Assignment#4 Report

In this project, my GA can find most of right keys from the test. But sometimes I still need some “luck”. This is, not always return the exactly right key, but not too far from it.

In the experiment, I tried to use both Steady-Stated GA and the Generational GA. The result shows the Generational GA works better. In the Fitness Function choice, I tried both Euclidian distance fitness function and Bhatthacaryya. And the result shows Bhatthacaryya works better. I also tried to modified the Fitness Function. My idea is, if the frequency from the English table is bigger than 0.008 (I get this by 600/71987, since 71987 is the SUM of the English frequency), gets 1.5 times of fitness(Bonus!), and if the frequency is smaller than 0.001, then it gets 0.5 times of fitness(Punishment!). This change helped my algorithm find a better result.

The detail of my GA is list below:

GA	Generational GA
Generation Times	500~1000 (Depend on Text Length)
Population	50 -280 (Depends on Text Length)
Fitness Function	Bhatthacaryya
Xover Type and Rate	Uniformed Xover / 50% rate
Mutation element	2 ~ 3

In this experiment, I could observe that the longer text it easier to find the result. This is because of the longer text can be easier to applied to the frequency table, and get more samples for finding a more common English word. In my opinion of this project, we don’t really need to find a exactly same key to figure out what is the “Real Message” is. Since we can find a key which is close enough (it’s hard to say how enough, but I guess 2/3 of right position of a key should be good enough), the decoded message will be more “Readable”. For example, for a human who’s native language is English, and ask him to read “Hallo, My nama is Sheung”, he may be easy to recognize this message is “Hello, My name is Shaung”. Apply to the real life, cooperate with this algorithm and human brain, people can decode the message well. Another of my opinion, this project is really fun, thanks Dr. Heckendorn!