theory. While this is a very simple problem, with only four possible strings (00, 01, 10, and 11), its simplicity makes it easier to present the schema equations and illustrates several features that will be important in subsequent discussion of a more complex instance of the HIFF problem.

In this presentation we will use vectors of the form $[a, b, c, d]$ to indicate a set of proportions or selection probabilities (depending on the context). The values in this vector will be associated with either the string basis $[00, 01, 10, 11]$ or the schema basis $[**, *1, 1*, 11]$, again depending on the context.

Let us assume, then, that we start with the initial distribution (in the string basis) of $[0, 1/2, 1/2, 0]$. We convert this to proportions in the schema basis by multiplying by the appropriate transformation matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix}$$

This means that 100% of the initial population matches the schema $**$ (which will of course always be true), half the population matches $*1$ and half matches $1*$ (because half the initial strings have a 1 in the first position and half have a 1 in second position), and no individuals match 11 (since there are no instances of that string).

We then use Equation (1) from the Schema Theorem to compute the transmission probabilities. The transmission probability for $*1$, for example, is

$$\begin{aligned} \alpha(*1) &= (p(**)p(*1) + p(**)p(*1) + p(*1)p(**))/3 \\ &= (1 \times 1/2 + 1 \times 1/2 + 1/2 \times 1)/3 \\ &= 1/2 \end{aligned}$$

Doing this for all four schemata in the schema basis yields $[1, 1/2, 1/2, 1/12]$ as the distribution (in the schema basis) after crossover.

Given this distribution in the schema basis, we can then convert back to the string basis by multiplying by the inverse of the transformation matrix above:

$$\begin{bmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1/2 \\ 1/2 \\ 1/12 \end{bmatrix} = \begin{bmatrix} 1/12 \\ 5/12 \\ 5/12 \\ 1/12 \end{bmatrix}$$

Gen	$\phi(00)$	$\phi(01)$	$\phi(10)$	$\phi(11)$
0	0	1/2	1/2	0
1	1/12	5/12	5/12	1/12
2	5/28	9/28	9/28	5/28
3	59/228	55/228	55/228	59/228
∞	$\frac{1+\sqrt{33}}{16}$	$\frac{7-\sqrt{33}}{16}$	$\frac{7-\sqrt{33}}{16}$	$\frac{1+\sqrt{33}}{16}$

Table 1: The exact proportions in the string basis for the first few generations of the 2-bit HIFF problem. The limit distribution is given in the last row (labelled ∞).

Gen	$\phi(**)$	$\phi(*1)$	$\phi(1*)$	$\phi(11)$
0	1	1/2	1/2	0
1	1	1/2	1/2	1/12
2	1	1/2	1/2	5/28
3	1	1/2	1/2	59/228
∞	1	1/2	1/2	$(1+\sqrt{33})/16$

Table 2: The exact proportions in the schema basis for the first few generations of the 2-bit HIFF problem. The limit distribution is given in the last row (labelled ∞).

This says that after one generation, the strings 00 and 11 are each represented by 1/12 of the individuals in the population, and 01 and 10 are each represented by 5/12 of the individuals.

We can then use the formula from Definition 3 to compute the selection probabilities (in the string basis). The selection probability of 01, for example, is

$$p(01) = \frac{5}{12} \times \frac{f(10)}{\bar{f}} = \frac{5}{12} \times \frac{2}{(7/3)} = \frac{5}{14}$$

Doing this for all the strings yields $[1/7, 5/14, 5/14, 1/7]$ as the selection probabilities after the first generation.

At this point we've completed one full generation. We can now convert back to the schema basis and compute the transmission probabilities using the Schema Theorem, then convert back to the string basis and compute the selection probabilities, etc., for as many generations as we wish. In Tables 1 and 2 we have tabulated the proportions for several generations in both the string basis and the schema basis. Figure 2 also shows the changes in the proportions over time; note the speed with which the proportions converge to their limit distributions.

There are several things to note about the string basis proportions in Table 1. First, the proportions of 00 and 11 are increasing and will come to dominate the population as we would expect. Second, the exact proportions become increasingly complex. In fact, since the proportions in the limit distribution are irrational numbers (see below), each column of this table is the beginning of a rational sequence whose limit is the irrational limit value in the final row. As a result, these rational proportions grow more complex as they approach their irrational limit. Lastly, and not surprisingly, the symmetry in the initial distribution is maintained across time, with $\phi(00) = \phi(11)$ and $\phi(01) = \phi(10)$ in all generations.

An important question is what sort of dependencies exist

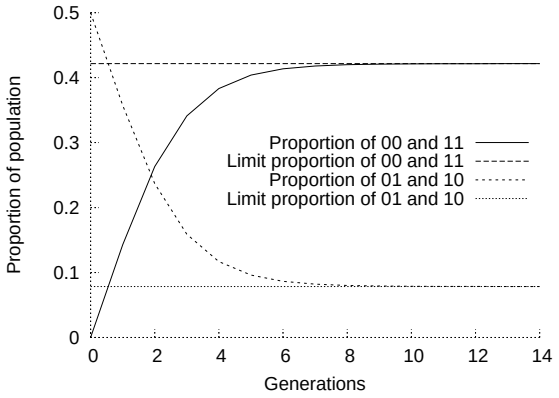

Figure 2: Proportions in the string basis over time for the 2-bit HIFF problem

in this distribution data. There are four proportions, but it's clear that they are not all independent. Since the proportions in the string basis must add up to 1, knowing any three is sufficient to compute the fourth. Further, because of the symmetry in this case, knowing any *one* proportion is enough to determine all four. Thus the proportions at each generation can be seen as a function of a single value that is changing over time.

While it is not difficult to use the string basis proportions to verify that there is only one independent variable, it's immediately obvious when looking at the schema basis proportions in Table 2. Here, the proportions of three of the schemata remain constant over time, and only the fourth one changes, making it quite clear that the process is driven by a single independent variable. This provides an excellent example of how a change of basis can make the underlying structure of the process much clearer [1].

3.2 2-bit limit distribution

Repeatedly iterating the schema equations as we did in the previous section can help us understand problem dynamics over time. It can also suggest important properties such as the stable symmetry and the existence of a single independent variable as discussed above. In some cases we can further use these equations to exactly solve for important structures such as limit distributions.

In the 2-bit HIFF problem, for example, we can start with a distribution of $[1, x, y, z]$ in the schema basis and run through one round of crossover and selection, yielding a new distribution:

$$\left[1, \frac{3x + xy + 2z}{d}, \frac{3y + xy + 2z}{d}, \frac{2(xy + 2z)}{d}\right] \qquad (3)$$

where $d = 6 - 3x + 2xy - 3y + 4z$. If $[1, x, y, z]$ is a limit distribution, then it must be unchanged by an iteration of the schema equations. Thus, by setting this new distribution equal to the initial distribution $[1, x, y, z]$ we can solve for all the possible limit distributions. In this case there are five different solutions to the resulting set of equations:

$x = y = z = 0$ $\qquad$ $x = y = z = 1$

$x = 1, y = z = 0$ $\qquad$ $y = 1, x = z = 0$

$x = y = 1/2, z = \frac{1+\sqrt{33}}{16}$

The first four solutions correspond to cases where the entire population consists of copies of a single string (00, 11, 01, and 10 respectively). The last solution (which is included as the last line in Tables 1 and 2 and indicated in Figure 2) can be shown to be the limit distribution for the symmetric case discussed above. In particular, one can use Equation (3) to show that any distribution of the form $[1, 1/2, 1/2, z]$, where $0 \leq z < (1 + \sqrt{33})/16$, will transform after one round of crossover and selection to some distribution $[1, 1/2, 1/2, z']$ where $z < z' < (1 + \sqrt{33})/16$. Therefore the limit of the sequence of z values must be $(1 + \sqrt{33})/16$.

Thus the limit distribution for the symmetric case consists of roughly 42% all zeros and 42% all ones, and then roughly 8% each of the sub-optimal strings 01 and 10. This means that in the limit of the 2-bit HIFF problem starting with symmetric distribution above, the GA is focussing the substantial majority of it's "attention" on the global optima. As we shall see in the next section, however, this is less true as we increase the number of bits in the problem.

String	Proportion	Fitness
0000, 1111	0.303947	12
0011, 1100	0.0779981	8
0001, 0111, 1000, 1110	0.0402681	6
0010, 0100, 1011, 1101	0.0149496	6
0110, 1001	0.00533775	4
0101, 1010	0.00228135	4

Table 3: The string basis proportions in the limit distribution for the 4-bit HIFF problem. Note that strings with the same proportion are grouped together.

3.3 4-bit HIFF problem

Using techniques presented above, we were also able to iterate the schema equations for the 4-bit HIFF problem as well as solve for the limit distribution when using the balanced initial distribution described in Section 2.2. The formulas for the limit distribution consist of a large number of complicated rational functions, and we have not included them here because of space restrictions.

Solving for the limit distribution yields the proportions shown in Table 3 and in Figure 3. Note that the bulk of the system's "attention" is again focused on the two global optimum, with just over 60% of the population being instances of those two strings. This is, however, quite a bit less than the over 80% allocated to the global optima in the 2-bit case; the two second most fit strings, for example, collectively represent over 15% of the population.

Interestingly, not all strings with the same fitness have the same proportions, presumably due to differences in their likelihood of being constructed via crossover. For example, two of the four least fit strings (0110 and 1001) have proportions that are more than twice those of the other two least fit strings (0101 and 1010). The strings with the higher proportions, however, can be constructed in two crossover steps from instances of the most fit strings, while the two with the lower proportions require three crossover steps starting with optimal strings. Thus the difference in proportions is not surprising, as those with lower proportions require the construction and subsequent selection of two intermediate sub-optimal strings in order to be constructed. We discuss this phenomenon further in Section 3.4, where we use the concept of effective fitness to analyze our results, and in Sec-

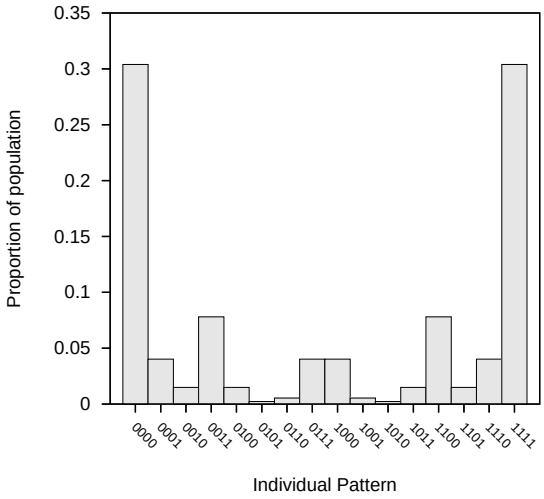

Figure 3: Limit distribution for the 4-bit HIFF problem.

tion 3.5, where we use statistical methods to explain nearly all of the limit distribution data.

Given that the limit distribution for the 2-bit HIFF problem turned out to be the function of a single independent variable, it seemed reasonable to look for that possibility in the 4-bit case as well. Unfortunately neither the string or the schema basis made the dimensionality of the problem clear. Calculations with Mathematica^{TM}, however, revealed that the proportions in the 4-bit case were again based on a single independent variable. Thus, if the proportion of the single independent schema in the limit distribution is known, one can compute the proportions of all the other strings and schemata. The formulas for doing this are quite complex, however, and are therefore not included.

3.4 Effective fitness analysis

In Sections 3.2 and 3.3 we mathematically solved for the limit distributions for the 2 and 4-bit HIFF problems, respectively. In this section, we will further examine these limit distributions using the concept of effective fitness, which was defined in Section 2.5.

In schema theory, a schema's transmission probability at time t ($\alpha(H,t)$, using the notation presented in Section 2.3) can be expressed as a function of effective fitness. From [7] we have:

$$\alpha(H,t) = \phi(H,t)\frac{f_{\text{eff}}(H,t)}{\overline{f}(t)} \tag{4}$$

As observed in Section 2.3, in an infinite population a schema's transmission probability at time t is equal to its proportion within the population at time $t+1$ (or, in our notation, $\alpha(H,t) = \phi(H,t+1)$). Based on Equation (4), we see that a schema's proportion within the population at time t will *always* be different at time $t+1$ unless $\alpha(H,t) = \phi(H,t)$. The only way for that condition to be met is to have $f_{\text{eff}}(H,t)/\overline{f}(t) = 1$. Put more simply, for the proportion of a schema to remain constant over time, the effective fitness of that schema must be equal to the average fitness of the population at time t.

In a limit distribution, by definition, none of the schema proportions change over time. The only way for this to be true is if every schema possesses the same effective fitness, and that effective fitness is equal to the average fitness within the population [10, 11].

We would then expect the effective fitness of each string to equal the average fitness in the population in the limit distributions found in earlier sections. This is in fact the case, confirming that we have successfully found limit distributions for both the 2 and 4-bit HIFF problems. In the 2-bit case each string's effective fitness is about 3.69, and in the 4-bit case each string's effective fitness is about 9.93. In both these cases these effective fitnesses were the average fitness of the population in the limit distribution.

We also see that, in the 2 and 4-bit cases, the effective fitness of the optimal strings is scaled down from their unmodified fitness because crossover is sufficiently destructive for these strings (and at these proportions) that there are fewer instances of those strings than one would expect with selection and no crossover. On the other hand, the fitness of *all* the other strings is scaled up in their effective fitnesses, as crossover constructs more instances of those strings than would be expected using selection alone. In the 4-bit case, for example, the strings 0011 and 1100 have the second highest fitness (8) and are easily constructed in a single crossover step from the optimal strings 0000 and 1111. This accounts for their reasonably high proportions, which are shown in Table 3 and Figure 3.

3.5 Statistical Analysis

There is clearly interesting structure in the limit distribution for the 4-bit HIFF problem, as discussed in Section 3.3. To gain further insight into this structure, we performed a variety of statistical analyses. Simple linear regression, using the proportion of each schema within the population as the response variable and their fitness as the predictor, produced an R^2 value[2] of about 0.9. We had expected fitness to play a large role in determining schema proportions in the limit distribution, and it did. On the other hand, it clearly didn't explain all the distribution data. Thus we chose to perform a more detailed statistical analysis in the hopes of explaining much of the remaining 10% of the data.

Figure 4 plots the fitness of each schema versus its proportion in the limit distribution, and it is clear that there is not a linear relationship between these values. This suggests that a simple linear model is insufficient for explaining this relationship, although correcting for this does not, in any of the cases examined in this study, change the results significantly.[3] This figure also indicates that there are strings that share the same fitness, but have different proportions in the limit distribution.

This then led us to an examination of other factors that might influence schema formation. Clearly, effective fitness is not useful as a predictor here, because all effective fitnesses

[2]An R^2 value is a statistical measure of how much of the response can be explained by the predictors. An R^2 of 1, for example, indicates that *all* the data is explained by the predictors.

[3]Correcting the model involved a number of linearizing transforms to both the response and the predictor. Further transforms were necessary to correct the multiple linear regression model used later, but once again they did not change the results significantly. The R^2 values between the corrected and uncorrected models differed by less than 0.0004.

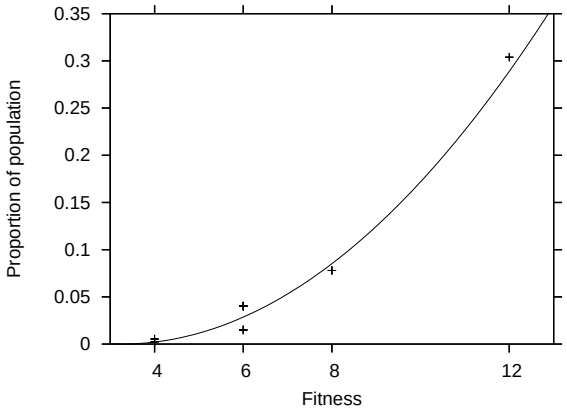

Figure 4: Fitness vs. Proportion for each string in the limit distribution of the 4-bit HIFF problem. The curve is a quadratic fit to the data points.

are equal in the limit distribution (see Section 3.4). The earlier results from Section 3.3 suggested we look at the "distance" of a string from the global optima in crossover space, i.e., the number of crossovers necessary to reach a global optimum.

In general this notion of distance is a complex concept which depends on a number of dynamic features such as the structure of the current population. We chose to approximate this distance with the number of *break points* each schema possessed. We define a break point to be a crossover point in a schema that separates a 1 and 0, or a 0 and 1. For example, the schema 1001 has two break points, while the schema 1010 has 3. Since both global optima have zero break points, and every (one point) crossover can eliminate at most one break point, the number of break points provides a simple lower bound on the number of crossovers necessary to construct a global optimum from a given string.

It seems likely, therefore, that schemata possessing equal fitnesses but different numbers of break points would have different proportions in the limit distribution. In particular, in a population dominated by the global optima, schemata with fewer break points should be easier to obtain than those with more break points. This conjecture is supported by the data in Table 3.

Adding the number of break points as a predictor did not significantly improve our model. However, when the interaction of fitness and break points were added as well, there was a tremendous improvement. The R^2 for the model including the interaction between fitness and break points was above 0.9986, indicating that it explains nearly one hundred percent of the proportions in the limit distribution. With the inclusion of break points, therefore, we are able to statistically explain nearly all of the limit distribution proportions obtained in Section 3.5.

4. FUTURE WORK

While this research generated a host of useful results, it also raises many questions and opens other doors. Possible extensions of this work include exploring cases of the HIFF problem with longer bit strings, further study of the role of dimensionality in the HIFF and other problems, comparing these theoretical results to empirical runs with finite populations, and applying these techniques to the randomized HIFF problem.

One obvious extension of this work is to explore larger instances of the problem. Our analysis of the 2 and 4-bit HIFF problems revealed a number of interesting features, and it would be valuable to see how these features do or do not extend to other instances.

In Sections 3.1 and 3.3 we found that both the distributions for both the 2 and 4-bit HIFF problem were essentially one-dimensional in that they were determined by a single value. This was easy to read from the schema proportions in the 2-bit case, but considerably more difficult to verify in the 4-bit case. An important question is whether more complex cases of the HIFF problem continue to have this property, since this might suggest that under an appropriate transformation HIFF problems are in fact reasonably simple. To help us better understand the structure of the HIFF problem, though, it would also be useful to better understand the *why* of this dimensionality. Is the HIFF problem part of a general class of "one-dimensional" problems, for example? Is there some shared structure in this class that can be used to help solve those problems?

Another important area of exploration is the relationship between the theoretical distributions generated with the schema equations under the assumption of an infinite population and the empirical results generated with finite populations. While we would obviously expect a finite population to behave differently from an infinite population, several studies have shown a strong relationship between theoretical predictions from the schema equations and empirical results for certain problem domains [8, 9, 6, 5]. Preliminary results with the HIFF problem, however, suggest (not surprisingly) that empirical runs converge quickly to a limit distribution where the entire population consists of copies of the same string. This suggests that the symmetric limit distributions found here are not stable, and that even if empirical runs are started with symmetric initial distributions, sampling error soon breaks the symmetry and the runs converge to a single string. Further study of this phenomena would help us better understand the role of sampling in finite populations, as well as shedding more light on the HIFF problem itself.

Finally, in [16] Watson suggests that the HIFF problem is easily solved by a GA provided that it has sufficient diversity (so that the necessary building blocks are present) and strong linkage (so the building blocks are transmitted effectively). He supports this in part through exploration of the randomized HIFF problem, where the the bits are shuffled to break up the linkage of the building blocks. An interesting extension of the work in this paper would be to apply these techniques to the randomized HIFF problem and other variants with weaker linkage. We should be able to shed light on the relationship between the linkage of a building block, its effective fitness, and its proportions in limit distributions.

5. CONCLUSIONS

One of the primary accomplishments of this paper is a deeper understanding of the HIFF problem. In Sections 3.2 and 3.3 we were able to find limit distributions for both the 2 and 4-bit HIFF problems, and analysis of those distributions showed that they were functions of a single independent variable. In the 2-bit case, we showed that the population reached the limit distribution very rapidly (in under

ten generations). The results presented here and work in progress suggest that as the number of bits in the problem are increased, the total proportion of individuals sampling the global optima in the limit distribution seems to decrease significantly. This finding helps to explain why the HIFF problem becomes dramatically more difficult as the size of the bit strings grow.

Examination of string proportions in the limit distributions revealed that strings with the same fitness did *not* always have the same proportions. To explain this behavior, we used statistical methods (in Section 3.5) to examine the limit distributions. Statistical analysis revealed that about ninety percent of the proportions can be explained by fitness alone. To explain the remaining ten percent, we introduced the concept of break points, which are a measure of the number of crossover steps needed to form a string from the global optima. With the inclusion of the interaction between fitness and break points, we statistically explain the limit distribution proportions nearly completely, showing that they are almost entirely dependent on these two predictors.

In addition to the numerous discoveries we made about the HIFF problem, in Section 3.1 we provide a detailed demonstration of how to apply schema theory to analyze a problem. We also illustrate how the choice of basis can simplify schema calculations, as suggested in [1]. While we were able to solve for exact limit distributions in the 2 and 4-bit cases using these techniques, it seems likely that the increasing complexity of higher bit cases will present significant scaling problems for this approach. Given that the number of schemata (independent of the basis chosen) is 2^n, where n is the number of bits, additional coarse graining is going to be necessary if we are to apply this kind of analysis to more complex problems.

We also saw in Section 2.3 that the exact proportions generated by the schema equations are rational functions that grow increasingly complex as the generations proceed. This puts bounds on our ability to compute exact proportions over large numbers of generations. If we're comfortable with floating point approximations this restriction goes away, but we need to be aware of the possibility of these approximations drifting away from the true values.

Despite these concerns about scaling, it's clear that we were able to generate useful results on these smaller problems, and that those results have value beyond the particular problem. Our analysis shows that we can solve for limit distributions in simplified cases, and that the populations (under the infinite population assumption required by our analysis) approach these limits very quickly. Further, while the proportions of strings in the limit distribution are driven in significant part by their fitness, it's clear that other dynamics play a significant role, and a role that appears to grow as the problems become more complex.

6. REFERENCES

[1] C. Chryssomalakos and C. R. Stephens. What basis for genetic dynamics? In K. Deb et al., editors, *GECCO (1)*, volume 3102 of *Lecture Notes in Computer Science*, pages 1018–1029. Springer, 2004.

[2] D. E. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading Massachusetts, 1989.

[3] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.

[4] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.

[5] N. F. McPhee, A. Jarvis, and E. F. Crane. On the strength of size limits in linear genetic programming. In K. Deb et al., editors, *GECCO (2)*, volume 3103 of *Lecture Notes in Computer Science*, pages 593–604. Springer, 2004.

[6] N. F. McPhee and R. Poli. Using schema theory to explore interactions of multiple operators. In W. B. Langdon et al., editors, *GECCO*, pages 853–860. Morgan Kaufmann, 2002.

[7] R. Poli. Exact schema theorem and effective fitness for GP with one-point crossover. In L. D. Whitley et al., editors, *GECCO*, pages 469–476. Morgan Kaufmann, 2000.

[8] R. Poli and N. F. McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part I. *Evolutionary Computation*, 11(1):53–66, 2003.

[9] R. Poli and N. F. McPhee. General schema theory for genetic programming with subtree-swapping crossover: Part II. *Evolutionary Computation*, 11(2):169–206, 2003.

[10] C. R. Stephens and J. M. Vargas. Effective fitness as an alternative paradigm for evolutionary computation I: General formalism. *Genetic Programming and Evolvable Machines*, 1(4):363–378, 2000.

[11] C. R. Stephens and J. M. Vargas. Effective fitness as an alternative paradigm for evolutionary computation II: Examples and applications. *Genetic Programming and Evolvable Machines*, 2(1):7–32, 2001.

[12] C. R. Stephens and H. Waelbroeck. Effective degrees of freedom in genetic algorithms and the block hypothesis. In T. Bäck, editor, *ICGA*, pages 34–40. Morgan Kaufmann, 1997.

[13] C. R. Stephens and H. Waelbroeck. Schemata evolution and building blocks. *Evolutionary Computation*, 7(2):109–124, 1999.

[14] C. R. Stephens, H. Waelbroeck, and R. Aguirre. Schemata as building blocks: Does size matter? In W. Banzhaf and C. R. Reeves, editors, *FOGA*, pages 117–133. Morgan Kaufmann, 1998.

[15] R. A. Watson. Modular interdependency in complex dynamical systems. In E. Bilotta et al., editors, *Workshop Proceedings*. 8th International Conference on the Simulation and Synthesis of Living Systems, December 2002.

[16] R. A. Watson, G. S. Hornby, and J. B. Pollack. Modeling building-block interdependency. In *PPSN V*, 1998.

[17] R. A. Watson and J. B. Pollack. Hierarchically-consistent test problems for genetic algorithms. In Angeline, Michalewicz, Schoenauer, Yao, and Zalzala, editors, *Proceedings of 1999 CEC*, pages 1406–1413. IEEE Press, 1999.