

Modeling Building-Block Interdependency

Richard A. Watson Gregory S. Hornby Jordan B. Pollack

Dynamical and Evolutionary Machine Organization Group
Volen Center for Complex Systems – Brandeis University – Waltham, MA – USA
richardw@cs.brandeis.edu

Abstract. The Building-Block Hypothesis appeals to the notion of problem decomposition and the assembly of solutions from sub-solutions. Accordingly, there have been many varieties of GA test problems with a structure based on building-blocks. Many of these problems use deceptive fitness functions to model interdependency between the bits within a block. However, very few have any model of interdependency between building-blocks; those that do are not consistent in the type of interaction used intra-block and inter-block. This paper discusses the inadequacies of the various test problems in the literature and clarifies the concept of building-block interdependency. We formulate a principled model of hierarchical interdependency that can be applied through many levels in a consistent manner and introduce *Hierarchical If-and-only-if* (H-IFF) as a canonical example. We present some empirical results of GAs on H-IFF showing that if population diversity is maintained and *linkage* is tight then the GA is able to identify and manipulate building-blocks over many levels of assembly, as the Building-Block Hypothesis suggests.

1 Introduction

Problem decomposition – the method of forming solutions by first breaking down a problem into sub-problems – is a central tenet behind the Building-Block Hypothesis ((Holland, 1975), (Goldberg, 1989)). Yet there has been much controversy over whether the GA can identify and manipulate sub-solutions or building-blocks appropriately ((Forrest & Mitchell, 1993b), (Goldberg and Horn, 1994), (Mitchell *et al.*, 1995)). Much of this controversy arises from the inadequacy of test problems in the literature and in turn this inadequacy arises from lack of clarity on the seemingly straight-forward concept of problem decomposition. More specifically, although problem decomposition implies some form of modularity it does not require that sub-problems be independent from each other. However, most building-block problems in the literature have no model of dependency between the building-blocks. This paper offers clarification on the issues of problem decomposition and in particular discusses how a problem may be decomposable into sub-problems and yet have strong non-linear interdependencies between these sub-problems. We define a construction method for a new class of problems, hierarchically decomposable problems, which exemplify those for which GAs should be well-suited according to the intuition of the Building-Block Hypothesis.

Before we proceed further we must define some terms. Although the basic concept of decomposing a problem into sub-problems may seem clear there are many ways that our straight-forward interpretation may be inadequate. Firstly, we clarify that a *separable*

problem is a problem which can be divided into sub-problems each of which has a fixed optimal solution that is independent of how other sub-problems are solved. We define a *decomposable* problem to be something more general: simply a problem that has sub-problems, and we specifically allow that these sub-problems need not be separable - i.e. the optimal solution to one sub-problem may be different according to the solution of another sub-problem. In this case we say that a sub-problem is *dependent* on another, or in the bi-directional case, the sub-problems are *interdependent*. Included in the term decomposable is the notion of identifiable component parts, i.e. *modularity*, so a system of uniformly related variables with no modular structure is excluded from the term decomposable. Next we look at the notion of hierarchy. The Building-Block Hypothesis describes the assembly of bits into blocks, blocks into bigger blocks, and so on to provide a complete solution - the hierarchical aspect of the process is clear in our intuitions. Our general notions of GAs, building-blocks and the Building-Block Hypothesis all imply that the process continues over many levels *in the same fashion*. We define a *consistent* hierarchical problem as one where the nature of the problem at all levels is the same. In a consistent problem, the difficulty of solving a sub-problem and the difficulty of solving the whole problem (given the solutions to the sub-problems) should be the same. This will be a central characteristic of our new problem class.

The following section discusses the properties of test problems in the literature in the light of the concepts defined above. Section 3 describes the general class of functions that can be used to construct principled decomposable problems, and 3.1 introduces the *Hierarchical If-and-only-if* (H-IFF) function as the canonical example of this class. Section 4 discusses the properties of H-IFF and adds justification for our construction method. Section 5 describes empirical results of GA trials using H-IFF.

2 Existing test problems from the GA literature

Before introducing our new problem class we first review some common test problems from the GA literature. We will be looking for a consistent hierarchical problem that exhibits modular sub-problems and interdependency between these sub-problems.

Whitley *et al.* (1995 and 1995b), provide a review of test-problems from the GA literature and summarize, “most common test problems used in the past to evaluate genetic algorithms lack” the inclusion of “strong non-linear dependencies between state variables.” That is, although, the contribution of a single variable to the overall evaluation may be non-linear, there may be no non-linear interactions *between* variables, and therefore the optimal value for each parameter can be determined independently of all other parameters.

There are several building-block style functions - the Royal Road (RR) functions (Mitchell *et al.*, 1992, Forrest & Mitchell, 1993), revised Royal Road functions (Holland 1993), (see also, Jones, 1995), concatenated trap functions (Deb & Goldberg, 1992), and others - that clearly emphasize a gross-scale building-block structure. But, like the real-valued functions that Whitley investigated, these consist of concatenated blocks, each of which may be optimized independently in a cumulative fashion. The R1 version of the Royal Roads problem (Forrest & Mitchell, 1993), for example, can be imagined as a staircase leading search in a stepwise fashion in the correct direction. In concatenated trap functions and the revised Royal Road functions, the fitness gradient leads search away from the solution to each block. But, whereas the bit-wise landscape is fully deceptive,

the 'block-wise landscape' is fully non-deceptive. That is, within a block the bits are not separable but each block is still separable from every other block (i.e. the optimal setting for a bit remains the same regardless of the setting of bits in other blocks) and again the sub-solutions accumulate linearly. To continue the analogy, although each tread on the stair is inclined unfavorably the whole staircase still leads to the global optima.

The R2 function, (Forrest & Mitchell, 1993), and the revised Royal Road functions make hierarchy explicit. However the interactions between bits within the first level blocks are very different from the interactions between blocks in subsequent levels. Although the fitness contributions of the base-level blocks accumulate non-linearly because of 'bonus' contributions from higher levels the blocks are still separable in that the optimal bit-setting for a block does not change.

Goldberg *et al.* (1993) acknowledge that they have "temporarily ignored the possibility of subfunction crosstalk". *Crosstalk*, is defined by Kargupta (1995) as "a property of both the algorithm and the landscape" and appeals to the idea that the algorithm may be misled in identifying building-blocks by changes in fitness contribution. Accordingly, Goldberg *et al.* suggest that noise in the fitness function may be an appropriate model for crosstalk. But this interpretation of crosstalk is not interdependency as we have defined it. We favor a definition of building-block dependency that is not reliant on the algorithm being used - specifically, the observation that the optimal bit-setting for a block is different in one context than in another is algorithm independent.

Whitley *et al.* (1995) argue for test-problems that have non-separable components, and propose an "expansion" method for constructing scaleable non-separable functions from a non-linear base function of two variables. By selecting pairs of variables in an overlapping fashion and summing the results they create a non-separable function over an arbitrarily large set of variables. They also propose that using all possible pairs, rather than a minimal set of overlapping pairs, provides a method to vary the amount of interdependency. This notion has some similarities with the NK landscapes of Kauffman (1993). The NK landscapes certainly do model the interdependency of variables - the contribution of each of N variables is defined as a random look-up function of itself and k other variables. The expanded functions of Whitley are analogous to NK landscapes with $k=2$ (and with a repeated base-function instead of random lookup tables). Using additional pairs is not wholly different from increasing k. This seems like a step in the right direction especially since the value of k, in some sense, defines the degree of interdependence.

But this is still not wholly satisfying for now we cannot see the building-blocks. In the NK model (and the expanded function method) the dependencies form a uniform network; there is no modularity. Our preliminary work explored varieties of the NK model that cluster the variables into groups with many intra-group dependencies and few inter-group dependencies- an idea inspired by the dependency matrices used by Simon (1969). Goldberg *et al.* (1993), also suggests adding changes in fitness according to "the presence or absence of bit combinations across subfunction boundaries". But this method does not provide a principled hierarchical structure - merely a clustered, single-level one. To be consistent the interdependency of building-blocks should be defined as a function of the blocks as wholes not as a function of particular bits within the blocks.

Thus, none of these problems from the literature model building-block interdependency in a consistent hierarchical fashion.

3 Hierarchically decomposable problems

In this section we introduce a recursive construction for defining hierarchically consistent decomposable fitness functions from two base functions. Section 3.1 gives our canonical example, Hierarchical-if-and-only-if (H-IFF). This enables the construction of a problem, reminiscent of R2, that exhibits a gross-scale hierarchical building-block structure. But unlike R2, the blocks in this function are not separable.

The construction we will use defines the fitness contribution of a building-block separately from the *meaning* of the block. The meaning of a block is given by a *transform* function that defines what kind of block results from a particular combination of sub-blocks. The transform function ‘decodes’ or ‘interprets’ a block, and it is this that defines the interdependencies between sub-blocks. By making the definition of interdependencies separate from the fitness contributions we will be able to define two (or more) blocks that are indistinguishable from each other in respect of their individual fitness contributions yet which may have different results when put together with other blocks at the next level of the hierarchy.

A string of symbols $B = \{b_1, \dots, b_n\}$, over an alphabet S , where $n=k^p$, represents a hierarchical block structure, where k is the number of sub-blocks in a block, and p is the number of levels in the hierarchy. Each block at the bottom level of this hierarchy, consisting of k symbols each, will be converted into a single symbol by a *transform* function, t . This transform function defines the meaning of each block. This creates a new string (with length $k^{(p-1)}$) that is the decoding of the block structure to the first level. This process is repeated for each level in the hierarchy to give the single symbol that is the meaning of the whole structure. Thus, the recursive transform function, T , transforms any block structure to its meaning, a single symbol:

$$T(B) = \begin{cases} b_1 & \text{if } |B|=1, \\ t(T(B^1), \dots, T(B^k)) & \text{otherwise.} \end{cases}$$

where t is a base function, $t:S^k \rightarrow S$, that defines the resultant symbol from a block of k symbols, $|B|$ is the number of symbols in B , and B^i is the i^{th} sub-block of B i.e. $\{b_{(i-1)d+1}, \dots, b_{id}\}$ (where $d=|B|/k$).

Now we may use $T(B)$ to construct $F(B)$, the fitness of a block structure. Specifically, the fitness of a block structure will be the fitness contribution of its transform (scaled for its size) plus the sum of the fitnesses of its sub-blocks. Hence the recursive function, F , defined using the base function f :

$$F(B) = \begin{cases} f(B) & \text{if } |B|=1, \\ |B| f(T(B)) + \sum_{i=1}^k F(B^i) & \text{otherwise.} \end{cases}$$

where f is a base function, $f:S \rightarrow \mathfrak{R}$, giving the fitness of a single symbol.

The recursive constructions ensure that the fitness function is hierarchical and consistent. To illustrate, we can 'unroll' $F(B)$ for a four variable problem with blocks formed from pairs (i.e. $k=2, p=2$) to give:

$$F(a,b,c,d) = 4f(t(t(a,b),t(c,d))) + 2f(t(a,b)) + 2f(t(c,d)) + f(a) + f(b) + f(c) + f(d).^1$$

3.1 Hierarchical-If-and-only-if (H-IFF)

We now need to give examples of the base functions, f and t , that provide an interesting fitness landscape. Our canonical example is provided for a $k=2$ problem by the logical relation if-and-only-if (IFF), or equality. First we define $t(\{A,B\})$ by arbitrarily assigning 0 and 1 to the two solutions of IFF (i.e. $t(\{0,0\})=0$ and $t(\{1,1\})=1$), and null to the non-solutions (i.e. $t(\{0,1\})=t(\{1,0\})=null$). But since t should map $S^k \rightarrow S$ so that it can be used in the recursive construction, we must expand its definition to a pair of tertiary variables $\{0,1, \text{and null}\}$ instead of binary variables (Table 1). Then $f(A)$ naturally defines the two non-null transform values as desirable and null as undesirable, (Table 2). $F(B)$, $T(B)$ and our two base functions, f and t , complete our definition of Hierarchical-If-and-only-if (H-IFF).

A	B	$t(\{A,B\})$
0	0	0
0	-	-
0	1	-
-	0	-
-	-	-
-	1	-
1	0	-
1	-	-
1	1	1

Table 1, left, and **Table 2**, right, define the two base functions, f and t , for Hierarchical-IFF. (nulls are shown as "-")

A	$f(A)$
0	1
-	0
1	1

4 Properties of H-IFF: deception and dimensional reduction

Our general construction and our choice of base functions warrants some explanation, but let us first examine some of the interesting properties of H-IFF. Blocks of two bits assemble into blocks of four, then eight, and so on. There are two solutions to every block and, similarly, two global optima. These solution pairs for each block are always maximally distinct, i.e. all-ones or all-zeros. Also, notice that any setting of any bit can be included in one of the two global optima – there is no preference for any particular bit settings *per se*, only for particular *combinations* of bit settings. Further, suppose the

¹ This can be contrasted with the R1 function and the concatenated trap functions which have the form $F(a,b,c,d)=f(a,b)+f(c,d)$, which is separable and has only one level; and it is also distinct from the R2 function which has the form: $F(a,b,c,d)= f(a,b,c,d)+g(a,b)+g(c,d)$, which is multi-level but not consistent; and distinct from Whitley's expanded functions of the form: $F(a,b,c,d)= f(a,b)+f(b,c)+f(c,d)+f(d,a)$, which is non-separable and shows some consistency but is not hierarchical or modular; and lastly distinct from the N_k functions which have the form $F(a,b,c,d)= f_1(a,b)+f_2(b,c)+f_3(c,d)+f_4(d,a)$, which is like Whitley's function but less consistent.

fitness base function were modified so that $f(0)=0$ – in this case a search would find a staircase in the fitness landscape that would lead to the global optimum at all-ones (see Figure 1, “all null to all 1s”). Notice the small steps in fitness at each block of 4, and larger jumps at each block of 8, etc. (similar to the higher levels of R2). Similarly, a corresponding curve exists for the all-zeros solutions if $f(1)=0$ (with $f(0)=1$ and $f(\text{null})=0$ as before). To illustrate that the all-zeros maximum is mutually exclusive with the all-ones maximum it is shown at the opposite extreme. But, in H-IFF, both blocks of zeros and blocks of ones are rewarded, and the “all 0s to all 1s” curve indicates the combination of the previous two. We can see that the best local-optima (with 32-ones and 32-zeros) are maximally distant from both global optima, and the next best are maximally distant from them and the global optima, and so on. This, as we will demonstrate, makes H-IFF very hard for hill-climbers.

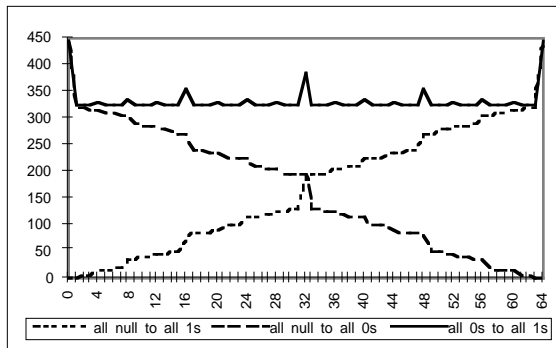


Figure 1: Three sections through the 64-bit H-IFF landscape (as used in the following experiments) showing two maximally distinct global optima (max. fit = 448) and illustrating the fractal structure for the local optima arising from the (consistent) hierarchical interdependencies. Bear in mind that we are only viewing 65 points out of a total 2^{64} . See text.

Consider the local optimum 00001111 for an 8-bit H-IFF. This string confers fitness for each of the two sub-blocks, 0000 and 1111, but no fitness contribution for the top level. Each substring that is rewarded is a substring of a global optima and *in this sense* the fitness contributions are not ‘misleading’. Deceptive fitness functions such as the concatenated trap functions are conceptually different. These fitness functions create artificial local optima that represent erroneous elements of a fitness function. They are completely (and deliberately) wrong in their assessment of low-order schema, but improve as schema increase in size. It does not make sense to apply this kind of misleadingness consistently through many levels of a hierarchy - the fitness function cannot be completely wrong at the top level.

A more informative point of view with which to understand the problem difficulty is to imagine that the fitness of a local optima is not so much ‘wrong’ as ‘incomplete’. That is, it does not apply any knowledge of subsequent levels - substrings are rewarded for what they are, even if they are not used in the best context. From this perspective the fitness contribution of a block may be misleading not because of its incompatibility with the global optima but rather because it is incompatible with its context.

How is it that we can make sense of a fitness contribution when it does not take account of context? After all, in H-IFF, a block might be completely right for one context and completely wrong for another! The information provided by the fitness contributions enables search to identify how to reduce the dimensionality of the problem space. In this manner search can progress from the combination of bits to the combination of low-order schema, to the combination of higher-order schema, and so on, exactly as the building-block hypothesis suggests. This does not mean that finding the right combinations will not be problematic at each and every level - it simply means that

at each level the search process has bigger building-blocks to experiment with. This dimensional reduction is the only way to make sense of problem decomposition when sub-problems are strongly interdependent. If there is only one solution to a building-block then finding the right combination of blocks cannot be problematic. Hence, we arrive at a fundamental requirement for a (non-trivial) decomposable problem - there has to be *more than one way to solve a (sub-)problem*. This is the motivation that leads us to conceptually separate the way in which a problem is solved from the fact that the solution is good - to separate meaning from fitness. It is, perhaps, the lack of this simple distinction between meaning and fitness, and the resultant tendency to try and define interdependency using functions that manipulate fitness values, that has prevented the prior formulation of a principled model for building-block interdependency.

To represent the two ways to solve a problem and one non-solution we need an alphabet of three symbols and accordingly f has two high-fitness entries and one low-fitness entry. Then we must ensure that these two high-fitness blocks arise from certain combinations of the high-fitness sub-blocks but not all combinations - hence, t is based on IFF where two out of four possible combinations provide non-nulls for use in the next level. Logical exclusive-or (being the negation of IFF) will suffice equally well in formulating a Hierarchical-XOR². Other Boolean functions of two variables are not suitable since they either have only one solution, or solutions dependent on only one variable. Other base functions over larger alphabets, and different values of k and p , may be used to produce asymmetric, biased, or possibly noisy fitness landscapes that have a consistent hierarchical decomposability.

5 Experiments using H-IFF

The following experiments use H-IFF to verify the GA's ability to identify building-blocks, resolve their interdependencies, and (re)combine them, as described by the Building-Block Hypothesis. Forrest and Mitchell (1993b, and Mitchell *et al*, 1995) use random mutation hill-climbing (RMHC) to demonstrate that the RR functions are not appropriate test-functions for GAs. Likewise, Jones (1993 pp. 57-58) demonstrates that the revised RR-functions and the concatenated trap-functions are inappropriate by applying the "Headless Chicken Test" (HCT) (a hill-climber that uses macro-mutations). Thus RMHC and HCT are chosen for comparison against the standard GA (both two-point and uniform crossover). These algorithms are each applied to a standard 64-bit H-IFF and a bit-shuffled H-IFF, both with and without fitness-sharing. The bit-shuffled H-IFF (a standard H-IFF with random bit-ordering) demonstrates that the GA's ability to manipulate building-blocks with crossover is dependent on the heuristic of bit-adjacency (see linkage disequilibrium, Altenberg 1995). The fitness-sharing method is used to enforce diversity since any algorithm that is to explore the space of solutions properly will need to maintain both solutions to a block at least as long as it takes to find the solution at the next level. We use a resource-based fitness-sharing method (similar to implicit fitness sharing) that uses explicit knowledge of the block-structure, and is

² It was tempting to give H-XOR as the canonical example of the class since this name perhaps reveals more of the difficulty in the resultant landscape - however, although the problem difficulty of hierarchical-equality (H-IFF) is exactly the same as that of hierarchical-inequality (H-XOR), a solution like 11111111 is much easier on the eye than 01101001.

therefore not a general purpose fitness-sharing method. Rather, it simply provides a tool for enforcing diversity, so that we can investigate the GA's ability to combine building-blocks when diversity is maintained. All four algorithms (1-4 detailed below) are applied to each of the 4 different varieties of H-IFF (a - d) in Figure 2.

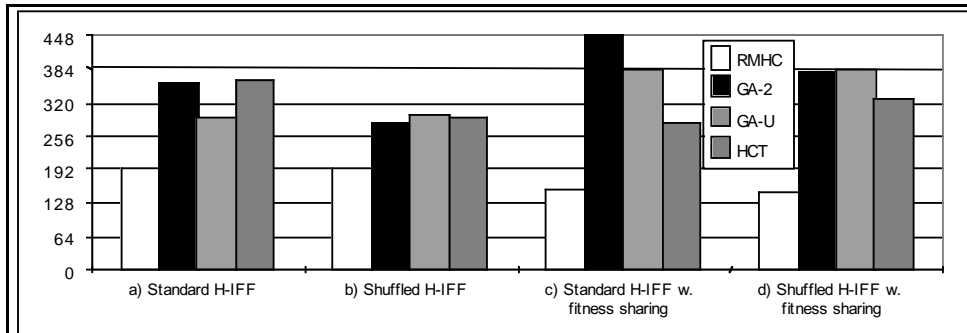


Figure 2: Random Mutation Hill Climbing (RMHC), GA with 2 point crossover (GA-2), GA with uniform crossover (GA-U), and the Headless Chicken Test (HCT), each applied to a) Standard 64-bit H-IFF, b) Shuffled H-IFF (H-IFF with random bit-ordering), c) Standard H-IFF with fitness sharing, d) Shuffled H-IFF with fitness sharing. Each point is the best fitness from the last generation, averaged over 10 runs. In all runs there was no significant improvement in the best fitness after approximately 400 generations. A 64-bit H-IFF is used which has a maximum fitness solution of 448. a) As expected, RMHC performs poorly, becoming trapped in local optima. GAs do better but also become trapped. Although the 2-point cross-over performs better than uniform, the HCT suggests that the GA gets as far as it does because of macro mutations rather than the combination of pre-adapted blocks. From comparison with the HCT performance we see that the GA performance is not very impressive. b) When the bits of a H-IFF problem are randomly shuffled the problem is even harder - all algorithms perform as poorly as uniform cross-over performed on the standard H-IFF. This strengthens the hypothesis that the GAs representation of blocks in the standard H-IFF does add to the GA performance. c) When we enforce diversity in the population the standard 2 point GA succeeds easily. The HCT shows that the success of the regular GA is not because of macro mutations (as it is in the previous two graphs). The hill-climber performs even worse now; when it finds one block the resources are depleted and it has to move to another - so, although it can escape from local optima, the next block it finds is usually no better. This and the HCT confirms that the fitness-sharing methods did not trivialize the problem, and that the population of the GA is necessary. d) demonstrates that the heuristic of bit-adjacency in identifying building-blocks is required for GA success as expected. Poor linkage reduces the performance of the GA with 2 point crossover to the performance of uniform crossover.

- 1) **RMHC: Random Mutation Hill-Climbing:** Mutation rate is $2/64$ probability of assigning a new random value. RMHC is run for 10^6 evaluations.
- 2) **GA-2: GA with 2 point crossover:** 1000 individuals are run for 1000 generations. Elitism of 1% is used. The ranking function uses Michalewicz's exponential scaling with 0.01 for the value that determines the selection pressure (Michalewicz, 1995). Two-point crossover is used with a probability of 0.3. Mutation as per RMHC.
- 3) **GA-U: GA with uniform crossover:** Other parameters as per GA-2.

- 4) **HCT: Headless Chicken Test:** The headless chicken test (Jones, 1995) is a GA in all respects except that when cross-over is performed one of the parents is an entirely random string which thus replaces whole chunks of the genotype with random bits whilst leaving other sections intact. If a regular GA performs no better than the HCT then the crossover operator is not providing any useful advantage. All other parameters are the same as GA-2.

The results in Figure 2 show that the GA is able to identify building-blocks and combine them appropriately as the Building-Block Hypothesis suggests. The results demonstrate that the success of the GA is not attainable with RMHC or with macro-mutation hill-climbing (HCT). They also show that this success is dependent on maintaining diversity and on good linkage. Together these results indicate that H-IFF is exemplary of the class of functions that GAs are good for.

6 Conclusions

This paper has reviewed the inadequacies of current GA test-problems. In particular, the emphasis has been on the modeling of problems with a hierarchical decomposable structure where sub-problems are not separable. The concept of a transform function that can be applied recursively to construct a multi-level problem with strong non-linear interdependencies is identified and separated from the concept of a fitness function. The recursive constructions can be parameterized by the size of the alphabet, the number of sub-blocks per block, and the number of hierarchy levels. Also, the characteristics of the landscape can be modified by the choice of base functions. The Hierarchical-IF-and-only-IF function is introduced as a canonical example of the class.

The experimental results using H-IFF demonstrate that the GA can assemble blocks hierarchically, as per the Building-Block Hypothesis, but only when allowed to use the heuristic of bit-adjacency and when diversity can be maintained. This indicates that H-IFF is exemplary of the class of problems that GAs are good for. Also H-IFF provides a good test-problem for investigating linkage-learning algorithms and fitness-sharing methods.³

We have seen that interdependency (where the contribution of a block may be misleading because of its incompatibility with other blocks) may be a more fruitful perspective on GA-difficulty than deception (based on incompatibility with a global optimum). Also, H-IFF demonstrates that interdependency can be modeled deterministically and that noisy fitness functions are not required for modeling building-block interdependency. Lastly, H-IFF is highly multi-modal but is amenable to GAs - which verifies that the structure of a landscape is more important than the number of local-optima (Goldberg *et al.* 1994). So although H-IFF is not a complete model of problem difficulty it is relevant to the issues of deception, noise, and multi-modality as well as building-block interdependency.

³ Although this paper has focused on introducing H-IFF, other research has developed a variant of the Messy GA which (unlike the algorithms tested here) solves the shuffled H-IFF in approximately 500 generations. However, this is a rather non-standard algorithm (interestingly, we only use a 'splice' genetic operator with no mutation or cross-over) and it is not yet clear which of the new features are essential to success.

Acknowledgments

The first author thanks Inman Harvey for providing the impetus to pursue these concepts and laying the challenge to ground them in a concrete model. Many thanks also to Una-May O'Reilly for her valuable clarifying discussion and assistance in reviewing the literature. To the members of the DEMO group, especially, Sevan Ficici and Pablo Funes appreciation for assisting us in the research process. And, thanks to the reviewers for their suggestions.

References

1. Altenberg, L, 1995 "The Schema Theorem and Price's Theorem", FOGA3, editors Whitley & Vose, pp 23-49, Morgan Kauffmann, San Francisco.
2. Deb, K & Goldberg, DE, 1989, "An investigation of Niche and Species Formation in genetic Function Optimization", ICGA3, San Mateo, CA: Morgan Kauffman.
3. Deb, K & Goldberg, DE, 1992, "Sufficient conditions for deceptive and easy binary functions", (IlliGAL Report No. 91009), University of Illinois, IL.
4. Forrest, S & Mitchell, M, 1993 "Relative Building-block fitness and the Building-block Hypothesis", in Foundations of Genetic Algorithms 2, Morgan Kaufmann, San Mateo, CA.
5. Forrest, S & Mitchell, M, 1993b "What makes a problem hard for a Genetic Algorithm? Some anomalous results and their explanation" Machine Learning 13, pp.285-319.
6. Goldberg, DE, 1989 "Genetic Algorithms in Search, Optimisation and Machine Learning", Reading Massachusetts, Addison-Wesley.
7. Goldberg, DE, & Horn, J, 1994 "Genetic Algorithm Difficulty and the Modality of Fitness Landscapes", in Foundations of Genetic Algorithms 3, Morgan Kaufmann, San Mateo, CA.
8. Goldberg, DE, Deb, K, Kargupta, H, & Harik, G, 1993 "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy GAs", IlliGAL Report No. 93004, U. of Illinois, IL.
9. Goldberg, DE, Deb, K, & Korb, B, 1989 "Messy Genetic Algorithms: Motivation, Analysis and first results", Complex Systems, 3, 493-530.
10. Holland, JH, 1975 "Adaptation in Natural and Artificial Systems", Ann Arbor, MI: The University of Michigan Press.
11. Holland, JH, 1993 "Royal Road Functions", Internet Genetic Algorithms Digest v7n22.
12. Jones, T, 1995, Evolutionary Algorithms, Fitness Landscapes and Search, PhD dissertation, 95-05-048, University of New Mexico, Albuquerque. pp. 62-65.
13. Jones, T, & Forrest, S, 1995 "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms" ICGA 6, Morgan & Kauffman.
14. Kauffman, SA, 1993 "The Origins of Order", Oxford University Press.
15. Michalewicz, Z, 1992, "Genetic Algorithms + Data Structures = Evolution Programs" Springer-Verlag, New York, 1992.
16. Mitchell, M, Holland, JH, & Forrest, S, 1995 "When will a Genetic Algorithm Outperform Hill-climbing?" to appear in Advances in NIPS 6, Morgan Kaufmann, San Mateo, CA.
17. Mitchell, M, Forrest, S, & Holland, JH, 1992 "The royal road for genetic algorithms: Fitness landscapes and GA performance", Procs. of first ECAL, Camb., MA. MIT Press.
18. Simon, HA, 1969 "The Sciences of the Artificial" Cambridge, MA. MIT Press.
19. Smith, RE, Forrest, S, & Perelson, A, 1993 "Searching for Diverse, Cooperative Populations with Genetic Algorithms", Evolutionary Computation 1(2), pp127-149.
20. Whitley, D, Mathias, K, Rana, S & Dzuberka, J, 1995 "Building Better Test Functions", ICGA-6, editor Eshelman, pp239-246, Morgan Kauffmann, San Francisco.
21. Whitley, D, Beveridge, R, Graves, C, & Mathias, K, 1995b "Test Driving Three 1995 Genetic Algorithms: New Test Functions and Geometric Matching", Heuristics, 1:77-104.