

# Behavior of Finite Population Variable Length Genetic Algorithms Under Random Selection

Hal Stringer

University of Central Florida  
College of Engineering and Computer Science  
Orlando, FL 32816-2362  
stringer@cs.ucf.edu

Annie S. Wu

University of Central Florida  
College of Engineering and Computer Science  
Orlando, FL 32816-2362  
aswu@cs.ucf.edu

## ABSTRACT

In this work we provide empirical evidence that shows how a variable-length genetic algorithm (GA) can naturally evolve shorter average size populations. This reduction in chromosome length appears to occur in finite population GAs when 1) selection is absent from the GA (random) or 2) when selection focuses on some other property not influenced by the length of individuals within a population.

## Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – *heuristic methods*

## General Terms

Algorithms, Performance, Experimentation.

## Keywords

Genetic Algorithm, Variable Length, Random Selection.

## 1. INTRODUCTION

In [10], Wu and Stringer investigated the use of a new form of genetic algorithm (GA) to develop rule sets for autonomous micro air vehicles (MAVs). This “Chunking GA” or ChGA combines a variable-length GA with a shared communal memory that allows evolution of both individual chromosomes and memory simultaneously. Memory can be accessed by individuals in the population to improve their solutions over the course of a GA run.

The ChGA exhibits a number of unique properties which warrant further investigation. One of these is the way in which a ChGA evolves its memory contents to contain an optimal or near optimal solution. The ChGA appears able to separate good genes from bad (or less good) and store them in the shared memory structure. In [8] the authors provide a more formal investigation of this “winnowing” effect and the relationship between memory capacity and solution size.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25-29, 2005, Washington, DC, USA.  
Copyright 2005 ACM 1-59593-010-8/05/0006...\$5.00.

A second unique property of the ChGA is its ability to reduce the size of individual base chromosomes (e.g., MAV rule sets) over time. This reduction occurs without explicit parsimony pressure built into either the ChGA or the fitness function.

A series of experiments was conducted prior to starting this work using a ChGA and various fitness functions to verify that reductions in base chromosome lengths were not the result of the MAV simulator or  $m \times n$  MaxSum functions used in previous works by Wu and Stringer. Although not reported here, the results of those experiments show that winnowing and reduction in base chromosome size occur to varying degrees with most of our test functions. In addition it appears that reduction in base chromosome length always occurs after the contents of memory are filled with a near optimal solution.

We have come to believe that reduction in base chromosome length is related to reduction in selection pressure as the ChGA finds a near optimal solution and stores it in memory. To put this in proper perspective, we must look at what happens during a ChGA's execution.

As mentioned previously, the ChGA evolves the contents of its shared memory simultaneously with access to that memory by individuals within the population at large. At some point in the ChGA's execution, memory contains numerous high-value genes. These genes are in turn accessed by more and more individuals within the population. Over time, the contents of memory becomes the primary contributor to these individuals' fitness. The contents of their base chromosomes becomes less relevant. As the run continues, the majority of individuals reference the same memory contents (or slots) and we begin to see similar fitness values for most of the population – the population converges on references to the same memory slots. As a result, selection among these like individuals within the population becomes essentially random in nature.

In this work we focus specifically on understanding and modeling how the absence of selection pressure (random selection) can cause reduction in chromosome sizes for a variable-length GA given 1) a finite population and 2) a uniform distribution of chromosome lengths within a population.

We begin in Section 2 with a discussion of work by other researchers related to our area of interest. We test our ideas using a variety of experiments and models in Sections 3 and 4. Empirical data from models in these sections show that absent selection pressure related to chromosome size, a variable-length finite population GA can evolve populations with shorter average-sized chromosomes. Our conclusions are contained in Section 5.

## 2. RELATED WORK

In this work we investigate changes in chromosome length under random selection (no selection pressure). No mention of this specific topic was found in our search through the GA literature. Examples may exist which relate to Grammatical Evolution or Pittsburgh Learning Classifier Systems but we leave the search in those areas for future work. The examples found to date have actually come from the Genetic Programming (GP) community. In this section we summarize those examples and discuss their relevance to GAs under similar conditions.

The first reference concerning random selection on chromosome size was found in [9]. It was observed that under random selection bloating did not occur (Figure 7.1 of that work actually shows a slight reduction in tree size). Tackett theorized that bloat was proportional to selection pressure – the greater the selection pressure caused by fitness, the faster the increase in chromosome size. It was suggested that bloat might be further strengthened by hitchhiking forces which carry non-coding regions along with useful material into next generation individuals.

Experiments performed in [1] confirm some of the Tackett’s findings. Bloat is shown to be related to selection pressure. The absence of selection pressure stops the growth of program size. Like Tackett, the authors found a “slow reduction in program size” under random selection. This reduction was attributed to biases in the crossover operator that require children to be less than a pre-specified length restriction.

A recent series of papers by Langdon, McPhee, Poli and Rowe as well as Stephens and Waelbroeck have culminated in new theory regarding variable-length structures that has direct bearing on our work. A summary follows of publications in that series which are directly relevant to our work.

In [3], a new schema theory is proposed for GP. This new theory is specific to GP using two new genetic operators: *one-point crossover* and *point mutation*. The new crossover operator works by selecting a point on both parents from a common region. The common region consists of those portions of both parents, starting from the root with the same arity of nodes and related edges. A single point is selected uniformly from edges within this region to serve as the crossover point. Subtrees below this point are then swapped between parents to produce two new children. This type of one-point crossover does not increase the depth of the children with respect to both parents but does encourage mixing of subtrees. The use of this specific form of crossover allowed Poli and Langdon to develop their new schema theorem which mathematically models “the competitions between programs with different structure and programs with the same structure.”

At about the same time, Stephens and Waelbroeck were developing a new exact schema theorem for genetic algorithms [6, 7]. The primary contribution of these works was the addition of schema reconstruction via crossover and mutation to Holland’s original, more pessimistic schema theorem. Another important finding from [7] is the following:

“We also showed that, generically, there is no preference for short, low-order schemata. In fact, if schema reconstruction dominates, the opposite is true – typically large schemata will be favored. Only in deceptive problems does it seem that short schemata will be favored, and then only in totally deceptive

problems as the system will tend to seek out existing non-deceptive channels.”

In this quote the use of “short” and “large” refer to the defining length of schemata. This finding is important as a possible clue to the cause of bloat in variable-length GAs. Without some bloat control mechanism, such GAs grow rapidly in size during early generations as they work to combine building blocks into more complete and optimal solutions. The fact that large schemata can improve building block construction may explain why variable-length GAs grows so rapidly in early generations.

The idea of exact schema was applied to Poli and Langdon’s new schema theory in [5] to produce “a macroscopic exact schema theorem for genetic programming with one-point crossover.” Included in that work was an example that calculated the total transmission probability and effective fitness for a specific schema in a limited population consisting of functions of singular or 1-arity.

The example just described served as inspiration for two subsequent works: [2, 4]. In these papers, the authors develop an exact schema theory for linear structures (similar to variable-length GAs) consisting solely of 1-arity functions and a single terminal. Results were given for linear structures under both the previously described one-point crossover and *standard crossover*. In standard crossover, crossover points are chosen on both parents independently – no common region is used. This version of crossover is equivalent to that used in most variable-length GAs including the ChGA. We will not replicate here the details of Poli and McPhee’s papers but will summarize the points that are most applicable to our current work.

For one-point crossover, assuming an infinite population and constant fitness function (e.g., uniform or random selection), the average size of individuals over time remains constant – a fixed point equal to the average size of the initial population. In addition, the distribution of lengths within the population does not change over time. The distribution remains a fixed point also equal to the distribution of length within the initial population. One-point crossover has no effect on structure length.

The results under standard crossover are different. Again assuming an infinite population and constant fitness function, the average size from one generation to the next remains constant and fixed at a point equal to the average size of the initial population. But the distribution of lengths over time changes. Poli and McPhee show that under constant fitness, shorter-than-average structures are sampled more often than larger ones. Over time large individuals become larger but fewer in number while shorter individuals shrink but become more numerous. Thus we see a change in length distribution but no change in mean length from one population to the next. They conclude their work in [4] by showing that distribution of length under standard crossover is not a fixed point but a set of fixed points over time defined by a family of discrete gamma functions.

## 3. SIMULATING A VARIABLE-LENGTH GENETIC ALGORITHM

How do we verify our hypothesis that a GA without selection pressure causes the average size of a population to shrink? Since our motivation for this work is in part a desire to explain the shrinking genomes in [10], we must first verify that this behavior

is not problem specific. To do so we perform a series of empirical tests on a simple model of a variable-length GA with selection and crossover operators.

Our model starts with an integer array of 100 elements, each element representing a single individual in a 100-member GA population. The value of each array element represents the number of genes contained within an individual's chromosome (its length). Lengths can vary in the model from 0 to  $c$  where  $c$  is some size cap which invokes a truncation operator. For most of the experiments in this section, elements of an array are initialized to 20 to match the initial length used in prior ChGA works.

Before proceeding we should address the issue of "0" length chromosomes. In [8, 10] the ChGA allowed for individuals with base chromosomes of zero length. This type of "null" chromosome was possible since individuals could still access genes stored in the ChGA's communal memory structure. Without a memory structure, crossover is still possible between two parents, one being of 0 length. This scenario could be thought of as a form of asexual reproduction where the genetic material in one parent is divided between two children. Finally, for the experiments in this section the use of 0-valued array elements is a non-issue. If length is re-defined as the number of crossover points within a chromosome, then a 0-length individual is really a 1-gene chromosome, that gene being indivisible. Alternatively our experiments could also have been run using elements of size 1 to  $c+1$  and achieved the same results.

Within our model, a loop contains code which simulates random selection, crossover, and reproduction. No fitness function exists in our initial model since we want to test the performance of a GA without any selection pressure. We have also excluded mutation from our model as this operator normally affects the contents of a chromosome rather than its length.

The loop is executed 500 times to simulate 500 generations. During each "generation", a new child array of 100 elements representing the next generation is created.

A simulated 1-point crossover populates elements in a child array. Two parent elements are chosen (selected) from the first array at random. Two random integers between 0 and each parental element's value are chosen as the crossover points. Two children are produced and copied into the child array as follows: *Child1* equals the first crossover point plus the value of *Parent2* less the second crossover point; *Child2* equals the second crossover point and the difference between *Parent1* and the first crossover point. The following example illustrates our simulated crossover:

Parent1 = 25      Xover Pt 1 = 6      Child1 = 6 + (7-5) or 8  
 Parent2 = 7      Xover Pt 2 = 5      Child2 = 5 + (25-6) or 24

The above calculation serves to model the effect of crossover on the length of chromosomes from one generation to the next. Remember our concern here is size, not fitness of the individual or its genetic composition.

### 3.1 1-Point Crossover and Random Selection

A number of model experiments of 1000 runs each were performed. At the end of each run, the average length of the final population (average of 100 array elements) was computed and classified according to size ranges in Table 1. The experiments

themselves varied by the size cap ( $c$ ) applied to larger chromosomes. In past works, we have used a size cap of 100 genes with right truncation similar to Table 1(b).

**Table 1. Number of runs with avg. size of final population within various ranges for GA simulations using random selection, 1-pt crossover, and initial gene size of 20**

Ranges for Avg Population Size at End of Run	Number of Runs Within Given Ranges			
	(a) Cap=50	(b) Cap=100	(c) Cap=250	(d) Cap=1B
<10	895	740	673	665
(10 – 20]	98	163	140	133
(20 – 30]	7	63	73	67
(30 – 40]	0	28	36	30
(40 – 50]	0	6	25	22
(50 – 60]	n/a	0	15	22
(60 – 70]	n/a	0	17	13
(70 – 80]	n/a	0	12	10
(80 – 90]	n/a	0	4	5
(90 – 100]	n/a	0	4	3
>100	n/a	0	1	30
Avg. Size All Runs	3	6	11	16

As Table 1 shows, the vast majority of populations ended with an average length below the starting average of 20 genes per individual. This occurred even when a size cap of 1 billion (in effect no cap) was implemented. A smaller size cap (in this case below 250) may act as a dampening force causing more "shorter" populations. But a cap itself does not seem to be the force driving the reduction.

Could our short populations be due to the initial array values (20)? To find out we conducted the same experiment with array elements initialized to 200 thus simulating longer genes per chromosome. Results are shown in Table 2. The distribution of population averages across ranges is similar to that in Table 1 (assuming larger ranges) and the average size for all runs varies approximately by a factor of 10.

**Table 2. Number of runs with avg. size of final population within various ranges for GA simulations using random selection, 1-pt crossover, and initial gene size of 200**

Ranges for Avg Population Size at End of Run	Number of Runs Within Given Ranges			
	(a) Cap=500	(b) Cap=1000	(c) Cap=2500	(d) Cap=1B
<100	879	725	667	661
(100 – 200]	115	170	143	138
(200 – 300]	6	68	75	67
(300 – 400]	0	31	37	32
(400 – 500]	0	6	26	22
(500 – 600]	n/a	0	13	19
(600 – 700]	n/a	0	16	11
(700 – 800]	n/a	0	13	16
(800 – 900]	n/a	0	6	2
(900 – 1000]	n/a	0	3	3
>1000	n/a	0	1	29
Avg. Size All Runs	44	77	122	167

### 3.2 Results with Fitness-Based Selection

The previous experiments indicate that, absent selection pressure, a GA can shorten the average size of a population over time.

What if selection pressure is incorporated into our model? Experiments were conducted similar to those shown in Tables 1 & 2 using tournament selection with fitness maximization in lieu of random selection. A cap of 1 billion was set for all experiments in this group to allow near unlimited growth in chromosome size

In the first experiment (Table 3 column (a)), each parent and child element in the array is assigned a “fitness” equal to its size thus favoring longer individuals. Not surprisingly, the average sizes (and fitnesses) of the final populations are enormously large. All final populations are well over 100 genes in terms of average length. This experiment shows how fitness, tied in some way to length, increases the average size of the population dramatically – another illustration of Langdon and Poli’s assertion that fitness causes bloat.

In the second experiment, fitness is equal to 100 minus the size of the individual (favoring smaller individuals). In every case, all individuals evolve to a size of 0 (no genes) hence all 1000 final populations average in the 0-10 range (see Table 3, column (b)).

**Table 3. Number of runs with avg. size of final population within various ranges for GA simulations using tournament selection, 1-pt crossover, and initial gene size of 20**

Ranges for Avg Population Size at End of Run	Number of Runs Within Given Ranges			
	(a) Longer is Fitter	(b) Shorter is Fitter	(c) Fitness = $(Fp1+Fp2)/2$	(d) Random Fitness
<10	0	1000	611	709
(10 – 20]	0	0	149	107
(20 – 30]	0	0	77	52
(30 – 40]	0	0	42	25
(40 – 50]	0	0	23	21
(50 – 60]	0	0	22	11
(60 – 70]	0	0	14	10
(70 – 80]	0	0	10	9
(80 – 90]	0	0	8	7
(90 – 100]	0	0	4	7
>100	1000	0	40	42
Avg. Size All Runs	799722254	0	21	17
Avg. Fit All Runs	799722254	100	83.49	49.34

The third experiment starts by assigning a random fitness value to each array element. During execution, the children of each mating are assigned fitness values equal to the sum of their parents’ values divided by two. The result is a fitness assignment that does not depend on the length of the parents. Our simulated GA in this scenario evolves many populations with individuals of short average size (see Table 3, column (c)).

In our last experiment, fitness values for all array elements (parents and children) are assigned randomly. Fitness is in no way tied to the size of the individual. Nor does fitness from prior generations have any bearing on selection in the next generation. Like the results in column (c) we see that populations are by far, smaller than the initial starting size. The fact that fitness is randomly assigned causes a complete absence of selection pressure associated with length and we see that our GA simulation naturally evolves shorter populations. Note that this column is very similar to Table 1, column (c) which used purely random selection without any fitness function.

### 3.3 Other Crossover Types

The three prior sets of experiments (Tables 1, 2 and 3) modeled a GA using 1-point crossover. Two more sets of experiments were conducted to see if reduction in average size across populations occurs using 2-point crossover. The results of these experiments are contained in Tables 4 and 5.

These tables show that reduction in average chromosome size for a population can occur with at least one other form of crossover. A comparison of results from Table 1 & 4 to those in Tables 3 & 5 shows however that the size reduction under 2-point does not appear to be as great (or fast) as that occurring with 1-point.

**Table 4. Number of runs with avg. size of final population within various ranges for GA simulations using random selection, 2-pt crossover, and initial gene size of 20**

Avg Population Size At end of Run	Number of Runs Within Given Ranges			
	(a) Cap = 50	(b) Cap = 100	(c) Cap = 250	(d) Cap = 1B
<10	761	598	563	561
(10 – 20]	212	228	189	188
(20 – 30]	27	100	89	80
(30 – 40]	0	51	44	45
(40 – 50]	0	21	27	26
(50 – 60]	n/a	2	23	17
(60 – 70]	n/a	0	20	14
(70 – 80]	n/a	0	15	14
(80 – 90]	n/a	0	11	6
(90 – 100]	n/a	0	9	13
>100	n/a	0	10	36
Avg. Size All Runs	5	10	16	19

**Table 5. Number of runs with avg. size of final population within various ranges for GA simulations using tournament selection, 2-pt crossover, and initial gene size of 20**

Ranges for Avg Population Size at End of Run	Number of Runs Within Given Ranges			
	(a) Longer is Fitter	(b) Short = Fitter	(c) Fitness = $(Fp1+Fp2)/2$	(d) Random Fitness
<10	0	1000	555	605
(10 – 20]	0	0	179	153
(20 – 30]	0	0	77	61
(30 – 40]	0	0	53	46
(40 – 50]	0	0	38	35
(50 – 60]	0	0	22	20
(60 – 70]	0	0	20	15
(70 – 80]	0	0	10	13
(80 – 90]	0	0	8	9
(90 – 100]	0	0	8	3
>100	1000	0	30	40
Avg. Size All Runs	838,601,988	0	20	18
Avg. Fit. All Runs	838,601,988	100	83.66	49.60

### 3.4 Population Size and Run Length

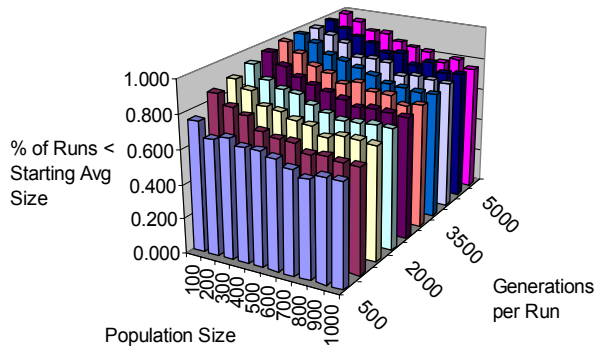
Results found in [2, 4] show that, given an infinite population and a flat fitness landscape, there should be no change in the average length of individuals within a population over time. The model we

have constructed in this section uses a finite population of 100 elements. Do our results change if larger populations are used?

To answer that question we re-ran experiments using our model but varied both the size of the population (from 100 to 1000 elements) and the number of generations (500-5000). For each size/generation combination, a total of 500 runs were made. In each run, all members of the population began with a length of 50. The number of runs ending with a population of average length less than 50 (the average starting length) were counted, converted to a percentile, then plotted in Figure 1.

Figure 1 shows that as population size increases, the reduction in mean size within a population slows. In other words, the phenomenon we have seen with our ChGA experiments and those conducted earlier in this section becomes less apparent with larger populations.

Also shown in Figure 1 is the affect of time on this reduction. The longer the GA runs, the greater the reduction in mean length. For example, when using an array of 100 elements and 5000 generations, over 99% of all runs ended with a mean length less than 50 (the mean length for the starting population). At the opposite end of the spectrum (1000 elements, 500 generations) only 60% of runs ended in populations with mean lengths of less than 50.



**Figure 1. Graph showing effect of population size and run length on percentage of runs with ending average population sizes below the starting size.**

Earlier in this work we mentioned results found in [1, 9]. Those authors found no increase in mean program length under random selection with Langdon and Poli noticing a slight reduction in size. A review of their experiment parameters show that Tackett used a population of 500 individuals and 24 generations per run. Langdon and Poli’s experiments also used 500 individuals but 50 generations per run. It may be that given more time or smaller populations, those researchers would have seen a more pronounced reduction in program size. Of course, the use of parse trees in their experiments rather than the GA’s more linear structure may have prevented size reduction regardless of the number of runs or larger population sizes tried.

Finally, the results in Figure 1 may be seen as a limited confirmation of [4]. That work concludes that “shorter structures are sampled exponentially more frequently than larger ones.” This is despite the fact that the average length within an infinite population is a fixed point over time. It is not surprising that the mean length in our experiments decreases over time given 1) that

GAs are stochastic algorithms and subject to sampling errors in finite populations and 2) that over-sampling of smaller individuals is inherent in linear structures using 1-point crossover under random selection. Sampling errors occurring in a GA with small populations are more likely to cause selection of smaller individuals as parents thereby reducing the mean size for the entire population over time.

### 3.5 Discussion

The simulations performed in this section yield several important bits of information. First, absent selection pressure our GA model prefers to evolve finite populations of shorter average size given sufficient time (generations). Without a size cap, roughly 75%-80% of all final populations have an average chromosome size less than the starting length regardless of initial starting size. Second, a size cap does exert downward pressure on chromosome size but does not appear to be the primary cause of reductions in average size. Last but not least, time is a factor in this reduction – the more individuals in a population, the more time required for reduction to take place.

Results clearly show that a finite population GA without any selection pressure tends to evolve populations with shorter average chromosome sizes. But as shown in Table 3 & 5, absence of selection is not the complete story. The GA evolves larger-size populations only when longer individuals are considered more fit. Shorter populations can evolve in the presence of selection as long as 1) fitness and size are not related to one another or 2) fitness favors more compact solutions.

These simulations seem to confirm at least that fitness can be the cause of bloat if longer individuals in a population are considered more fit. The reason for longer individuals being more fit is itself a subject for future investigation.

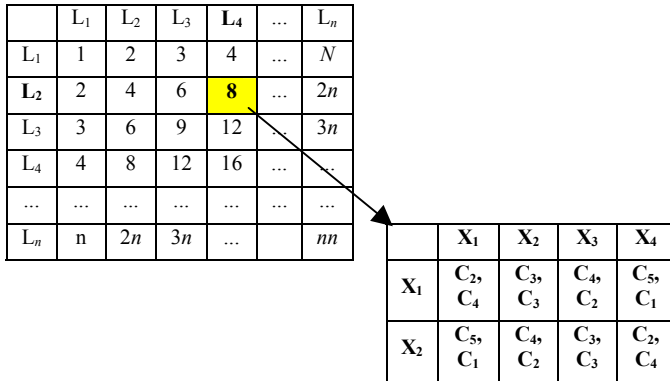
## 4. SHORTER THAN AVERAGE CHILDREN

Given that a variable length GA can evolve shorter mean length populations over time, what might be the cause of this reduction? Our next set of experiments focuses on a possible cause – the percentage of children produced by a GA which are shorter than the average size of the parent generation. If this percentage is above 50% we may be able to quantify the over sampling of smaller structures described in [2] and find a force that aids in reduction of mean genome size from one generation to the next.

To set up our experiments we define a theoretical population as one occurring at some generation  $t$  consisting of  $2n$  individuals. The individuals which make up this population consist of two of every size from 1 to  $n$  crossover points. We use the notation  $L_i$  to indicate the size of an individual where  $i$  is the number of crossover units (bits or genes) within the genome. The population in total makes up an  $n \times n$  matrix with each row and column corresponding to one of the population members (see Figure 2).

The number of possible *crossover events* for any given *mating* is the multiplication of the number of crossover points in each of the parents. For example, a total of 8 unique crossover events is possible between a single mating of two parents, one with four crossover points and the other with two. We can fill each cell in our population matrix with the number of possible crossover events that can take place for any two chromosomes of size  $j$  and  $k$  where  $j, k \leq n$  by simply multiplying the row and column

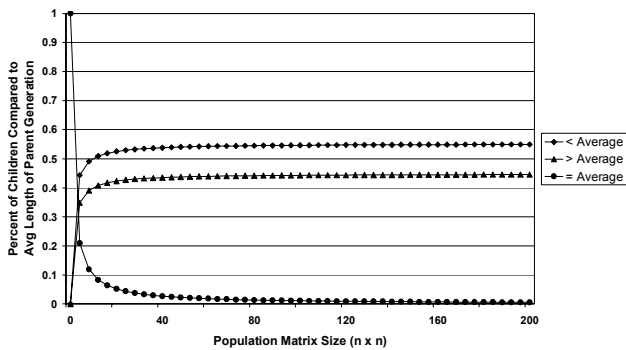
indexes. Each element of the matrix in Figure 2 shows the number of possible crossover events between parents with crossover point lengths of  $j$  and  $k$  up to  $n$ .



**Figure 2.  $n \times n$  population matrix cell showing length of children created by each possible crossover event. Number of rows and columns in the inner matrix determined by the number of crossover points in the parent genomes.**

Let us now extend our illustration in Figure 2 by expanding a single cell to contain an inner matrix representing each of the unique crossover events possible for a given mating. We use the notation  $X_i$  to indicate the different crossover points for both individuals where  $i$  is the location of the crossover point along the genome. Children produced by each possible crossover event are shown as  $C_l$  where  $l$  is the number of crossover points (length) in each child. For the mating of any  $L_2$  and  $L_4$  chromosome there are eight possible mating events which produce children of various sizes.

Our first experiment counts the number of children whose length is less than, equal to, or greater than the average size of the parent population. From a run with a 200x200 population matrix, the percentages are 34.25%, .5% and 65.25% respectively. The high percentage of greater-than children would seem to indicate that a GA actually produces children longer than the prior generation's average size. But these numbers do not include any weighting of the counts by mating occurrences. If a form of weighting is used which factors in the probability of mating between parents then the percentage of possible children of size less than the parent generation approaches 55% (see Figure 3).

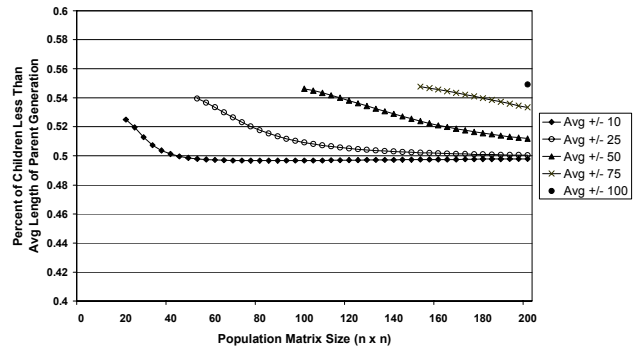


**Figure 3. Percentages of children with size less than, equal to, or greater than the average size of their parent generation for  $n \times n$  population matrices varying in size from  $n=1$  to 200.**

The experimental method used to produce Figure 3 actually tests  $n$  different matrices as  $n$  goes from 0 to 200. Both the average size of the population and the variation in array elements (difference between smallest and largest) in each matrix is increased uniformly during program execution. Two additional sets of experiments were performed to isolate which of these two factors might be the cause of the increase in shorter children.

In the first set of additional experiments, the size of the variance was fixed for each run (e.g., 10, 25, 50, 75, 100). The starting point for the program was determined by multiplying the variance by 2. For example, a variance of 25 requires a 50x200 population matrix. As the program executes it essentially moves a window 50 rows high from a starting point of 1 to an end point of 150. As a result the average size of the population matrix moves from 25 to 175. The number of mating occurrences stays constant but the number of crossover events increases as the average size within the window grows.

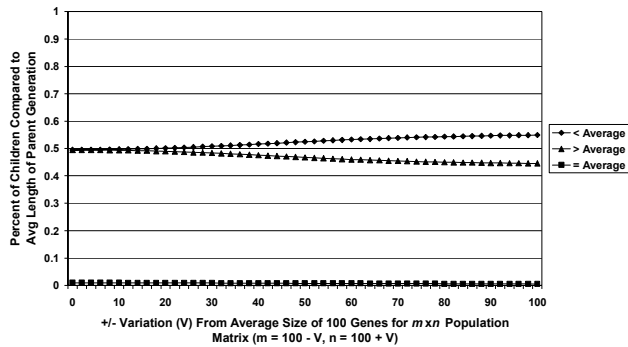
The results of this experiment are found in Figure 4 and are two-fold. First, the size of the variance definitely makes a difference. Of the five tests, a variation of 100 between smallest and largest individual in the window has the highest percentage of children whose size is below the average size of the parent population (see single point on the right side of the graph). Second, the number of less-than-average children decreases as the average size of the population goes up. This seems to indicate that the ratio of average chromosome size to diversity of size within the population matters.



**Figure 4. Effect of changes in variance on percentages of children whose size is less than the mean length of the parent population matrix.  $n$  is equal to the average times 2.**

The next experiment (Figure 5) makes this more clear. In this test, the average size of the matrix was kept at 100. The variance was increased over time from 0 to 200. This method can be thought of as a  $m \times n$  matrix expanding out from a fixed center point acting as the average.  $m$  and  $n$  change over time with  $m$  always equal to the average less the variance and  $n$  always equal to the average plus the variance.

Figure 5 clearly shows that as the size variance grows so does the probability of creating children that are of smaller size than their parents. Note however that the growth is not infinite. At the end of the experiment (200x200), the percentage of lesser children reaches 54.9% – close to the 55% mentioned earlier.



**Figure 5. Effect of changes in variance (given a fixed average) on percentages of children whose size is less than the average size of the parent population for a  $m \times n$ .**

Based on empirical evidence it appears that the probability of choosing shorter individuals for mating in the next generation is no longer a purely random event absent selection pressure. Our experiments seem to lend support to the finding in [2] that smaller and smaller structures are sampled over time for linear structures under random selection.

Empirical results show that this probability approaches 55% given a sufficiently large variation in genome size, assuming uniform distribution within the population. The fact that this percentage is only a little higher than 50% means that it can probably be overcome by stochastic effects. This small difference also gives an indication of why finite populations shrink in the absence of selection pressure but only over an extended period of time. If the probability was significantly higher, the rate of decline in size would be very rapid.

## 5. CONCLUSIONS

In this work we have conducted a series of experiments which provide strong evidence that variable-length GAs can naturally evolve populations of shorter mean length over time if selection is random or not associated with length in some way.

Empirical evidence is given which lends support to the conclusions drawn in [4] with regard to standard crossover – specifically that a variable-length GA, given an infinite population and standard crossover, over samples shorter individuals in each generation. Our experiments show that this same idea applies to GAs of finite population sizes.

## 6. REFERENCES

- [1] Langdon, W.B. and Poli, R., Fitness causes bloat. In *Proceedings of the Second On-line World Conference on Soft Computing in Engineering Design and Manufacturing*, (London, 1997), Springer-Verlag, 13-22.
- [2] McPhee, N.F. and Poli, R., A schema theory analysis of the evolution of size in genetic programming with linear representations. In *Genetic Programming, Proceedings of EuroGP'2001*, 2001, Springer-Verlag, 108-125.
- [3] Poli, R. and Langdon, W.B., A new schema theory for genetic programming with one-point crossover and point mutation. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, 1997, Morgan Kaufmann, 278-285.
- [4] Poli, R. and McPhee, N.F., Exact schema theorems for GP with one-point and standard crossover operating on linear structures and their application to the study of the evolution of size. In *Genetic Programming, Proceedings of EuroGP'2001*, 2001, Springer-Verlag, 126-142.
- [5] Poli, R., Exact schema theorem and effective fitness for GP with one-point crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000)*, 2000, Morgan Kaufmann, 469-476.
- [6] Stephens, C.R. and Waelbroeck, H., Effective degrees of freedom in genetic algorithms and the block hypothesis. In *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, 1997, Morgan Kaufmann, 34-40.
- [7] Stephens, C.R. and Waelbroeck, H., Schemata evolution and building blocks. *Evolutionary Computation*, 7 (2), 109-124.
- [8] Stringer, H. and Wu, A.S., Winning wheat from chaff: The Chunking GA. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)*, 2004, Springer, 198-209.
- [9] Tackett, W.A., *Recombination, Selection, and the Genetic Construction of Computer Programs*, Ph.D Thesis, Department of Electrical Engineering Systems, University of Southern California, 1994.
- [10] Wu, A.S. and Stringer, H., Learning using chunking in evolutionary algorithms. In *Proceedings of the 11th Conference on Computer-Generated Forces and Behavior Representations*, 2002, 243-254.