Genetic Resilience and Variable Mutation Rates

Gerardo Gonzalez GERARDO.GONZALEZ-1@OU.EDU
University of Oklahoma

Abstract

It has been shown that evolutionary computation methods are influenced not only by the fitness function explicitly defined by the user but also by the genetic resilience inherent in the evolutionary mechanisms. Given an environment with specialized ecological niches that allow for high fitness and niches that are less specialized but have a lower maximum fitness, Jones and Soule (2006) investigate the effect on the resilience of individuals of a crossover mechanism that allows chromosomes to increase or decrease in size. They conclude that after the genes gain size resulting in resilience, the algorithm performs as if it is influenced only by the fitness function. In the present work, we investigate the effect of variant mutations on resilience in evolutionary computation, specifically in genetic algorithms. The core of the experiments study the effects of a mapping function from fitness to mutation rate on the shape of the existing fitness landscape. We determine whether, by manipulating this fitness-to-mutation-rate function, individuals will effectively become more resilient and the population will converge based on fitness alone, given an environment such as the one used by Jones and Soule.

1. Introduction

Evolutionary computation is a part of the machine learning field that uses algorithms inspired by natural evolution for optimization. In the evolutionary computation literature many results show that the evolutionary mechanism is influenced not only by fitness but by other forces inherent in the algorithm itself (Soule, 2005). Resilience is one of these forces that drive the evolutionary mechanism.

Genetic Resilience is the ability of a chromosome to withstand changes in its fitness when operators like mutation or crossover are applied. The problem is that in special cases, the need for genetic resilience can override the pressure of the fitness function causing convergence at undesired places in the search space.

Soule and Jones (2006), in their study of resilience in genetic algorithms, show that a way for a chromosome to gain resilience is for it to grow, and that after resilience has been gained, the algorithm will be driven by the fitness function solving the problem.

In this paper we will study the effects of resilience in genetic algorithms as shown by Soule (2005), and also the effects of variable mutation rates on resilience with the focus of finding whether mutation can be used to gain resilience.

The variable mutation rate study is interesting because in the standard use of genetic algorithms, fixed length chromosomes are used, eliminating the possibility of using chromosome growth to gain resilience as shown by Soule (2005).

The hypothesis is that by learning a function between mutation and some property of the genetic algorithm, the resilience pressure will be decreased or increase as necessary, and the algorithm will be able to converge according to the fitness function.

Even though no experimentation of the effect of resilience in real world problems has been study so far, the study of variable mutation rates are still an interesting problem that shines light on the effects of mutation in the evolutionary mechanism.

In this paper, the results from the successful reproduction of the experiments from Jones and Soule (2006) will be shown, plus the results of the initial experiments with variable mutation. Also some hypothesis on the cause of the negative results from the variable mutation rates experiments will be discussed, and ideas on why the initial hypothesis failed under the current system.

2. Problem Definition and Algorithm

2.1 Task Definition

The task is to study genetic resilience in genetic algorithms and how it can be influenced through crossover and mutation.

2.1.1 Effects of Crossover

To study the effects of crossover in resilience a controlled fitness landscape is chosen . The landscape used for this paper is represented by the function shown in Figure 1 which is also one of the landscapes studied by Soule (2005). This fitness function is simply a mapping between the sum of the chromosome, and a fitness value given by the landscape. For example a chromosome 0 1 1 4 4 4 1 would be summed to 15 (15 is also called the value of the chromosome), and by looking at the fitness function the chromosome receives a fitness of 0.

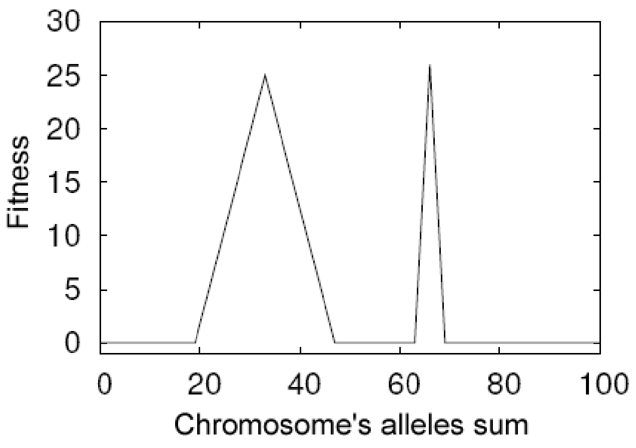


Figure 1

This landscape consist of two peaks. The first peak is wider with a maximum fitness of 25, while the second peak is narrower with a maximum fitness of 26. Even though the difference in fitness between the two peaks is only one point, the population should converge to the narrower peak given the fitness function because the narrower peak has the greatest fitness as its peak.

In Soule's experiments, the results show convergence to the wider peak. This is caused by the lack of resilience in the population.

Crossover, which is the only operator that is used in the literature experiments, is very destructive causing drastic shifts in fitness between parents and offspring. Since the first peak is wider, even if the offspring shifts, there is still a great chance that the offspring will be fit. The members of the population at the narrower peak will be drastically affected by any shift making the offspring very unfit.

By doing this mental experiment, it can be seen that even though the narrower peak is more fit, no chromosome is able to stay at this peak because it is to narrow. The only way to stay in the narrower peak is for the chromosomes to be resilient. In Soule's experiments, the solution to the problem comes from chromosome growth and resilience gain by this growth. For example a long chromosome with many 0's will give away mainly 0's during crossover maintaining its fitness, this means that its offspring will have a greater chance to be as fit as itself or better.

Even though a solution for this problem has already been study by Soule, the reproduction of two of his experiments, one with chromosome growth and one without, will set a framework for further study of the fitness landscape. Because once the algorithm used in this paper presents similar behavior to the genetic algorithm in the literature, a point of comparison will be established for the further studies presented in this paper.

2.1.2 Effects of Mutation

The second task is to study the effect of mutation on resilience. For this, the same framework from the literature reproduction will be used, and new results will be acquired only by changing the operators used during the evolution of the population, in this case the only operator will be mutation.

The hypothesis for further study is that there exists a function that will map some seen attribute of the genetic algorithm to a mutation rate, that will cause the desirable parts of the fitness landscape to become more stable, and the undesirable parts more unstable. For example in the landscape of Figure 1, the desirable function will make x values that are further away from the narrow peak more unstable by assigning a high mutation rate, and as the x values come closer to the narrow peak, the function will assign smaller mutation rates, so that these areas will be more stable and the chromosomes will be able to stay at those more desirable fitness values with greater probability.

In theory it sounds simple, but in a real life genetic algorithm the landscape is not seen, and other attributes will have to be used to find the mapping function.

In this paper two attributes will be studied:

– chromosome fitness
– Difference of fitness between a parent and its offspring

2.2 Algorithm Definition

2.2.1 Basic Framework

The basic framework of this experiment is a common genetic algorithm with the parameters used by Jones and Soule (2005,2006). These parameters are summarized in Table 1.

The row labeled integer values refer to the possible values a gene can have. It is also important to note that a member of the population can be selected more than once to be part of the tournament, and there is not replacement. This means that the new offspring will not participate on the selection of the generation in which it was born.

2.2.2 Crossover

Two types of crossover are used, constant and proportional, as defined by Jones and Soule (2006).

During the experiments that use crossover, a 0.9 crossover rate was used.

2.2.3 Mutation Functions or Maps

Two mutation functions are used, each related to one of the attributes defined in section 2.1.2.

Fitness	F(sum of integers in chromosome)
Integer Values	0,1,4
Population Size	500
Selection	3 member tournament (1 for each parent)
Iterations	2000 generations
Maximum Length	3000 genes
Initial Population	Random, of lengths 5 to 59.
Number of Trials	50
Crossover	Proportional and Constant (explained below)
Peak 1 Height	25
Peak 1 Location	33
Peak 1 Width	14
Peak 2 Height	26
Peak 2 Location	66
Peak 2 Width	3

2.2.3.1 First Function

For the first function an Array of size 100 is used. This is the map that will hold the mutation rate of each value in the x coordinate in the function in Figure 1.

All of the values in the map are initialized randomly between 0 and 100.

The algorithm also keeps track of the maximum fitness seen through the history of the genetic algorithm.

The update rule is captured in this equation:

*Map[chromosome_value] = ((max_seen_fitness – chromosome_fitness) / max_seen_fitness) * 100*

2.2.3.2 Second Function

The second function uses a two dimensional Array as the map. The first dimension is of size 100, and the second dimension of size 2. This map will be used to keep track of two values for each position in the x coordinate of the function in Figure 1. The first value is the mutation rate, and the second value is the difference between the parent and the offspring.

The mutation rate is initialized randomly between 0 and 100. The difference can be initialize to any significantly high value, in this case 200 was chosen as the initial value.

The algorithm also keeps track of the maximum seen fitness through the history of the genetic algorithm.

The pseudo code for the update rule is described below:

```
Mutation_rate = Map[chromosome_value][0]
Mutate chromosome according to Mutation_rate to obtain offspring
Calculate fitness for both parent and offspring
Calculate the fitness_difference between the offspring and the parent
If fitness_difference > Map[chromosome_value][1] (or past seen difference)
    Map[chromosome_value][0] -= ((fitness_difference – Map[chromosome_value][1] ) / max_seen_fitness)
Else
    Map[chromosome_value][0] += ((Map[chromosome_value][1]- fitness_difference ) / max_seen_fitness)
Map[chromosome_value][1] = fitness_difference
```

3. Experimental Evaluation

3.1 Methodology

The criteria for evaluation, or the measure of success is the convergence of the population to the narrow peak in the landscape in Figure 1 which has its peak value at 66 in the x coordinate.

The convergence to this value will test whether the proposed method in an experiment is causing the population to become more resilient or not.

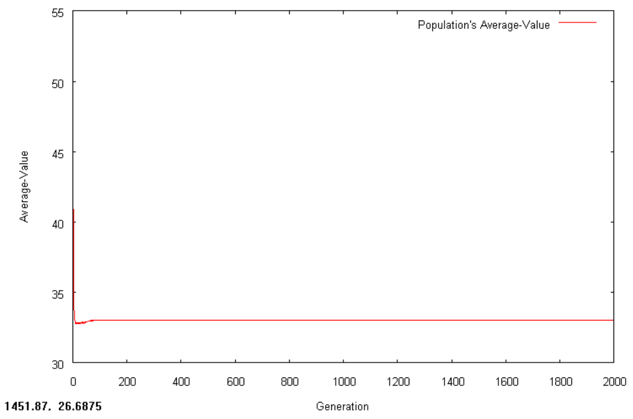
For each experiment, all of the populations were saved through all the 2000 generations. Then the average values of the populations were calculated from the population data. This average population value will show were the population is converging to through the experiment.

Also for the first two experiments, the average number of 0's, 1's and 4's was calculated. This data shows that an increase of 0's and 1's in the population as the chromosomes grow is what is causing the gain in resilience and the convergence.

3.2 Results

The first set of results shown in Figure 2 are those of the experiment with proportional crossover.

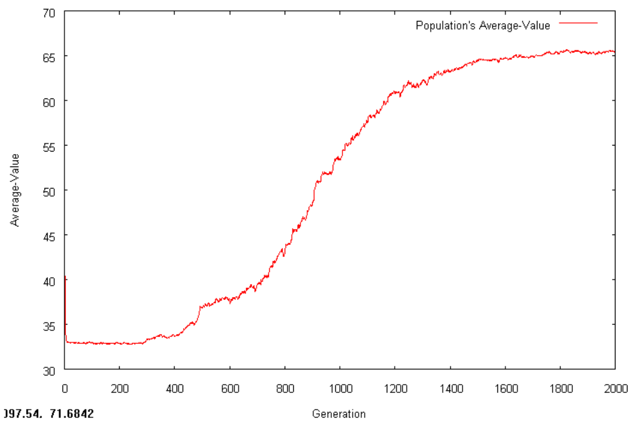
In this Figure, the population's average shows convergence exactly as expected according to the literature results from Soule (2005). The population converges really quickly to the narrower peak at value 33 because there is no growth, and no resilience being gained by the population.



Average of the allele's sum over 5 trials with proportional crossover.

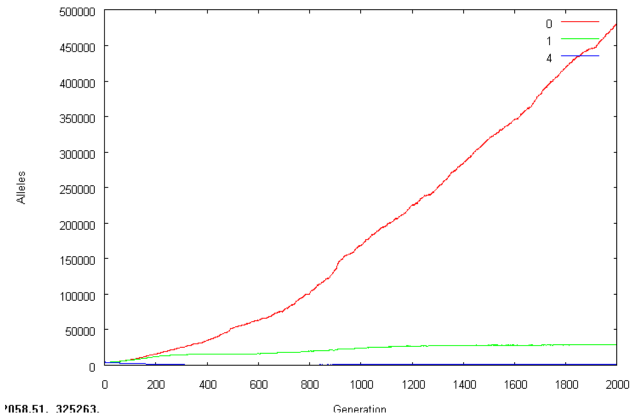
Figure 2

The second set of results are those shown in Figure 3 and 4. These are the results from the experiments with constant crossover. As expected, in Figure 3 it can be seen how at the beginning the population converges to the wider peak at 33, but as the population gains resilience through the chromosome growth and increase in 0's and 1's in the population shown in Figure 4, the population eventually converges to the narrower peak at the value 66 in Figure 3.



Average of the allele's sum over 50 trials with constant crossover.

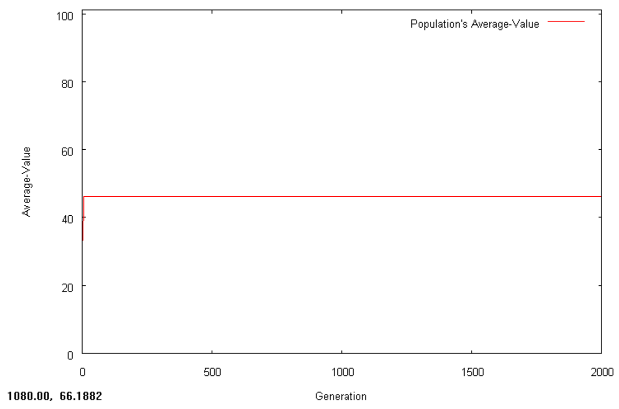
Figure 3



Average of number of 0s, 1s, and 4s during 50 trials

Figure 4

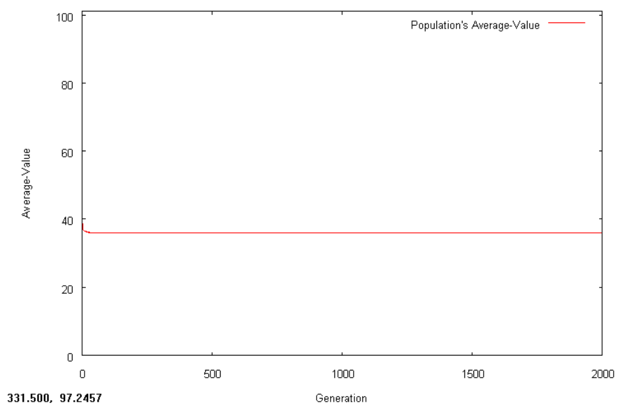
The third set of results are from the first mutation rate function described in section 2.2.3.1. These results are shown in Figure 5. They show a population convergence at an intermediate value between the two peak at around 49.



Average of the allele's sum over 5 trials with the first mutation function.

Figure 5

The fourth and last set of results are from the second mutation function described in section 2.2.3.2. These results are shown in Figure 6. This graph shows convergence at an intermediate value again, this time it is around 37.



Average of the allele's sum over 5 trials with the second mutation function.

Figure 6

3.3 Discussion

As described with the first to sets of results in section 3.2, the two experiments that reproduce the literature were satisfactory, and show results similar to those acquired by Jones and Soule (2006). This was expected.

The last two sets of results are the most interesting ones, and the ones that extend the study of resilience in genetic algorithms.

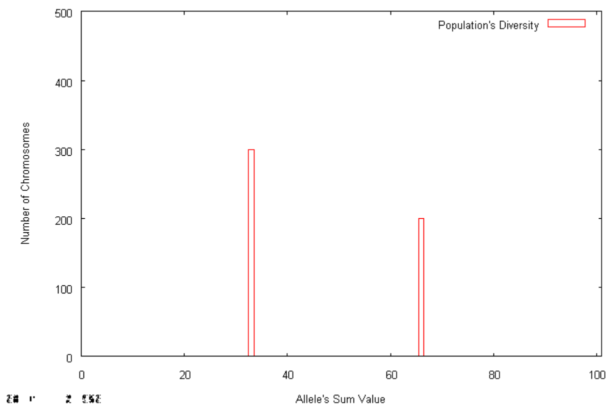
In the last two experiments as mentioned in section 2, we are only using mutation.

The hypothesis was that there exist a function that can map an attribute of a genetic algorithm to different mutation rates causing convergence to a desired place in the search space.

That desire place for these experiments is the peak at 66.

The idea behind the first function was to create a map that would assign mutation values according to the distance between a chromosomes fitness and the maximum fitness in the system. This is why in Figure 5 some convergence is seen around the value 49. This value is an average resulting from the population being divided in between both peaks as seen in Figure 7.

Now that the results are seen, it makes sense that the function behaved like this since the pressure of the function is pushing the population to the highest value, and also to those values closest to the highest value. The problem is that the highest value (26), is at one peak, while the next closest value (25) is at a completely different peak.

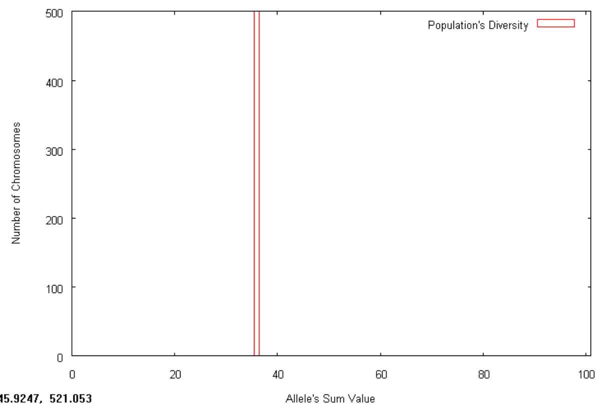


Average over 5 trials of the population diversity at generation 120 with the first mutation function

Figure 7

The last set of results are harder to analyze because the algorithm of the function and the update rule are somewhat more complex. The idea behind this function was to modify the mutation rates by pushing them in the direction in which the offspring would be more fit than the parent, and at the same time causing some pressure to reproduce at the maximum fitness seen. Apparently this expectations failed, but still some interesting graphs came out of the experiment.

In Figure 8, it can be seen that the population converged to a value around 37. But 37 is sort of a random place to converge. Even though this results don't seem satisfactory according to the initially stated measure of success, I believe that it is a successful trial. The reason why this trial is consider successful is because it proves that mutation maps can transform a landscape and cause for a population to converge to a desire place in the search space without being affected by the resilience pressure. 37 might not be a desirable place in the search space, but this only means that the parameters of the algorithm need to be adjusted, not that the hypothesis has failed.



Average over 5 trials of the population diversity at generation 120 with the second mutation function

Figure 8

4. Related Work

All of the related and relevant work made on the study of resilience in genetic algorithms has been quoted and part of it has been reproduced in this paper. There is more work in biological genetic resilience that might be relevant in the future, but as of now it is outside of the scope of this paper.

5. Future Work

Definitely more experimentation with other attributes and functions based on those attributes need to be explored.

The idea of replicating the way in which mutation is handle in biology is an interesting study. In biology the mutation rate for a given chromosome is embedded in one of the genes of the chromosome.

Also, the significance of resilience in more realistic problems can be a great stepping stone to this research making it more relevant and desirable.

6. Conclusions

Genetic resilience is a significant force in the evolutionary computation mechanism, and since most standard genetic algorithms require for the chromosome length to be fixed, the solutions proposed by Jones and Soule in their experiments (chromosome growth) become useless. This is why variable mutation rates show themselves as promising alternative to counter the resilient pressure causing convergence to a global maximum in the search space.

6. Bibliography

Jones, Soule. Comparing genetic robustness in generational vs. steady state evolutionary algorithms. In Proceedings of the 8th annual conference on Genetic and evolutionary computation. Seattle, Washington, USA 2006. 143 – 150.

Soule, T. Resilient Individuals Improve Evolutionary Search. Artificial Life, 12, 1 (Winter 2005), 17-34.