COSC 3P71 A2

Danny Nguyen

November 2019

Abstract

Genetic algorithm is an algorithm that provide a decent answer to a real-world problems we usually encounter noisy information using the theory of evolution by Charles Darwin. In this assignment, we implement a simple genetic algorithm, perform several experiments, collect and analyze the data. To demonstrate a simple genetic algorithm, we apply the concept of genetic algorithm on solving a decrypted text by generating an "optimal" key so that the result would be somewhat readable. We also discuss which parameter values and crossover methods provide better results.

1 Introduction

The purpose of this report is to demonstrate how basic genetic algorithm operates by finding a "good" key to decrypt a message. Genetic algorithm is inspired by Darwin's theory of evolution, where a certain group of people that has valuable survival characteristics will likely to survive and produce offspring. Those offspring will then retain and improve their characteristics which allow them to adapt to the environment, survive and continue to produce offspring. Genetic algorithm also include the methods such crossover and mutation in which pushing the population forward through production of new characteristics. The problem with genetic algorithm is that it does not scale with complexity. Because as complexity increases, more space needed to compute will likely increase as well, thus it will reach a point where no more space is available.

2 Background

Due to the fitness function provided, the lower the result gets, the better the key is. We will see that as generations pass by, the fitness value of the population and the best will decrease and finally flattened. The program performs by setting up all of the essential parameters such as initial population, key size, crossover and mutation rates, ... Then, the initial population is created randomly with the seed to make sure that we will get the same population if we use the same seed. First, we evaluate the initial population, print out the parameters along with the fitness value of the fittest chromosome and the average fitness value of the population. Second, we apply tournament selection, crossover and mutation. Finally, we increase the generation by one and repeat the same processes until we reach the desired generation. For the paragraph where the key length is 26, we set the maximum generation as 50 and 100 for the paragraph where the key length is 40. The reason is because after a couple runs, the population for the key of size 26, reach the point where it could not get any better and retain the same value until the end. Thus, we adjust a lower maximum generation to reduce excessive data. While the population, for the key length of 40, requires more time to reach the point where it cannot get any better. Therefore, we have different maximum generations for different paragraphs.

3 Experimental Setup

Because genetic algorithm is an idea originated from biology, Darwin's theory of evolution. The parameters are set up to represent how the evolution theory affects the population throughout generations, but simpler than in the real world. The following parameters are essential and required in order for a simple genetic algorithm to work:

- Key size: 26. We are using char array to hold individual characters that representing chromosomes.

- Population size: 5000. We are using char 2d array to hold the keys.

- Maximum generation: We adjust the maximum generation as 50 for 26 chromosome and 100 for 40 chromosome. We are using integer.

- Crossover rate: There are two crossover rate that we are using. 1.0(100

- Mutation rate: We are using two mutation rate 0.0(0

- The seed: We are using five different seeds to get different initial population and random numbers, but still be able to have the consistent in order to have a better evaluation when we use the same seed number. We are using seed 1, 2, 3, 4, and 5.

For the crossover methods, we are using two different methods, one-point crossover and uniform crossover. As for mutation, we use inversion mutation. The fitness function as well as encrypt and decrypt methods are provided from the assignment.

The following is a pseudocode that describes how the genetic algorithm works:

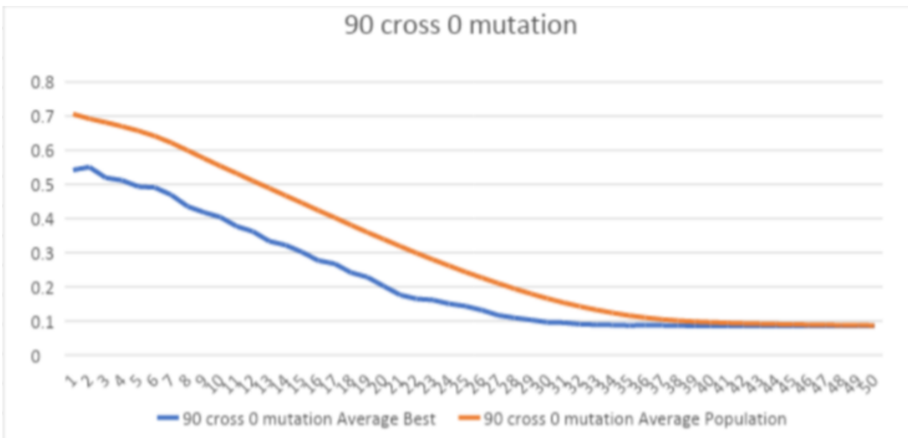
Initialize parameters

For loop from 1 to maxgen, do:

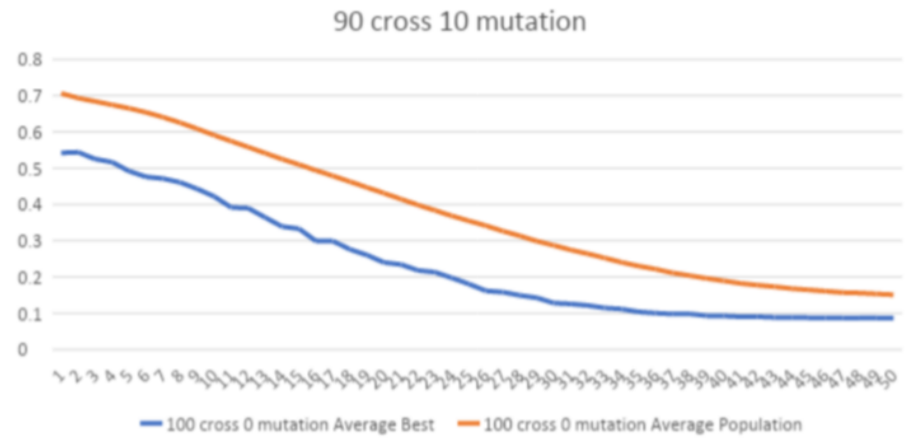
Evaluate the initial population
Increase generation by 1
Apply tournament selection
Apply crossover
Apply mutation
End of loop

4 Result

As Darwin's theory of evolution describes, the fitness value of the population as a whole and the fittest individual get better over time. Regardless of different seeds are used to generate different population and different numbers, the fitness values for both the whole population and the fittest decrease overtime and reach a point that could not be improved. However, there are generations where the fitness values seem to increase rather than decrease. This might imply that the population may not always improve, but become worse for one or two generations.

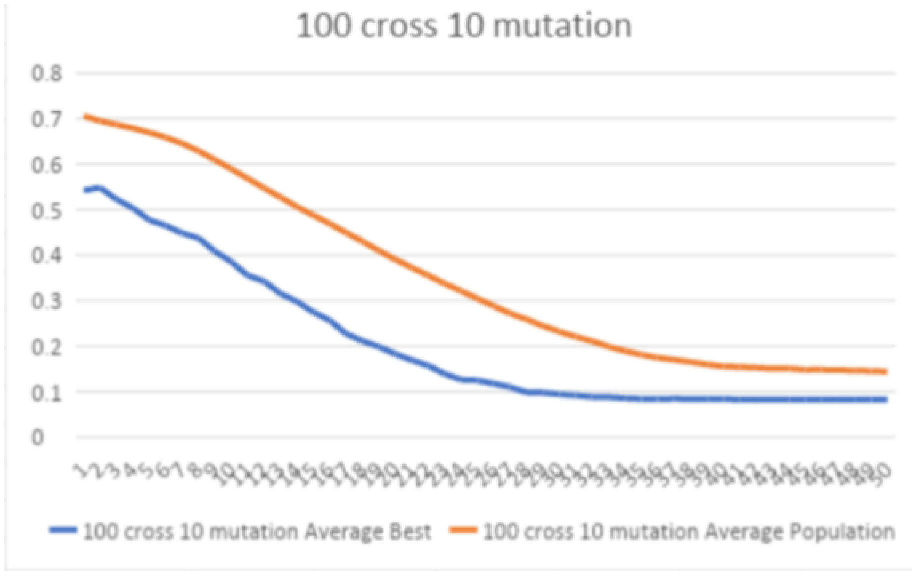


Another thing that happens is that the percentage of mutation does affect the fittest chromosome and the population as a whole. For the mutation rate of 0%, despite the percentage of crossover rate, the population as a whole would get better quite fast as the generation getting closer to the maximum generation. And eventually, the whole population has the fitness value that is close to or identical to the fittest chromosome. This means that the population is consisting of only the very best chromosomes. At first, this might be a positive result because all of the chromosomes become the best and every one is the same. But, this also means that once the fittest and the population reach the point where every chromosome is the fittest chromosome, the population and the fittest could not improve any further and stuck at that particular value. On the other hand, with the introduction of mutation of 10%, the average population is not improving at a fast pace compared to the mutation rate of 0%. Also, the average population does not have the fitness value that is close or identical to the fittest chromosome. Which means with mutation, the population will improve at a steady pace and will likely to have more genes/characteristics variety to adapt to new environments. For experiment, we try to run our genetic algorithm with 100% of crossover and 20% of mutation to see if there is any positive changes. Unfortunately, 20% mutation rate creates too much diversity which leads to the population's improvement increase steady but slower than the pace of 10% mutation. The gap between the fittest chromosome and the rest of the population is also bigger. Mutation is an important mechanism that allows the population to diverse and allows those with appropriate genes/characteristics to adapt and reproduce in new generations.

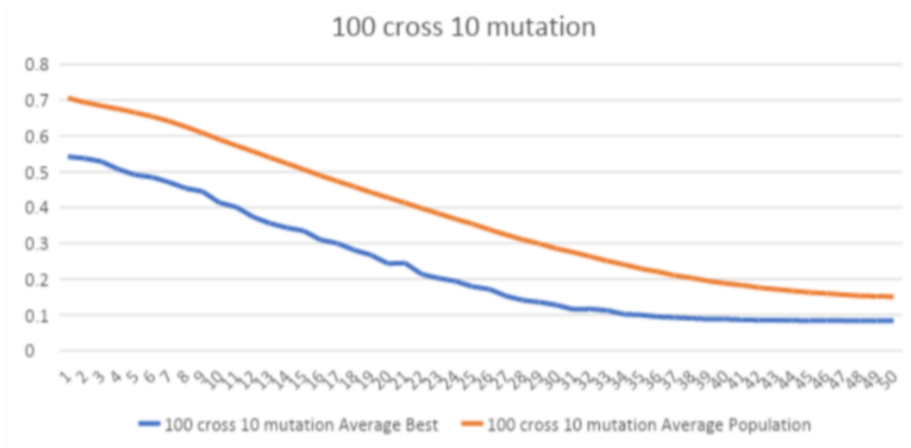


Compare to the graph of 0% mutation rate, the 10% mutation rate has slopes that are less curvy.

After analyzing the graphs between different methods of crossover, we found little difference between the uniform crossover and the one-point crossover. Both methods provide almost similar final fitness values and results from the t-test with unequal variances. However, the graphs of the uniform crossover become more curvy when the average best and average population approach its best fitness value. While the one-point crossover produces graphs such that both the average best and population have a steadier growth to its best fitness value. This difference indicates the uniform crossover allows the chromosomes to grow at a faster speed as the population starts to have improvement. This rapid growth leads to the average best and average population to reach the final fitness values sooner and retain the values for more generations compared to the one-point crossover. This rapid growth can be described as a snowball effect because of how quick the chromosomes getting better.



Uniform crossover has a curvy slopes where the average best and average population attain the final fitness value mid-way through the genetic algorithm.



One-point crossover has a steadier slopes where the average best and average population attain the final fitness value closer to the end of the genetic algorithm.

On the contrary, the one-point crossover method has its average best and average population reach the final fitness values as it gets closer to the end. Which creates a flatter slopes from the beginning to the end generations. This slower growth may reflex that there are chromosomes in the population develop genes/characteristics that are not valuable in this environment. Which is somewhat similar to how mutations introduce new changes to the average population and lead to more diversity in the population in terms of chromosomes.

This difference between the uniform crossover and one-point crossover opens more options which are suitable for different needs. With the rapid growth, but come to less diverse in the population, the uniform crossover requires less generations to allow the chromosomes to reach the final fitness value. If diversity is a more important factor, one-point crossover does need more generations or time to give the chromosome to grow in order to provide a wider, more varieties characteristics.

5 Discussion and Conclusions

This experiment represents a simple genetic algorithm and how it works by using the problems of decrypted text. By experimenting with different parameters values and different crossover methods, we are able to detect some factors that gives different chromosomes' development and end results. Those factors are mutation and crossover methods do allows chromosomes to have different approaches to the end fitness values. Mutation rate introduce new genes/characteristics that may benefits for when the environment changes. One-point crossover also provides similar result as mutation but not as impact as mutation. For uniform crossover, the method gives a rapid growth for the chromosomes, thus, it requires less time to run and to attain the best fitness value.