Genetic Algorithm

Simar Bhamra
Computer Science
Brock University
St.Catharines, Canada
sb17ho@brocku.ca, 6364665

***Abstract*—This document is a model to demonstrate how Genetic Algorithms works. Explaining what Genetic Algorithm is and how it is implemented to provide optimal solution for finding keys with minimun fitness to decrypt a given Ciphertext.**

I. Introduction

Genetic Algorithm is a search based algorithm that works towards providing an optimal solution for a given problem. It initialises a random population of chromosomes which undergo selection, crossover and mutation. It is based on the concept of Darwinian Principle of reproduction and survival of the fittest. It works towards providing an optimal solution by genetically reproducing a population of individuals over a series of generations.

Parent Selection is applied in genetic algorithm to find the fittest candidate from the entire population and apply crossover and mutation on the selected individuals. The different types of parent selection methods are Tournament Selection, Rank Based Selection, Fitness-Proportionate Selection, Roulette Wheel Selection.

Crossover is also referred to as recombination, it is a genetic operator used to combine the genetic information of the two individuals to generate a new offspring. There are different ways such as 1-Point Crossover, N-Point Crossover, Uniform Order Crossover, Order(OX) Crossover, Partially Mapped Crossover, Cycle Crossover etc. to generate new offsprings from the existing individuals in a population to create a new population.

Mutation is like crossover, it is also a genetic operator but instead to working on 2 individuals, it on a single individual to maintain genetic diversity from one generation to another generation to provide a more optimal solution. The different types of mutations are Inversion, Insertion, Scramble, Displacement and Reciprocal Exchange etc.

Genetic Algorithm is being used to find the solution i.e. a key used to encrypt text with the Vigenere cipher. Encryption of the data is important as we store a lot of data in form videos, pictures and various other relevant forms online. In order to preserve the privacy of the data, therefore a number of Crypto-systems are used for this purpose. The researchers use a number of algorithms to ensure the security of the data and preserve it from malware attacks. Thus the solution is necessary so as to prevent privacy breaches.

II. Background

A. Genetic Algorithm

Genetic algorithm is a search technique which is based on the evolution strategy to find the best possible solution using a heuristics. It involves parent selection, crossover, mutation etc. In genetic algorithm, we generally initialize a population with any random characters, integers etc., the initial population is evaluated for each generation where the purpose of the generation is to evolve or basically find the best possible solution, after the evaluation of the initial population, the characters/integers are selected through any of the selection methods such as tournament selection, rank based selection, fitness proportionate selection etc. The purpose of a genetic algorithm is to provide the best solution for a provide problem.

- **Tournament selection** k candidates compete with each other (k=2,3..,5), if the value of k is quite large then it results in premature convergence.

```
Tournament Selection(pop, k)
// pop is population and k is the number
    of competitors
for i in range(k) //where k=2,3,..,5
curr_candidate = random chromosome from
    pop
if best_candidate is empty:
best_candidate = curr_candidate
else curr_candidate.fitness <
    best_candidate.fitness:
best_candidate = curr_candidate
return best_candidate
```

- **Fitness Proportionate Selection** the fittest in the population dominates the unfit chromosomes for mating and propagating the features to the next generation resulting in premature convergence. The selection implies more selection pressure to the more fit individuals in the population.
- **Rank based selection** applies for the negative fitness as well, the purpose of this selection was to overcome the issues with Fitness Proportionate Selection, thus leading to equal evaluation of the individuals in the population, however it leads to poor parent selection as the fittest are evaluated equally.

1) Crossover: Crossover is analogous to reproduction and biological operators. Crossover takes in two parents and swap the characters/integers between the 2 parents depending on the

type of crossover used - N-point crossover, Uniform Crossover, Uniform Order Crossover etc.

- **1-Point/ N-Point Crossover** it takes 2 parents, we take a random crossover points, the child 1 takes the characters/integers from parent 2 to the left/right of the crossover point and the remaining characters from parent 1, whereas the child 2 takes the characters/integers from parent 1 and the remaining from parent 2.

```
One-Point Crossover(Chromo 1, Chromo 2)
generate a list of size equal to
    chromosomes
Initialize list with random decimal
    numbers
for length of list
   if value at index i is less 0.7 in
       list
      child 1 gets genes from
          individual 1
      child 2 gets genes from
          individual 2
   else
      child 1 gets genes from
          individual 2
      child 2 gets genes from
          individual 1

return chromosomes
```

- **Uniform Crossover** we create a list of size equal to parents of random numbers between 0 to 1 (can be decimal values), we iterate through the list and where we find value equal to 1, the character is the same index of the children where child 1 gets the character from parent 2 and child 2 gets character from parent 1, if its lower than 1 then rest of the characters of the children are added from parents. (correct a bit)
- **MixNMatch Crossover** we pick a random point similar to what we do in 1-Point Crossover, however the random point should be at least of length 4, after we get a random point instead of child 2 getting genes from parent 1 and child 1 getting genes from parent 2, we generate a list of random decimal values of length equal to random point, if the value less than 0.7 the children get genes from opposite parents up to length of random point and the remaining genes from the same parents.

```
MixNMatch Crossover(individual 1,
    individual 2)
random point within the chromosome
for size of random point
   Generate a list of random decimal
       values
for length of list
   if value at index i is less 0.7 in
       list
      child 1 gets genes from
          individual 1
      child 2 gets genes from
          individual 2
   else
      child 1 gets genes from
          individual 2
      child 2 gets genes from
          individual 1

return children
```

2) Mutation: Mutation can be simply defined as the random tweaks that can be done in the chromosomes to the chromosome values through chromosome operators such as Inversion, Insertion, Displacement, Reciprocal exchange, scramble mutation.

- **Inversion Mutation** we pick two random points say left and right, after selecting the two points we invert the characters between the two points

```
Mutation(individual)
Take two random alleles(left<right)

for i=left to right:
   Invert the substring

return the individual
```

3) Evaluation Function: Evaluation function is used to evaluate the chromosomes within a population. Evaluation function determines how good the chromosome/candidate is based on the problem.

The fitness function/ Evaluation Function used takes in cipher text and key as parameter and initialize the list with the expected frequency of the occurrences of alphabets in English. The cipher text as well as the is then sanitized i.e. eliminating the unwanted characters. ADD MORE

III. Experimental Setup

In the given problem where we used Genetic Algorithm in order to Crypt-analysis since the security of cryptography systems is more important than ever. The genetic algorithm implemented for the the purpose takes in three parameters key-size, cipher and comp (competitors) where key-size takes the size of possible key, cipher is the cipher-text provided for decryption and comp i.e. competitors is the number of competitors for the tournament selection, generally there shouldn't be a large number of competitors to avoid selection pressure, here there are 3 competitors with global parameters being the generations, crossover-rate, mutation-rate, global-best, initial population and new population. Every generation has a seed such that each run has a particular set of random values, helpful for debugging purposes.

Within the GA method, the initial population is initialized with random individuals or chromosomes with an initial evaluation of zero. After the initialisation, the individuals in the population are evaluated with respect to the provided cipher-text. Then within the for loop for n number of generations, the initial population is sorted in ascending order of the evaluated or fitness value and the fittest individual is considered as the global best for the current population.

After finding the global best, new population is initialised with random individuals of fitness zero and a population

tracker is initialised to track the individuals in the population. Then a while loop is initiated with condition until the new population size is less than the initial population size, within the while loop parent selection algorithm i.e. Tournament Selection is used to get best candidate or individual from the population.

- In tournament selection, best is initialised with a null value. Then for each individual in the population, a random individual is selected, if the best-candidate is empty assign the individual to the best-candidate, else if the individual's fitness is better than the best-candidate's fitness, replace the best-candidate with the current individual.

After parent selection, if the random real number generated is less than the crossover-rate, then the 2 candidates selected undergo crossover using either 1-Point Crossover or Uniform crossover or MixNMatch Crossover,

- In 1-Point Crossover, the chromosomes of the individuals obtained after parent selection passed to the method, where a random crossover point is taken, the child 1 gets the chromosomes from parent 2 to the right of the crossover point and the remaining chromosomes from the right and vice versa in case of child 2.

else if the random number is greater than crossover rate then return the candidates selected via parent selection. After crossover, the individuals are added to the new population with an evaluation or fitness score of 0 and the population size is increased by 1.

After the new population is generated, mutation is applied on a percentage of population depending on the mutation-rate. After mutation, the new population is evaluated on the with respect to the cipher text and elitism is applied to check if the best individual from the previous population is present in the current population as well, it not replace it with worse, else update the best individual if a better one is found. After the elitism, the previous population is replaced by the new population. For each generations, same steps are followed to generate new populations, after n numbers of generations, a final new population is obtained that contains the final key which can provide an optimal solution for the given cipher text.

IV. Results

After performing experiments which includes running the genetic algorithm 5 times for 5 different random number seeds with different genetic algorithm parameters.

As the tests are expected, we observe that the cipher-text with a key size of 26 gives a fitness value of approximately 0.13759809173083362 and generate a key *hisisasupersecurepasswordt* which is then used to decrypt the provided the ciphertext. From the graphs i.e. graph for best fitness for each generation and for average fitness for each generation shown below we can see that after few generations, the graph becomes steady since a key with best fitness value is found after a certain number of generations

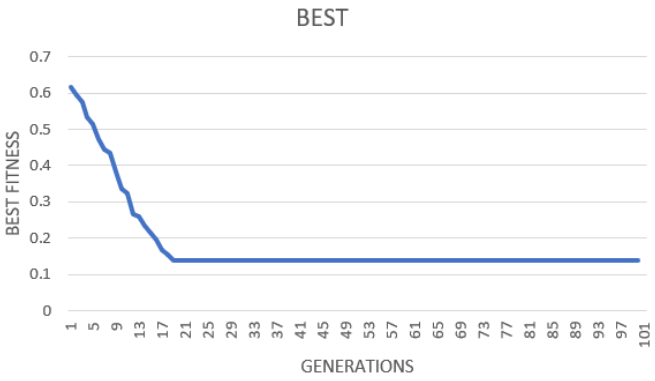

Fig. 1. Best Fitness Graph

As we can observe from the above graph, the graph gradually falls since initially the population has random chromosomes, but as the generations increases the new population developed through mutation and crossover has the chromosomes with provides the solution to the problem and lower fitness determine the optimality of the solution. The graphs shows the best fitness for each generation.

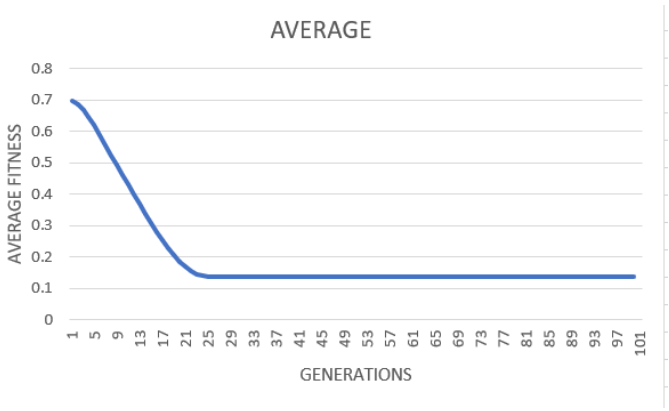

Fig. 2. Average Fitness Graph

Whereas the graph above determine the average of the fitness of all the chromosomes in a population for each generations and gradually as the generations increases the graph become linear since most of the chromosomes become same and have near to similar fitness. The average graph is smooth compared to the best fitness graph.

Whereas the cipher-text with a key length of 40, has the best fitness of 0.451801148 and generate *knltitaaueajrpitimiaippjluczmvpaiituauic* as the best possible key as the optimal solution for the given problem i.e. decryption.

As from the graph above it can be observed that as the initial population is generated the initial best fitness value achieved is 0.64631974, however as the generations increases the graph gradually falls same as for key size 26.

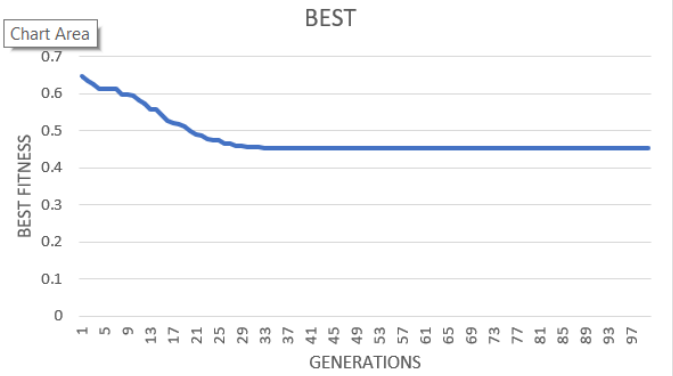

Fig. 3. Best Fitness Graph

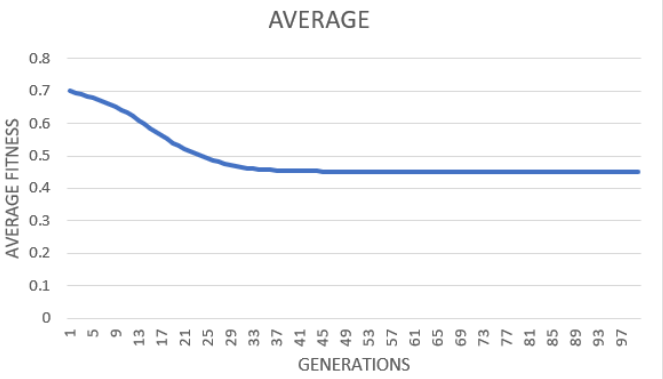

Fig. 4. Average Fitness Graph

However from the graph for the average fitness of the population, we observe the graph gradually falls same as the best fitness graph, however the fall is smooth compared to the best fitness value.

Thus from the observation we find that since the graphs are smooth and does not have immature changes i.e. increase or decrease in the fitness value of the chromosomes, thus there are not premature convergence.

From the experiments it is observed that the Uniform Crossover provides more optimal solution compared to the MixNMatch Crossover and One-Point Crossover as shown in the graph for the key size 26.

From the data analysed and the graphs, we observe that the uniform crossover provides the most optimal solution or lower fitness value followed by MixNMatch and One-Point crossover. Different parameters have different effect on the fitness value of the chromosomes in a population such as different mutation and crossover rates have different effect on the fitness.

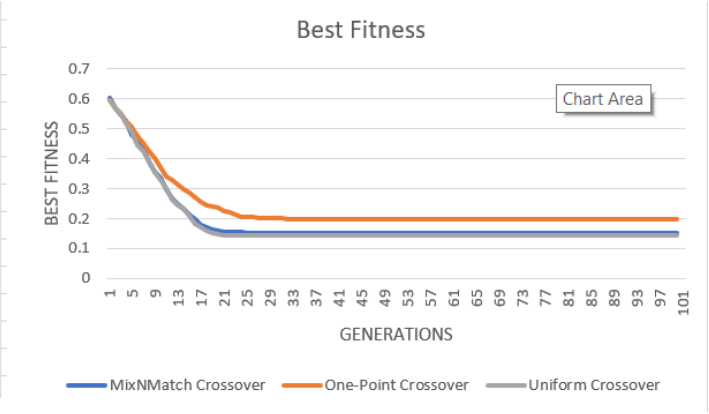

Fig. 5. Best Fitness Graph

TABLE I
ONE-POINT VS UNIFORM CROSSOVER

	One-Point Crossover	Uniform Crossover
Mean	0.237474942	0.182950756
Variance	0.008263465	0.010454256
Observations	100	100
Pooled Variance	0.00935886	
Hypothesized Mean Difference	0	
df	198	
t Stat	3.98531555	
P(T¡=t) one-tail	4.73491E-05	
t Critical one-tail	1.652585784	
P(T¡=t) two-tail	9.46981E-05	
t Critical two-tail	1.972017478	

V. CONCLUSIONS

As per the experiments, we observe that the Genetic Algorithm performs optimally in case of 26 key string and provides optimal fitness value whereas the Genetic Algorithm does not optimally compared to 40 key string thus giving higher fitness values.

The graph demonstrates that at crossover rate 100 and mutation rate 10, the Genetic Algorithm performs optimally in case of 26 Key whereas not in case of 40 Key length, thus we can say Genetic Algorithm does guarantee an optimal solution for every problem.

Every crossover has a different effect on the fitness of the chromosomes in a population as seen in *fig 5* as well as the table shown below and table 1, where the Uniform

TABLE II
MIXNMATCH VS UNIFORM CROSSOVER

	MixNMatch Crossover	Uniform Crossover
Mean	0.191387354	0.182950756
Variance	0.01004512	0.010454256
Observations	100	100
Pooled Variance	0.010249688	
Hypothesized Mean Difference	0	
df	198	
t Stat	0.589246537	
P(T¡=t) one-tail	0.278183805	
t Critical one-tail	1.652585784	
P(T¡=t) two-tail	0.556367611	
t Critical two-tail	1.972017478	

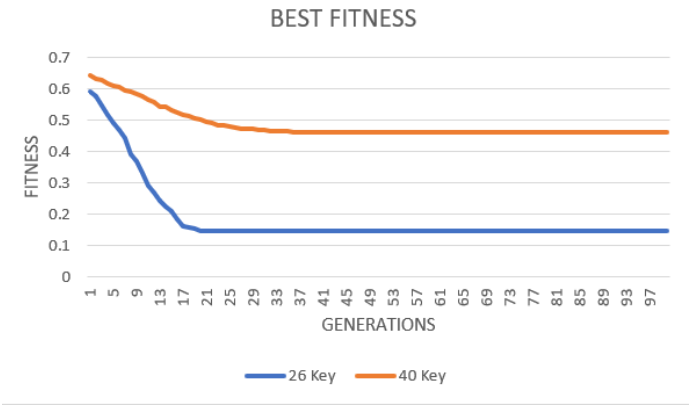

Fig. 6. Best Fitness Graph

Crossover perform best followed by MixNMatch Crossover and One-Point Crossover, where One-Point Crossover has the highest best fitness compared to others.

Genetic Algorithm is not the most reliable form of evaluation or for finding optimal solution for a given problem as seen in case of finding the best key for decryption of the given ciphertext. In Genetic Algorithm the premature convergence is dependent on the initial population, thus effecting the further generations generated.

REFERENCES

REFERENCES

[1] ”Crossover (genetic algorithm)”, En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm). [Accessed: 04- Nov- 2019].

[2] ”Genetic Algorithms - Population - Tutorialspoint”, Tutorialspoint.com, 2019. [Online]. Available: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_population.htm. [Accessed: 04- Nov- 2019].

[3] Dr. Beatrice, O. (2019). COSC 3P71 Genetic Algorithm lecture slides.