Jarrett C. Bonard
G. A. Project
Dr. John J. Aleshunas

Executive Summary

In this report is a comparison between multiple programs using genetic algorithms. Genetic algorithms optimize fitness equations to generate new populations. Populations are then continued to form until the required stopping criterion is met. Genetic algorithms are usually referred to in a Darwinian sense of evolution, by their ways of "evolving" to an answer.

Two gene reproducing algorithms and the computer programs involved have been designed here by Dr. John J. Aleshunas. Both of these algorithms are asexual in that they are able to reproduce offspring without two parent genes. Also, these algorithms call for a selected population, 10, to be evaluated on one of three fitness equations. The first algorithm looks at its original population placed in order according to their fitness, and then sections it off into thirds. Next the algorithm takes only those genomes of the fittest sector and begins to mutate them into the missing genes of the new population. The second algorithm takes the same population, and the original list of the population in order according to their fitness, and begins to mutate the survivor genes with randomly mutated genes. These randomly mutated genomes originate from any of the beginning three sections of the fitted population.

The programs continue to form new populations until the difference between the average fitness of a population and its predecessor's is equal to or smaller than the specified epsilon (*E)*:

$$O_n - O_{n-1} = E\ ,$$

where O is the average fitness for a population, and n is the number of generations.

When it had finished, a maximum value was found optimized for the corresponding fitness equation. A test could now be made between the actual Y value, and the constant maximum Y value. Percentage Errors were found by subtracting the Y_{max} from the Y_{actual} and then dividing by the $Y_{max.}$ What is found here is a not so simple answer to obvious question, "If I wanted the best, why wouldn't I only keep the best?" Actually, one algorithm presents itself quite well in that holds up against any of the others, and it randomly mutated two genes. The algorithm had the lowest Root Mean Square Error, and used the fitness equation of degree four to test the individuals

$$Y = X^4 + 7X^3 + 6X^2 + 72X$$

Other questions arose in two of the algorithms, however, and the two both used the same fitness equation. The third fitness equation proved that it was not able to find a maximum value for Y_{actual} because of the small section of people who had a positive fitness. The random selection process and our fitness equation enabled our results from being very high because the equation optimized the most negative number in order for the ratio to converge.

$$\begin{aligned} Y &= X^2 \text{ for } 1 \le X \le 3 \\ &= -X^2 \text{ for } 0 < X < 1 \text{ or } 3 < X \end{aligned}$$

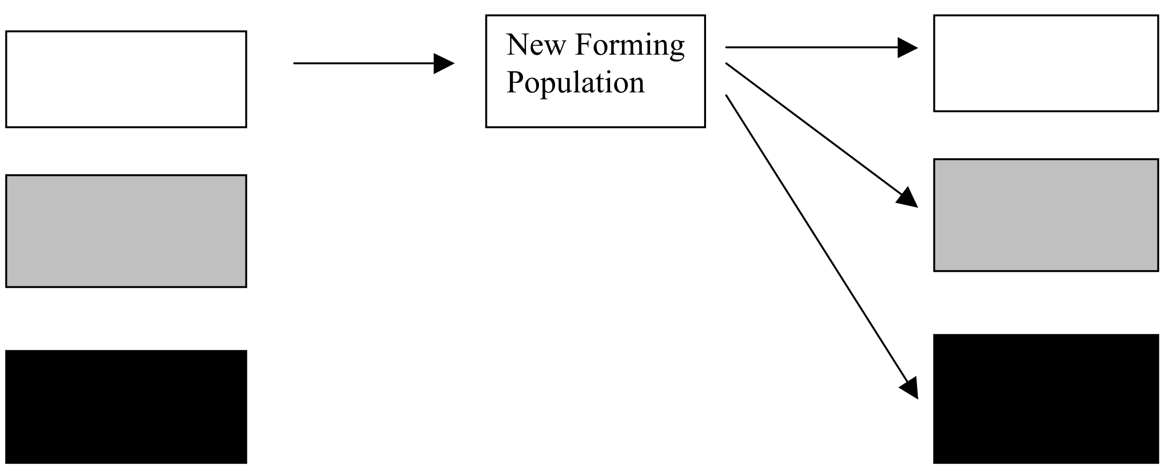
Problem Description

The purpose of the problem was to determine an asexual algorithm which could quickly generate the population of most fit individuals. The problem in this study was to generate enough populations to be able to see any proof needed to support a theory. One by one, fifty instances were generated for each program. The data was recorded and analyzed as described, and a solution theory was made according to the results.

Analysis Technique

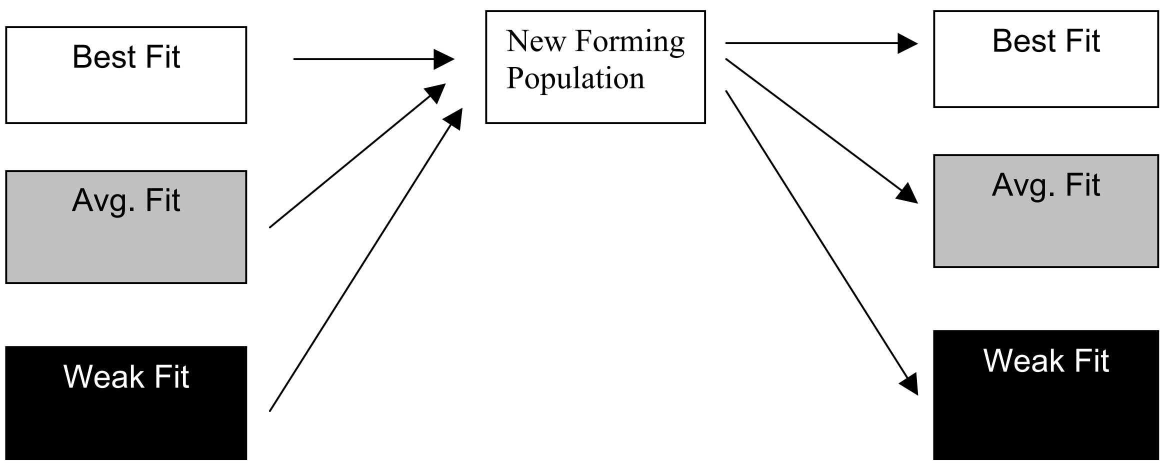
Genetic Algorithms use a problem solving process of evolution. Through natural selections, genetic algorithms, much like Darwin's theories of evolution, keep evolving until a certain criterion is met. The intent of this study is to look at similarities and differences between two main gene reproducing algorithms:

1. The first manner in which genomes are reproduced is to take a population of people, then test the individuals through a fitness equation (described later). The equation ranks the individuals in order of there fitness performances, and the resulting list is then cut into thirds. Now the algorithm will take only the strongest survivors (the upper third) and mutate new individuals from only the genes of the strongest survivors.



KEY: White-best fit third; Gray-average third; Black-below average third

2. The second technique used to generate populations was to take the population, and again test it through the fitness equations. Now the individuals were ranked in order of highest fitness scores. The genes were again brought over and a new population would be formed, mutating the missing genomes with random ones of any fitness level.



With the control of the program, alterations are made to different aspects of the algorithm in order to better see how the algorithms actually work. One factor is the equation at which the individuals will be tested. Each example uses a fitness equation that will determine the best fit individuals. The three equations for optimizing are as follows:

1. $Y = X^2$
2. $Y = X^4 + 7X^3 + 6X^2 + 72X$

$Y = X^2$ for 1 £ X £ 3

3. $= -X^2$ for $0 < X < 1$ or $3 < X$

The first of these equations is merely a quadratic equation. The second being a fourth order equation, and third is discontinuous one. The fitness of each individual is then found by taking the maximum value for Y, and dividing that value by the Y average of the population. This will generate the order of the individuals with respect to their fitness scores.

The algorithms cease to generate new populations when the difference in the average population fitness does converge to a specified *E*, or epsilon. The difference however may not be exact to the specified *E*, but actually defined as being less than *E*.

Some of the alterations made, besides that of the fitness equation, were to modify the population itself. This meant to change the population size, and see how quickly the algorithms could converge. The number of genes in each individual was also a variable which affected the results, and after finding an appropriate size, it was set to 10. The value for epsilon could be changed, which would result in a faster or slower converging rate. However, this would also entail that the results may not be very accurate as the algorithm had not been able to compute for a long enough period of time. Consequently, to ensure that our program would indeed come to an end, a limit to the maximum number of generations is set. This number was usually based at one million iterations, plenty long enough to produce a solution. Test trials were made in order to find a set of variations that pertained to all three of the algorithms so that each could come to a reasonable ending point, and a solution was given.

Assumptions

There were assigned values for the variables. These designated numbers were chosen because of the differences in the algorithms and programs. The values needed to be the same for all so comparisons could be made, but also would have to be large enough so that each program could run acceptably. These were the values chosen:

The number of elements in an individual's gene vector:	10
The number of individuals in the initial population:	10
The maximum number of generational iterations:	1,000,000
The value which measures whether the process has converged to an answer (Epsilon):	.000000001

A smaller population meant that the time generations would converge would not take as long because of the smaller search space. There were no obvious affects the population size had on the ability of the algorithm to converge other than the program took longer. Different values would indeed affect the time for the program to run, but the values chosen here were made so that each program would perform adequately and best algorithm could be found.

Results

The results of the study were insightful. For each algorithm tested, there were fifty different instances carried out. The performance of the second algorithm was higher than that of the first algorithm when the second fitness equation was used. This was the program that had the lowest Root Mean Square Error, which is the square root of the squared sums of the percentage errors, taken with respect to the total number of instances. This program also had the second lowest number of errors, having only one more than the algorithm using only the best survivor genes. The Root Mean Square Error value was computed for each program and then compared, along with an average of the number of errors which occurred. All of the algorithms were able to generate a convergence with a pretty close accuracy.

Reproduction Technique	Root Mean Square Error	% of Errors
Algorithm: 1 Fit. Equation: 1	0.13 %	8 %
Algorithm: 1 Fit. Equation: 2	0.12 %	16 %
Algorithm: 2 Fit. Equation: 1	0.57 %	18 %
Algorithm: 2 Fit. Equation: 2	0.06 %	10 %

Issues

There was an exception with two of the programs. The third fitness equation never found a maximum value for Y. Actually, it never even found a positive value. This meant that the search space was being limited by the fitness equation, and due to the very high negative number, the program was converging to a minimum instead of a maximum. Consequently, all of the data for the two algorithms using the third fitness equation had errors of 99.9%. Now even though there were never any values which were positive, it does not necessarily mean that they never could be.