Decryptying A Vignere Cipher With A Genetic Algorithmically Generated Key

Jordan Wallace (7025927)
Computer Science
Brock University
St. Catherines, Canada
jw20kx@brocku.ca

I. Introduction

The Vigenere cipher is a classical method of encrypting alphabetic text that uses a simple form of polyalphabetic substitution. In this report, I present a genetic algorithm implemented in Java to decrypt text encrypted using the Vigenere cipher. The genetic algorithm aims to generate an optimal key for decryption, utilizing the principles of natural selection and evolution.[1]

II. Background

Initialization:[1]

Generates an initial population of decryption keys that are stochastically determined. The number of decryption keys is equal to the population size.

Fitness Evaluator:[1]

Assesses the fitness of each individual in the population by comparing the decrypted text with an English letter frequency model. A lower fitness value means a better solution.

Selection:[1]

The parents of each new generation are determined through tournament selection. A selection process where *K* individuals are chosen from the population at random, leaving only the most fit of those individuals to move onto the crossover stage.

Crossover:[1]

N (in this case, 2) chromosomes are selected, then they swap genes to produce *N* offspring. These offspring are the individuals that make up the next generation. This is repeated until there are no more parent chromosomes available. Uniform and two-point crossover are used in the experiment.

Mutation:[1]

Randomizes the genes of chromosomes in the population depending on the mutation rate. This is done to maintain diversity within the population.

III. Experimental Setup

There were two different experiments. The parameters for each experiment varied, so a distinction is made here.

Parameters (Population Test):

Each test utilized uniform crossover and had a constant *K* value of 3. The maximum generation span was capped at 50 for each test. The most frequently changed parameter between trials was “Population Size”. Per each given preset of parameters, there were 5 different populations. The populations are as follows: 500, 400, 300, 200, 100 in this order. This was repeated with a seed of ‘1’, ‘2’, ‘3’, ‘4’ and ‘5’ respectfully.

Parameters (Crossover Test):

Throughout each test, a population size of 500 remained constant, as well as a *K* value of 3, and a maximum generation span of 50. There were 2 tests ran with each of the following preset parameters:[2]

1. Crossover Rate: 100%, Mutation Rate: 0%
2. Crossover Rate: 100%, Mutation Rate: 10%
3. Crossover Rate: 90%, Mutation Rate: 0%
4. Crossover Rate: 90%, Mutation Rate: 10%
5. Crossover Rate: 100%, Mutation Rate: 5%

This was repeated with a seed of ‘1’, ‘2’, ‘3’, ‘4’ and ‘5’ respectfully.

IV. Results

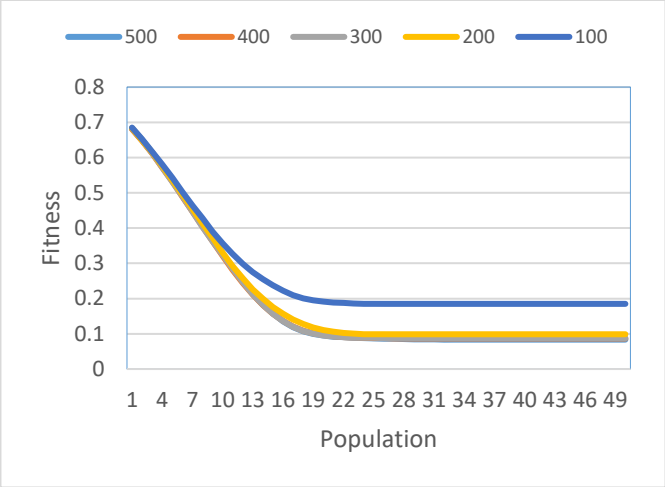
Population Test

The resulting data shows that an early convergence is negatively correlated with the size of the population, meaning on average, the most densely populated trials performed better than their lower population counterparts. The breakdowns of each population are as follows:

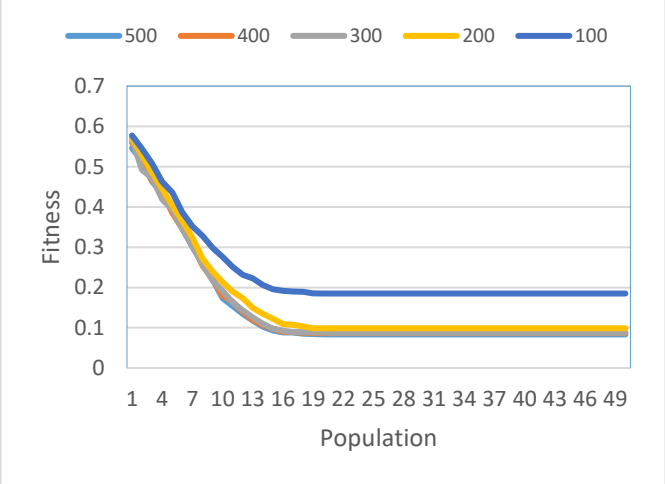
Crossover = 100%, Mutation = 0%. This is true for all the following results. The results are roughly the same for all preset parameters.

A. Data Set #1

Graph I. Average Fitness

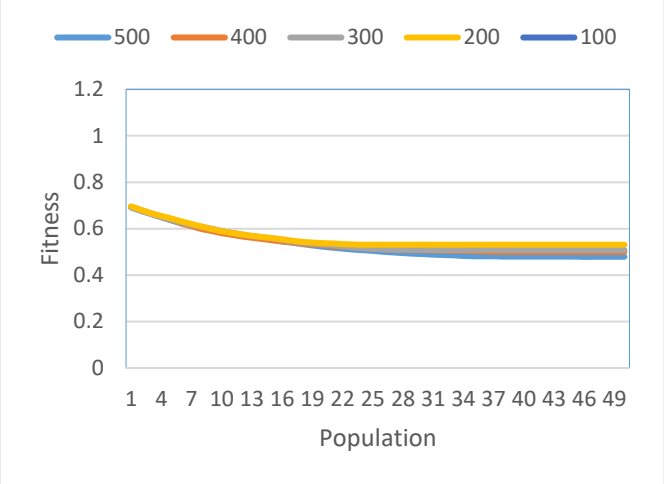


Graph II. Best Fitness

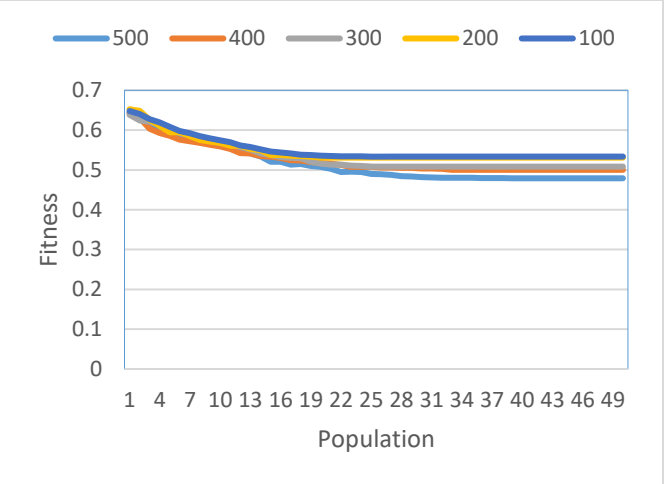


B. Data Set #2

Graph III. Average Fitness



Graph IV. Best Fitness



As seen by the above graphs, the data converges noticeably early with a population size of 100. The differences between population size and early convergence aren't as noticeable starting at a population of generation 200. This, however, doesn't mean that there is not a difference between a population of 400 and a population of 500, as a population of 500 always performs better than a population of 400 on average.

Using a T test by comparing the averages at population 500 vs. population 100, a P value of 2.94153505570219E-21 is given. This means that the null hypothesis can be rejected, meaning that the two populations have a statistically significant difference.

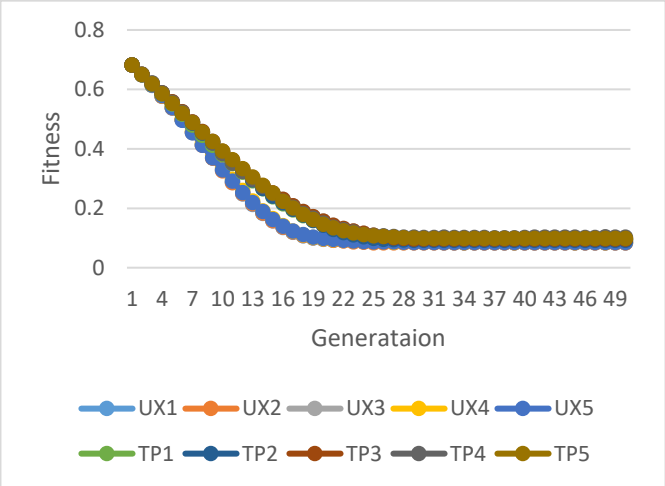
CROSSOVER TEST

The resulting data shows that uniform crossover performs better than two-point crossover for both data sets. For the following charts, UX: Uniform crossover, TP: Two-point crossover. Each seed runs the five preset parameters listed in Section. III.

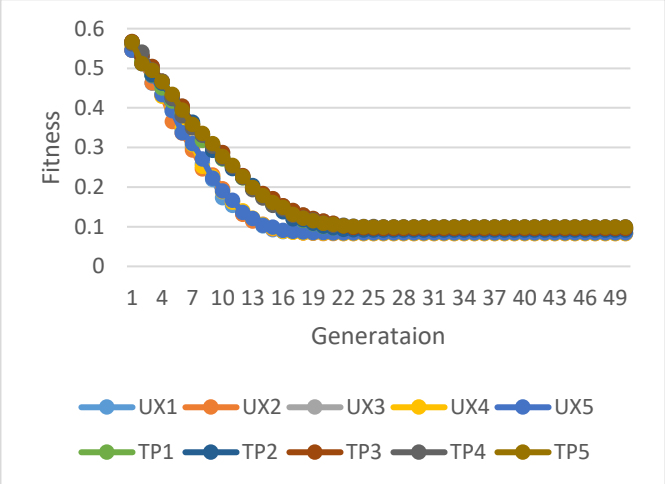
Each of the tests had a population size of 500. The difference between the data points, while less distinct than the population tests, are present.

A. *Data Set #1*

Graph V. Average Fitness
Note: The top 5 lines are two-point crossover



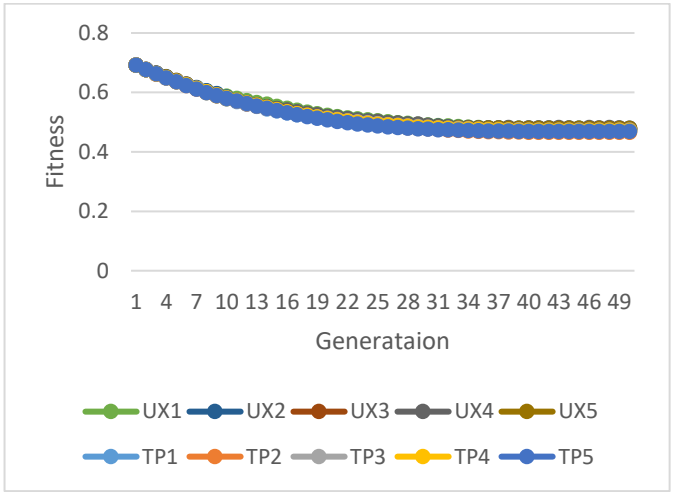
Graph VI. Best Fitness
Note: The top 5 lines are two-point crossover



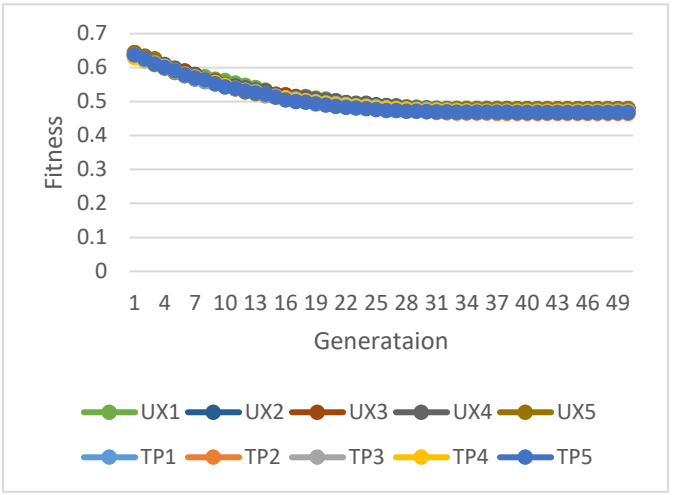
When running a T-Test on the best UX average vs. the best TP average for data set #1 (both occurring at Crossover Rate: 100%, Mutation Rate 10%), the P value is 1.0746050859729E-08. This is significantly smaller than the given alpha value (0.05), meaning that we can reject the null hypothesis. Therefore, the difference between uniform crossover and two-point crossover is statistically significant.

B. *Data Set #2*

Graph VII. Average Fitness
Note: The top 5 lines are two-point crossover



Graph VIII. Best Fitness
Note: The top 5 lines are two-point crossover



When running a T-Test on the best UX average vs. the best TP average for data set #2 (both occurring at Crossover Rate: 100%, Mutation Rate 10%), the P value is 2.20809703928881E-19. This is significantly smaller than the given alpha value (0.05), meaning that we can reject the null hypothesis. Therefore, the difference between uniform crossover and two-point crossover is statistically significant.

REFERENCES

[1] Dr. Ombuki-Berman, B 2023. ”GA_Assignment_slides” [Powerpoint], COSC 3P71: Introduction to Artificial Intelligence. Brock University. 10, November.

[2] Dr. Ombuki-Berman, B 2023. ”Assignment_2_2023” [PDF], COSC 3P71: Introduction to Artificial Intelligence. Brock University. 10, November.