The Defined Cliffs Variant in Dynamic Environments A Case Study Using the Shaky Ladder Hyperplane-Defined Functions

Abir Alharbi
Mathematics Department, King Saud University, 11495 Riyadh, Saudi Arabia
a_alharbi@mail.com

William Rand
Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL, USA 60208
wrand@northwestern.edu

Rick Riolo
Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI, USA 48109
rlriolo@umich.edu

ABSTRACT

The shaky ladder hyperplane-defined functions (sl-hdfs) are a test suite utilized for exploring the behavior of the genetic algorithm (GA) in dynamic environments. This test suite can generate arbitrary problems with similar levels of difficulty and it provides a platform for systematic controlled observations of the GA in dynamic environments. Previous work has found two factors that contribute to the GA's success on sl-hdfs: (1) short initial building blocks and (2) significantly changing the reward structure during fitness landscape changes. Therefore a test function that combines these two features should facilitate even better GA performance. This has led to the construction of a new sl-hdf variant, "Defined Cliffs," in which we combine short elementary building blocks with sharp transitions in the environment. We examine this variant with two different levels of dynamics, static and regularly changing, using four different metrics. The results show superior GA performance on the Defined Cliffs over all previous variants (Cliffs, Weight, and Smooth). Our observations and conclusions in this variant further the understanding of the GA in dynamic environments.

Categories and Subject Descriptors

F.2. [**Analysis of Algorithms**]: Misc. I.2.8 [**Artificial Intelligence**]: Search

General Terms

Algorithms

Keywords

Genetic Algorithms, Dynamic Environments, Shaky Ladder Hyperplane-Defined Functions, Building Blocks

1. INTRODUCTION

To facilitate controlled observations on the GA in dynamic environments, a test suite of problems is necessary. A test suite allows us to control the inputs to the system and define metrics for the outputs. Moreover, the more parameters of the system (e.g. time and severity of shakes, difficulty of the problem) that are controllable, the easier it is to test explanations for the observed behavior. The test functions that we use to explore the GA in dynamic environments, the shaky ladder hyperplane-defined functions (sl-hdfs) [3], are a subset of the hdfs [2]. Holland created these functions in part to meet criteria developed by Whitley [9]. Other test suites for EAs in dynamic environments exist, such as the dynamic knapsack problem, the moving peaks problem and more [1], but these other suites are primarily concerned with comparing absolute performance of different EA variants. The hdfs, on the other hand, are designed to represent the way the GA searches by combining building blocks, and thus are well suited for understanding the operation of the GA [4] [5] [6] [8]. Previously, work on sl-hdfs [4] [5] [6], has presented three ways of constructing the sl-hdfs (called "variants") by manipulating the way building blocks are constructed, combined, and changed. These different variants were called the Cliffs, Weight, and Smooth variants. Here we introduce a new variant which we call Defined Cliffs. This variant has defined short building blocks but combines them in complex and unrestricted ways. We begin by reviewing the sl-hdfs and the previous three variants. We then describe the new variant, and compare it to the previous variants using a variety of measurements. We examine the behavior of the GA on this new variant, discuss the results and draw some conclusions.

2. SHAKY LADDER HYPERPLANE-DEFINED FUNCTIONS AND VARIANTS

In this section we describe the sl-hdfs and the variants we will be exploring. For an in-depth explanation of the construction of the sl-hdfs see [6]. As mentioned, the sl-hdfs are a subset of Holland's hdfs [2], however, to make the hdfs usable as a test platform in dynamic environments we place three restrictions on the hdfs: (1) The Unique Position Condition (UPC), which requires that all elementary schemata contain no conflicting bits; (2) The Unified Solution Condition (USC), which guarantees that all of the specified bits in the positive-valued elementary level schemata must be present in the highest level schema, and that all intermediate schema are a composition of lower level schema; and (3) The Limited Pothole Cost Condition (LPCC), which states that the fitness contribution of any pothole plus the sum of the fitness contributions of all the building blocks in conflict with that pothole must be greater than zero. These three conditions guarantee that any string that matches the highest level schema must be optimally valued. Moreover it gives us an easy way to create a similar but different sl-hdf by changing the intermediate building blocks. This process of changing schemata is referred to as “shaking the ladder” [3]. Shaking the ladder changes the intermediate schemata which alters the reward structure that defines the fitness landscape. The three restrictions allow us to transform the full class of hdfs into a distinct but similar class that is better suited for exploring the behavior of the GA on dynamic environments.

There are many parameters that control the construction of the sl-hdfs. One parameter is the elementary schemata length (*l*) which is the distance between fixed bits in the elementary schemata. For the Cliffs and Smooth variants the elementary schemata length is not specified. When the elementary schemata length is not specified the fixed bit locations are chosen from the whole string using a uniform random process. When the length of the elementary schemata is unspecified, on average the length of these schemata will be large, approaching the length of the string. For the Weight variant, the elementary schemata length is set to $l = l_s/10 = 50$, where l_s is the overall length of the string. This relationship is based on Holland’s work [2].

To create the schemata we need to specify the weight that each schema contributes to the overall fitness function. Two parameters, mean and variance of the schemata weight, are used to specify the normal distribution from which the weight for each intermediate schemata is drawn. In all of the experiments described herein, the weight of the elementary schemata (*2*), potholes (*1*) and highest level schema (*3*) remains unchanged. In the Cliffs and Smooth variants, the weight of the intermediate schemata is also held constant at *3*. However in the Weight variant the weight of each intermediate schema is drawn from a normal distribution with mean of *3*, and variance of *1*.

In the sl-hdf there are three groups of schemata held constant: the elementary schemata, the potholes, and the highest level schema. The fourth set of schemata, the intermediate schemata, is the only group of schemata that changes. Thus the intermediate schemata can either be constructed out of any of the fixed groups of schemata, which we call the “unrestricted” construction method, or the intermediate schemata can be constructed out of just the elementary schemata, which we called the “restricted” construction method. This second method is more similar to Holland’s original description [2]. The unrestricted construction method is used in the Cliffs variant, while the restricted method is used in the other two variants previously examined (Smooth and Weight).

The schemata utilized for construction of the next level of intermediate schemata are selected randomly (“random” construction method) or from a prescribed order from the previous level (“neighbor” construction method). The random construction method creates new intermediate schemata at level *n*, by randomly choosing without replacement two schemata from level *n - 1* and combining them. In the neighbor construction method, all of the schemata at level *n - 1* are sorted by the location of their centers. The first two schemata that are next to each other in this sorted list are combined, and then the next two, until all pairs have been combined. If the random construction routine has been used then when the ladder shakes, all of the intermediate schemata are destroyed and new ones are created by randomly combining the lower level schemata to create the intermediate schemata, and weights are assigned by drawing from the distribution specified by the intermediate weight mean and variance. If the neighbor construction routine is specified, then the intermediate schemata are left alone and instead new weights are drawn and replace the weights associated with the already extant intermediate schemata. Thus when the neighbor construction routine is used the only thing that changes during the shakes of the ladder is the weights, and therefore this is called “shaking by weight.” When the random construction routine is used then the whole form of the ladder changes and thus this is called “shaking by form.”

In summary, the three variants from previous studies are:

1. The Cliffs variant which utilizes the most diverse and unrestrained set of building blocks in creating the intermediate schemata (the unrestricted, random method with unspecified string length). When creating a new intermediate schema using the unrestricted method, all of the previous level schemata, plus the potholes, and the highest level schema can be used to generate the new schemata. This has the effect of introducing ”cliffs” into the landscape. These cliffs arise because it is possible to create schemata that have a disproportionate reward (by combining with the highest level schema) and to create schemata that cover up a pothole temporarily so that when the ladder shakes dramatic changes in fitness occur.
2. The Smooth variant uses the restricted, random intermediate schemata construction method, and the intermediate schemata having unspecified string length. There are not as many sharp transitions in this landscape as the Cliffs variant.
3. The Weight variant uses the neighbor, restricted intermediate construction technique, and the elementary building blocks have a specified length. This variant most closely resembles the hdf construction routine described by Holland.

For more detailed descriptions of these variants see [7]. The main differences between these variants are detailed in Table 1. Previous work with these variants has shown that the GA performs better in dynamic environments than in static environments [3]. Also the GA performs better when transitions are abrupt (Cliffs) as opposed to those where the environment contains smooth transitions (Smooth and Weight) [7]. In the Smooth and Weight variants, in both the static case and the dynamic case, the GA becomes stuck on local optima. However, the Weight landscape since it has short elementary building blocks results in rapid progress in the early generations of a GA run before premature convergence sets in.

3. THE DEFINED CLIFFS VARIANT

Two observations of the GA’s behavior on the sl-hdfs that have been described before are (1) the sl-hdfs’ short elementary schemata lead to rapid performance increase early on relative to sl-hdfs built from longer schemata, and (2) the unrestricted construction method with shaking by form (e.g. the Cliffs variant) allows for an increase in performance because it prevents premature convergence. These two observations have lead us to construct a new sl-hdf variant, "Defined Cliffs," in which we combine short elementary building blocks with the unrestricted construction method and shaking by form method.

In the Defined Cliffs the term "Defined" comes from the fact that the elementary schemata length is set to a defined length, and the term "Cliffs" comes from the fact that all other parameters are set similar to those in the Cliffs variant. Most importantly the unrestricted construction technique and shaking by form are utilized. In the Defined Cliffs variant the intermediate schemata are constructed by an unrestricted construction method, from any of the fixed groups of schemata, rather than just the elementary schemata. When creating a new intermediate schema using the unrestricted method, all of the previous level schemata, plus the potholes, and the highest level schema can be used to generate the new schema. Because the combination of any schemata and the highest level schema is the highest level schema, many intermediate schemata are replaced by copies of the highest level schema. As a result any string which matches the highest level schema will have a much higher value relative to the other strings than it does in the other variants. In addition an elementary schema can be combined with potholes to create a new intermediate schema, which matches the pothole, but has a positive reward associated with it. However, since intermediate schemata are changed at every shake of the ladder, this positive reward is only temporary. When the ladder shakes this reward will be removed and any individual using that schemata will receive a penalty for the pothole that was "covered up" before. It is these two effects that make the Cliffs environment's changes more abrupt.

In the Defined Cliffs variant the schemata utilized for constructing the next level of intermediate schemata are selected randomly. When the ladder shakes (shaking by form) all the intermediate schemata are destroyed and new ones are created. The variance of intermediate schemata weights is set to 0 and the mean of the intermediate schemata weight is 3. The elementary schemata length is set to the defined length $l = l_s / 10 = 50$ where l_s is the over all length of the string in our case 500.

Table 1. sl-hdf variants and GA Parameters

Parameter	*Cliffs*	*Smooth*	*Weight*	*Defined Cliffs*
Population size	1000			
Mutation Rate	0.001			
Crossover Rate	0.7			
Generations	1800			
String length	500			
Selection Type	Tournament , size 3			
Number Elementary schemata	50			
Elementary Schemata Order	8			
Elementary Schemata length	undefined	undefined	50	50
Mean, Variance of Int. schemata	3, 0	3, 0	3, 1	3, 0
Intermediate Construction Method	unrestricted random	restricted random	restricted neighbor	unrestricted random
t_δ	100, 1801			
Number of Runs	30			

4. EXPERIMENT AND RESULTS

The basic setup for our experiments is a simple GA using the sl-hdf as its fitness function. The base GA presented here uses one-point crossover, per bit mutation, and full population replacement. The GA performance is studied at $t_\delta = 100$, and $t_\delta = 1801$. t_δ controls the number of generations between changes in the sl-hdf. $t_\delta = 1801$ represents a static environment, since the runs we will be presenting are only observed for 1800 generations, and $t_\delta = 100$ represents a dynamic environment. On the basis of previous results, it was found that $t_\delta = 100$ provides a good setting for understanding how the GA behaves in regularly changing environments [6]. Other values of t_δ have been investigated (t_δ = 1, 25, and 900) but the results are outside the scope of this paper. Earlier work has shown that the highest rates of fitness are attained by intermediate rates of change, t_δ = (25, 100) and these intermediate values have qualitatively similar behavior. It has been theorized previously that the high performance attained by these values is because with an intermediate shaking of the ladder the system is forced to maintain diversity and not rely too heavily on any particular set of building blocks. In cases where the environment changes slowly the GA finds local peaks and stays near them. In cases where the environment changes too quickly the GA is never able to maintain a set of intermediate schemata, and the best the GA can do is maintain a population which reflects an average of the environments it faces. However, if the time between changes is long enough, then the GA can track the changes in the environment, and the best individuals are able to attain high performance after changes. When $t_\delta = 100$ the system tracks and adapts to changes rapidly, almost constantly improving performance with only small dips for the changes [3]. Though there are some temporal differences for different intermediate valued t_δ's the general trend is similar, and thus we choose $t_\delta = 100$ as a representative.

All parameters used in the experiment are given in Table 1. The optimal value achievable by the sl-hdf is 1.0. All results below are presented at 10 generation increments in order to make the graphs easier to read. We present four different measurements to compare the behavior of the system in the Defined Cliffs variant to the previous variants: performance, threshold satisficability, loss robustness and bitwise diversity. For in-depth descriptions of these measures see [5].

In order to understand how the Defined Cliffs variant affects the behavior of the GA, we first present the results of the GA within the Defined Cliffs variant alone, and then we compare and contrast the behavior of the GA over time in the Defined Cliffs variant to the three previous variants on the four different measures.

4.1 Performance

Performance is the standard measure of how well the system performs on the task presented. To provide some description of the distribution (and variance) of the results, Figure 1 illustrates both the fitness of the best individual in the population (Best Performance) and the average fitness of the whole population (Average Performance) for $t_\delta = 100$ and $t_\delta = 1801$ averaged across 30 runs, presented every tenth generation.

These results show that the Best and the Average Performance for the constantly changing environment surpasses the static environment, even though initially the dynamic environment under-performs the static environment. In particular, after the 300th generation of the run the dynamic environment has achieved a superior fitness in both best individual and average fitness of the population. This is because the regularly changing environment prevents the GA from prematurely converging. Also, in regularly changing environments the Average Performance of the system falls farther than the Best Performance because when the ladder is shaken many individuals that were being rewarded before lose those rewards and their fitness falls greatly driving down the Average Performance of the system. Best Performance is not affected as much because there will usually be individuals who contain some of the schemata that are now being rewarded.

The behavior of the GA on the Defined Cliffs landscape is similar to its behavior on the Cliffs landscape. In both cases the GA performs very well and eventually discovers the optimal string. Also the GA is able to make rapid progress early on because the shorter elementary building blocks are easy to discover. This result is similar to that observed in the Weight variant.

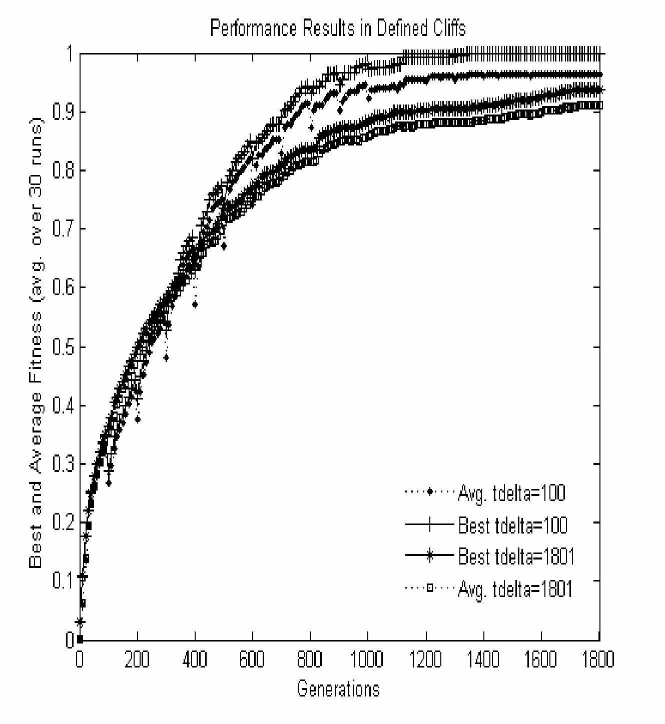

Figure 1. Performance Results in Defined Cliffs Variant

Figure 2 displays the Best fitness averaged over 30 runs of all four variants: Defined Cliffs, Cliffs, Weight, and Smooth in a regularly changing environment with $t_\delta = 100$. The GA obtains the best results throughout the runs when operating in the Defined Cliffs landscape. The GA operating in this environment rapidly increases fitness in the first generations, at about the same rate as the Weight variant. However, the GA operating in the Defined Cliffs variant continues to increase, surpassing the Weight variant, until it reaches an optimal value similar to those found for the Cliffs variant, and so achieves values much higher than those obtained within the Weight and Smooth variants. We believe the early rapid rise in fitness happens because the Defined Cliffs is discovering the short building blocks early on and thus is able to increase its fitness quickly, as it does in the Weight variant. Moreover, the rough changes in the landscape prevent the population from prematurely converging and allow the population to continue to increase in fitness, in a manner similar to its behavior for the Cliffs variant.

Standard error bars are not displayed due to the fact that they make the graph difficult to read, but results for the Weight and Defined Cliffs variants are statistically indistinguishable early on. Both of these variants are statistically distinguishable from the Cliffs variant during this time period (Generations 175-400). Later in the run, the Cliffs and Defined Cliffs variants are statistically indistinguishable, but both are statistically distinguishable from the Weight variant (Generations 1100-1800). In between these two periods the Defined Cliffs variant is statistically distinguishable and achieves superior performance to all other variants (Generations 400-1100).

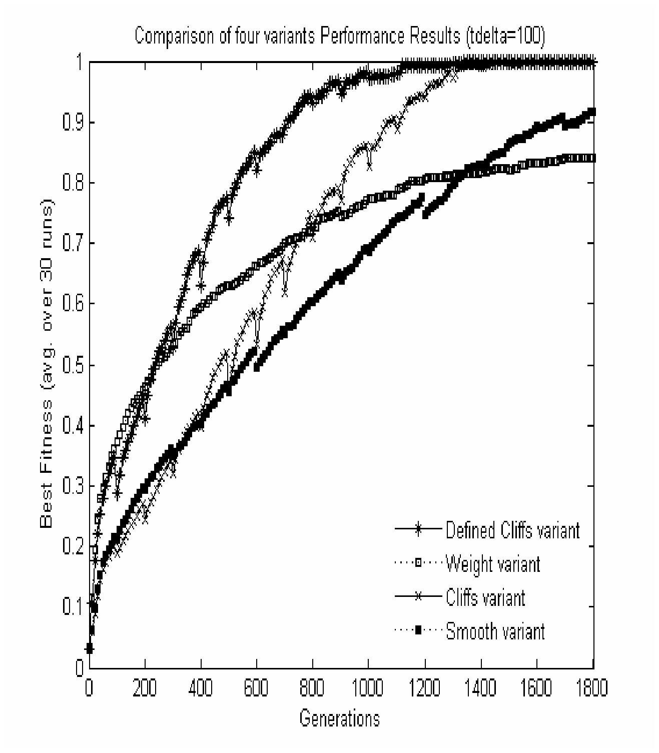

Figure 2. Performance of Defined Cliffs, Cliffs, Weight, Smooth Variants t_δ =100

4.2 Satisfiability

Satisfiability is the ability of the system to maintain a certain level of fitness and to avoid egregious errors. Figure 3 illustrates Threshold Satisfiability for t_δ = 100 in all four variants. Threshold Satisfiability is how many best of generation individuals out of 30 runs (Best Satisfiability) were able to satisfy a goal of a satisficeThreshold = 0.5 (where the satisficeThreshold specifies what fraction of the best fitness the individual is able to achieve; a satisficeThreshold of 0.5 indicates that the individual must exceed half the fitness value of an optimally valued individual).

The GA operating in the Defined Cliffs environment is able to outperform the other environments early on with regard to this measure. In fact, a large number of individuals quickly have a fitness exceeding half the optimal value. Again this is because the Defined Cliffs discovers the short building blocks early on and achieves a high level of threshold satisfiability similar to the Weight variant. There are drops after each shake but it quickly regains higher values and continues to increase. The drops in fitness due to shakes of the ladder decrease in magnitude in the last generations, and the drops in fitness are not any worse than the Cliffs variant. The Cliffs and Smooth variants start slow and achieve high results later in the run.

4.3 Robustness

The third measure we look at is robustness, which is a measure of how the system responds to change in the environment. We present the Loss Robustness of the best individual in the population (Best Robustness) for t_δ = 100 across the entire run (averaged across 30 runs). Loss Robustness is the current generation's fitness divided by the previous generation's fitness. If this value exceeds 1.0 then Loss Robustness is 1.0. Figure 4 displays these results in all four variants.

The results illustrate that in all variants every time the ladder is shaken the robustness decreases substantially but then recovers. In the early generations the decrease is large then later in the runs when the GA is not affected as much by shakes of the ladder. This is because at the end of the run the GA has found most of the intermediate schemata, but has not assembled them into one individual, and thus it is not affected as much by shakes of the ladder since there is some individual in the population that has the new intermediate schemata. In the beginning and middle are when the GA has the largest decreases in robustness, and this is because at this phase the GA has devoted lots of resources to exploring particular intermediate schemata. The Weight variant has the smallest drops and they are the same magnitude until the end of the run when they start to decrease. This is because shakes do not greatly affect the GA's behavior in the Weight variant. The Defined Cliffs variant shows the largest drops in the early generations but it is also the first variant to stop being affected by the shakes, which indicates that the GA is able to find the optimal string faster in this variant.

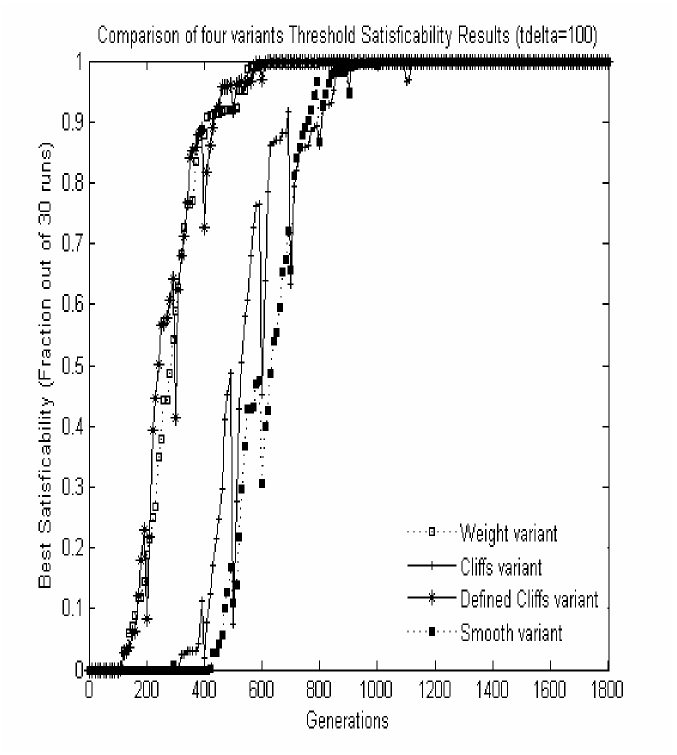

Figure 3. Satisficability Results in Defined Cliffs, Cliffs, Weight, Smooth Variants t_δ =100

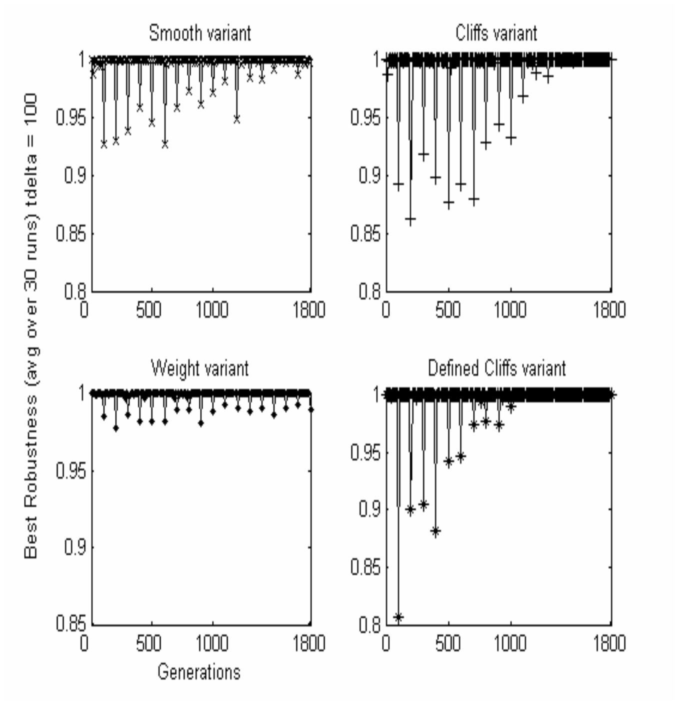

Figure 4. Robustness Results in Defined Cliffs, Cliffs, Weight, Smooth Variants t_δ =100

4.4 Diversity

Diversity is a measure of how different the members of the population are. To measure Bitwise Diversity, we use the scaled hamming distance of the population averaged across all pairwise combinations of individuals (Average Diversity). We present the results for t_δ = 100, with the values averaged over 30 runs in all four variants: Defined Cliffs, Cliffs, Weight, and Smooth, which can be seen in Figure 5.

In all variants the diversity initially drops significantly as the GA converges on those individuals that have some of the elementary schemata present in them, with the Weight and Defined Cliffs variants dropping further than the other two variants. After that, diversity increases, with the diversity of the Weight variant increasing at a faster rate than that of the other variants. As noticed in previous results [5], diversity increases when selection pressure decreases. Since there is little change in the selection pressure in the Weight variant, the diversity, after its initial increase, gradually levels off. Diversity stops increasing because the GA in the Weight variant quickly converges on a group of intermediate schemata and does not search for additional combinations similar to its behavior in performance. However, in the Cliffs, Smooth and Defined Cliffs environments there is more change in the selection pressure when the ladder is shaken and hence, the diversity decreases immediately after the change, and the increase is more gradual (around some increasing average). In the Defined Cliffs variant the drops after the shakes are only at the beginning of the generations and after the GA has found the optimal string, around generation 900. After this diversity stops being affected by these shakes and the graph becomes smoother but gradually increases reaching the same level as the Weight variant. This behavior also happens in Cliffs and Smooth variants but occurs much later in the generations, since this behavior is caused by the population converging. In the Cliffs variant it happens after generation 1200 since at that point the Cliffs variant has started to converge on the optimally valued schema.

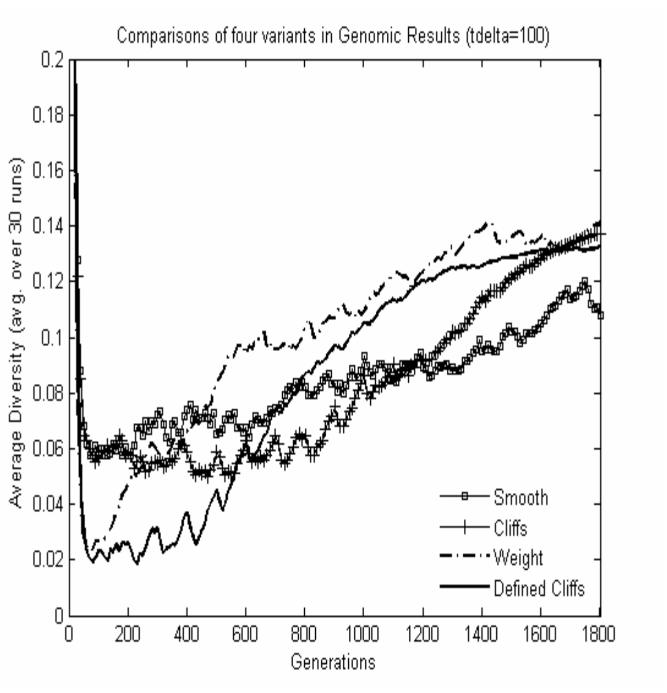

Figure 5. Diversity Results in Defined Cliffs, Cliffs, Weight, Smooth Variants t_δ =100

5. CONCLUSION AND FUTURE WORK

This paper describes a method for constructing a new variant of the sl-hdfs, the Defined Cliffs, by varying the way basic building blocks are created and combined. In particular, previous observations [7] of the GA's behavior on the sl-hdfs suggest two properties which are associated with high GA performance in dynamic environments: (1) sl-hdfs with short elementary schemata can initially perform quite well relative to sl-hdfs built from longer schemata, and (2) shaking by form along with the unrestricted construction technique allows for an increase in performance because it prevents premature convergence. Thus the Defined Cliffs variant uses short elementary schemata with sharp transitions in the environment. The results presented here show that GA behavior is superior on this new variant, compared to all previously explored variants. Moreover, the GA operating in the Defined Cliffs environment exhibits interesting behavior on the non-performance metrics that we examined. Thus these results support our claim that the differences on GA behavior over all the sl-hdf variants are a result of two main influences on GA performance: (1) short elementary building blocks, and (2) unrestricted landscapes with rough transitions. Short elementary building blocks mean that the GA can quickly increase fitness early on and achieve a high level of threshold satisficability in a small number of generations. Unrestricted landscapes with rough transitions mean that the GA does not prematurely converge and thus is able to move beyond local optima and achieve superior results. In conclusion, these experiments show that the construction of intermediate schemata, the method of changing those intermediate schemata and the length of initial building blocks can dramatically affect the behavior of the GA.

The sl-hdfs are currently a model of problems where there are basic constant building blocks, changing sub-problems, and a constant global optimum. However, it would be interesting for future work to examine other models. For instance, utilizing the sl-hdfs as a basis, we change a small number of the elementary schemata every ladder shake; this would also change the highest-level schema and thus more dramatically change the landscape. Yet the new sl-hdf would still be similar to the previous one, depending on the number of elementary schemata changed. This would allow us to create a new form of the sl-hdf that would model continuously changing environments.

By continuing to deepen our understanding of how the composition of building blocks can affect the GA's behavior practitioners can start to gain insight into how to better utilize GAs in real world problems. However, it would be useful to use the insights we have gathered so far to develop a closed-form model of how these various factors interact. In future research we hope to develop such a model that shows how changes in various parameters affect diversity and other measures. By accumulating regular observations in this environment we can observe how these interactions might occur and then hypothesize models to explain them. These experiments help to explore intuitions and understandings that have been informally observed. We plan to continue to conduct systematic controlled observations of the GA. We feel that this allows us to contribute to theory by providing a series of regular observations and to contribute to practice by providing suggestions for how to increase performance in a rich set of environments.

6. ACKNOWLEDGMENTS

We would like to thank the University of Michigan's Center for the Study of Complex Systems for the computer resources. We also acknowledge the Northwestern Institute on Complex Systems for support of WR.

7. REFERENCES

[1] Branke, J.: *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers (2001).

[2] Holland, J.H.: Building blocks, cohort genetic algorithms, and Hyperplane-defined functions. *Evolutionary Computation 8* (2000) 373–391.

[3] Rand, W., Riolo, R.: Shaky ladders, Hyperplane-defined functions and genetic algorithms: Systematic controlled observation in dynamic environments. In Rothlauf, F.et al., ed.: *Applications of Evolutionary Computing, Evoworkshops*: *EvoBIO, EvoCOMNET, EvoHot, EvoIASP, EvoMUSART, and EvoSTOC. Volume 3449 of Lecture Notes In Computer Science*., Springer (2005).

[4] Rand, W., Riolo, R.: The problem with a self-adaptive mutation rate in some environments: A case study using the shaky ladder Hyperplane-defined functions. In Beyer, H.G.et al., ed.: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, New York, ACM Press (2005).

[5] Rand, W., Riolo, R.: Measurements for understanding the behavior of the genetic algorithm in dynamic environments: A case study using the shaky ladder Hyperplane-defined functions. In Beyer, H.G.et al., ed.: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, New York, ACM Press (2005).

[6] Rand, W.: *Controlled Observations of the Genetic Algorithm in a Changing Environment: Case Studies Using the Shaky Ladder Hyperplane-Defined Functions*. PhD thesis, University of Michigan (2005).

[7] Rand, W., Riolo, R.L.: The Effect of Building Block Construction on the Behavior of the GA Shaky Ladder Hyperplane Defined Functions. In: *EvoWorkshops. (2006)* 776–787.

[8] Stanhope, S.A., Daida, J.M.: Optimal mutation and crossover rates for a genetic algorithm operating in a dynamic environment. In: *Evolutionary Programming VII.* Number 1447 in LNCS, Springer (1998) 693–702.

[9] Whitley, D., Rana, S.B., Dzubera, J., Mathias, K.E.: Evaluating evolutionary algorithms. *Artificial Intelligence 85* (1996) 245–276.