

Genetic Algorithm Research Project

Pei-Jen Chen, Mark Joseph Kenny, Jason Hua, Alex Anisimov
Rose-Hulman Institute of Technology, CSSE220

10th November, 2023

1 Introduction

In our genetic algorithm project, we developed a simulation that allowed users to define and manipulate individual chromosomes over a user-specified number of generations. Each chromosome in our simulation represented a collection of genes, where each gene could be characterized by a binary bit, either 1 or 0. The primary objective of our simulation was to evolve these chromosomes towards an optimal state, where all gene bits were set to 1, reflecting the most 'fit' chromosome.

1.1 Chromosome Representation and Evolution

The fundamental building block of our genetic algorithm is the chromosome, which is composed of genes, where each gene bit can take one of two values, 1 or 0. The objective of our simulation is to evolve these chromosomes to an optimal state, where all gene bits within a chromosome are set to 1. We refer to this state as a "perfect" chromosome, signifying the highest fitness.

1.2 Mutation and Evolution

To introduce diversity and facilitate evolution, we incorporated mutation into our simulation. During each generation, a random chance is applied to each gene, which may lead to flipping its bit value from 0 to 1 or from 1 to 0. The rate of mutation was customizable, allowing for different levels of genetic diversity within the population.

1.3 Optimization Objective

Our primary objective was to evolve the population over a user-specified number of generations while striving to achieve perfect chromosomes. This involved continuously assessing the fitness of each chromosome and selecting the fittest individuals for reproduction in each generation. The genetic algorithm aimed to mimic the natural selection process, favoring those chromosomes that were closer to the perfect state.

1.4 Variation in Mutation Methods

In our simulation, we explored various mutation methods to investigate their impact on the evolution of chromosomes. These methods encompassed a range of mutation rates, which allowed us to study the effects of mutation frequency on the rate of convergence to optimal solutions. We systematically compared different mutation strategies to gain insights into their effectiveness.

2 Basic Experiments

2.1 Experiment 1: Without crossover

Parameter	Value
Bit String Length	100
Population Size	100
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	False
Generations	500 (Terminate at Max Fitness)

It should terminate around 200 generations or so. The best chromosome is preserved in each generation. Without crossover, the best chromosome and each proceeding chromosome are copied twice and mutated. The best chromosome will be stored regardless of the mutation, ensuring the line of best fit never goes down, and that no adverse effects from mutation will affect the best chromosome.

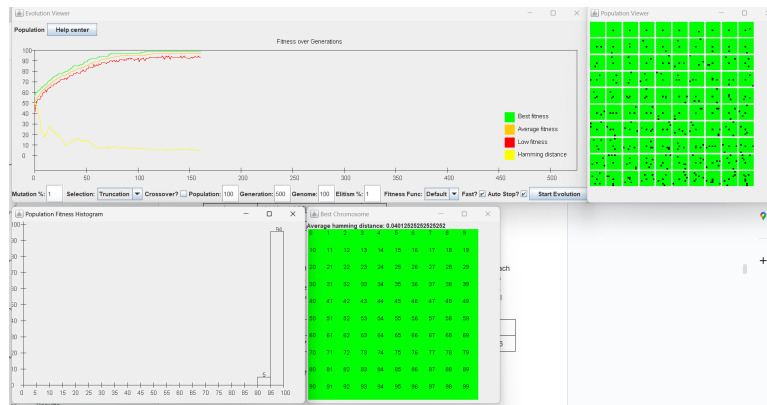


Figure 1: Simulation of Experiment 1

Trial	1	2	3	4	5	6	7	8	9	10
Generation	141	132	137	148	162	98	210	103	147	146

The results aligned better than we expected, with an average of 142 generations. However, if the mutation rate was higher the uncertainty in our result would increase as that would cause too much volatility in the experiment. Furthermore, if the elitism rate was too high, it would reduce diversity and therefore take more time to achieve the perfect solution.

2.2 Experiment 2: The number of generations it takes to find the perfect solution with crossover

Parameter	Value
Bit String Length	100
Population Size	100
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	True
Generations	500 (Terminate at Max Fitness)

We assumed that by enabling crossover, it should be able to do it better than without crossover. This should be around 130-150 generations.

The best chromosome is preserved in each generation. With crossover, the best chromosome and each proceeding chromosome have crossover conducted with each other and mutated. The best chromosome will be stored regardless of the mutation, ensuring the line of best fit never goes down. Given the benefits of crossover, the search space should be narrowed down further given that the two best chromosomes are combined twice and then mutated to potentially give a better version. This also increases the likelihood of better-quality chromosomes in the newer generation than the last in general.

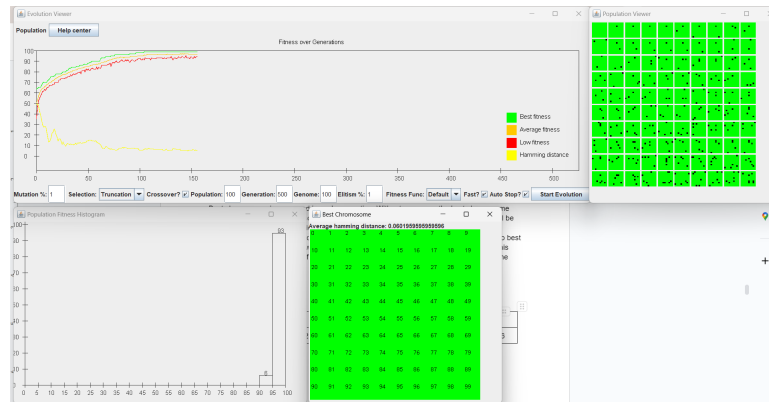


Figure 2: Simulation of Experiment 2

Trial	1	2	3	4	5	6	7	8	9	10
Generation	157	121	182	148	159	176	159	251	124	156

The experimental results were almost identical to those of when crossover was disabled. We thought that this could be caused by the small population size, which can limit the ability of crossover to have any real effect on the number of generations required to reach the perfect solution.

2.3 Experiment 3: The number of generations it takes to find the perfect solution without mutation

Parameter	Value
Bit String Length	100
Population Size	100
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	0%
Crossover	True
Generations	500 (Terminate at Max Fitness)

Due to the 1% elitism, the best chromosome is preserved in each generation. Given that the best chromosome is being bred with the 2nd best chromosome, this is more likely to produce a worse chromosome than a better chromosome, especially since having no mutation will also greatly reduce diversity. As the weaker chromosomes are gradually removed, and each chromosome is bred with more similar chromosomes due to the lack of diversity caused by the lack of mutation, the population converges to a single state.

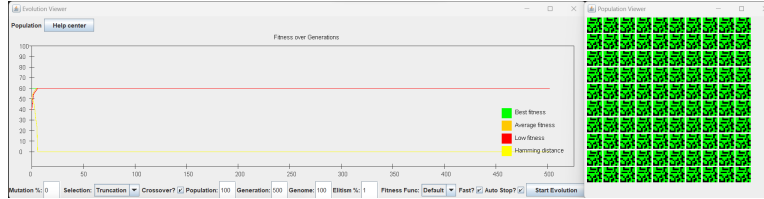


Figure 3: Simulation of Experiment 3

Trial	1	2	3	4	5	6	7	8	9	10
Generation	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

From Figure 3 and the table, it can be seen that after a few generations, the best fitness line remains constant and never reaches 100, as there never is a better chromosome than the initially generated chromosome. While the worst and average fitnesses increase, this is because they have converged to the best fitness chromosome, which means that it remained constant throughout all the generations. The convergence of the population to a single chromosome also causes the diversity to decrease completely, as displayed by the hamming distance falling.

2.4 Experiment 4: The effect of population size on the number of generations it takes to find the perfect solution

Parameter	Value
Bit String Length	100
Population Size	Variable from 100 to 1000
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	True
Generations	500 (Terminate at Max Fitness)

Due to elitism, the best chromosome is preserved in each generation. Without crossover, the best chromosome and each proceeding chromosome are copied twice and mutated. The best chromosome will be stored regardless of the mutation, ensuring the line of best fit never goes down. Given the benefits of crossover, the search space should be narrowed down further given that the two best chromosomes are combined twice and then mutated to potentially give a better version. This also increases the likelihood of better-quality chromosomes in the newer generation than the last in general. A larger population size means that there is more diversity in the population, which gives the GA more options to explore. This can help to prevent the GA from getting stuck in local optima.

Population	100	200	300	400	500	600	700	800	900	1000
Generation	146	134	113	105	95	93	85	85	72	66

This lined up with our expectations and is likely, as mentioned before, due to the genetic algorithm having more of a search space to go through.

3 Reproducing Results from a Classic Paper

To reproduce the results from the research paper, we had to add a third type of bit to the gene. This bit is '?'. It can either be a 1 or a 0. Each chromosome lives 1000 days per generation. Each day, the '?' is flipped between a 1 and a 0. This is the determining factor that helps decide which chromosomes are used in mating to pass their genetic data to the next generation. The higher its fitness, i.e. the earlier it reaches the good net (all 1s), the more likely it is to be selected for mating.

We expect our graph to match the results of the research paper. By this, we believe that the number of correct alleles (1s) will slowly increase, while the number of incorrect alleles (0s) will quickly fall down to 0, and the number of undecided alleles (?s) will slowly decrease. The reason why we expect the number of incorrect alleles to quickly go down is because of the way fitness is determined. If a chromosome's genetic data has an incorrect allele, this automatically means that it will never reach the good net. As such, it will always

have the lowest fitness, and thus a very high chance of not being selected for breeding. This effect translates from generation to generation, filtering out all the 0s until there are none left. On the other hand, this causes both the number of 1s to increase, and slowly decrease the number of ?s, given that chromosomes with lower numbers of ?s are more likely to reach the good net faster than the ones with higher numbers of ?s. This filtration eventually results in the number of 1s increasing while the number of ?s decreases.

Parameter	Value
Bit String Length	20
Population Size	1000
Selection	Research
Elitism	0% (i.e. No Chromosome Cloned)
Mutation Rate	0%
Crossover	True
Generations	100

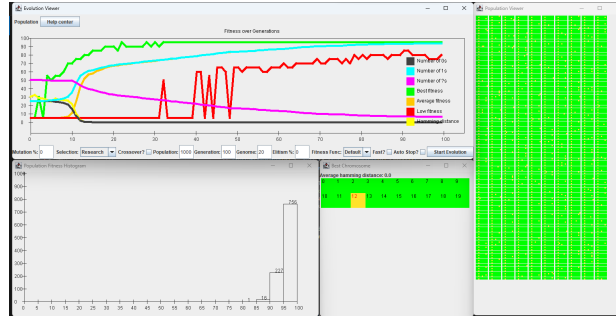


Figure 4: Simulation of Reproducing the Classic Paper

As seen in the image, the number of incorrect alleles (0s) goes down as soon as the number of correct alleles (1s) goes up. The number of undecided alleles (?) goes down as well, though the number of incorrect alleles quickly falls to 0, leaving only the correct and undecided alleles in the population. This aligns similarly with the results of the scientific paper (using the same parameters) and with our expectations.

The reason the red line fluctuates is because of the possibility that some chromosomes will never be able to reach the perfect net before the end date, because the question marks always change to 0, and never to 1. Hence, their fitness would be the minimum possible score after that, causing the spikes. The reason why our best fitness does not spike immediately is because it is dictated by the probabilistic score of it reaching the good net, and it is only as the number of ?s decreases that it will be able to increase higher and higher.

4 Further Experiments: Bit String Encoding Fitness Function

We incorporated a binary fitness function. This fitness function encodes the genetic data of the chromosome into a decimal number. The closer the decimal number is to a specific number, the higher its fitness. To determine the fitness, we used the formula:

$$\frac{1}{1 - |TargetNumber - ChromosomeNumber|} * 100$$

We set numbers that were only multiples of 10s in terms of binary, and which were lesser than or equal to 30 in total number of digits. This was to be able to more easily see the impact of the fitness function in the simulation.

We will be testing out the experiments by changing one other parameter as well. This is elitism, from 1% to 0%. We believe that the best fitness line will show no spikes in the 1% elitism given that the best chromosome is always preserved. This ensures that any mutations in the best chromosome that make it worse will not be carried over to the next generation. On the other hand, with 0% elitism, we believe that the best fitness line will exhibit multiple spikes. This is because any mutation of a bit can dramatically affect the value of the encoded number of the chromosome's genetic data, due to the way binary-to-decimal conversion works. For example, in the 6th position, a change of a gene from 0 to 1, will increase the value of the encoded number of the genetic data by 32. As the position increases, a change will either decrease or increase the value of the number by 2^{n-1} .

4.1 Experiment 5

Parameter	Value
Bit String Length	30
Population Size	100
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	1% (i.e. 1.0/N)
Crossover	True
Generations	100
Target Number	135300

When incorporating a binary fitness function into an evolutionary algorithm and introducing elitism with a rate of 1, we can anticipate a unique impact on the evolving population. Elitism, in the context of genetic algorithms, involves preserving a certain percentage of the best-performing individuals from one generation to the next. Specifically, an elitism rate of 1 means that the entire best-performing chromosome is carried over unchanged to the next generation.

In the case of a binary fitness function that evaluates the decimal representation of binary genes, this form of elitism essentially preserves the fittest



Figure 5: Simulation of Experiment 5

chromosome in its entirety. Since elitism retains the best individuals without any modification, the decimal value encoded by these genes remains unchanged across generations. As a result, the binary representation of the decimal value that is closest to the target will persist consistently in the evolving population. The overall fitness of the population, as measured by the binary fitness function, is likely to stabilize or even increase slightly with each generation, given that the fittest individuals persist.

4.2 Experiment 6

Parameter	Value
Bit String Length	20
Population Size	100
Selection	Truncation Selection
Elitism	0% (i.e. No Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	False
Generations	200
Target Number	135300

When implementing an evolutionary algorithm with an elitism rate of 0, a contrasting dynamic emerges in the population’s evolutionary trajectory. Elitism at 0 signifies that no individuals from the current generation are carried over unchanged to the next, and only the offspring of the evolving population contribute to the subsequent generation. In the context of a binary fitness function where the fitness is determined by the proximity of the decimal value encoded by binary genes to a target value, this absence of elitism introduces a notable level of volatility, as displayed by the spikes. The spikes, which are not typically seen in the normal fitness functions, are because of the way binary works. Even a single bit changing from 1 to 0 will dramatically affect the overall value encoded by the genetic data, especially if its position is greater than the 1st

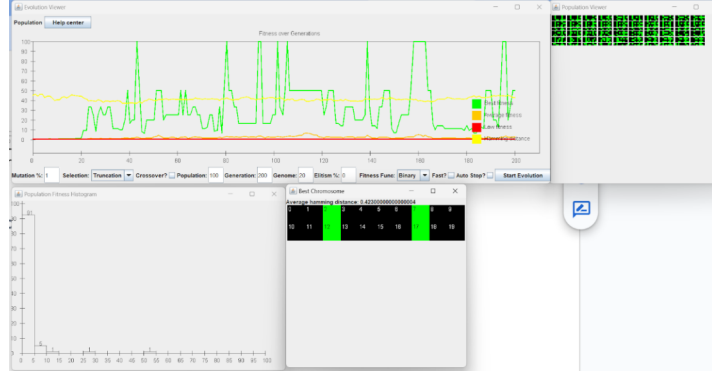


Figure 6: Simulation of Experiment 6

digit.

4.3 Experiment 7

Parameter	Value
Bit String Length	30
Population Size	100
Selection	Truncation Selection
Elitism	0% (i.e. No Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	False
Generations	100
Target Number	671613953

We also experimented with the given target number, 671613953. This resulted in a thirty-digit bit string, which produced the following simulation results.

Trial	1	2	3	4	5	6	7	8	9	10
Generation	29	73	N/A	N/A	N/A	N/A	44	N/A	N/A	N/A

We believe that this result is caused by the fact that once the population size is large enough, mutated bits before a certain position no longer matter. For example, at the 30th position, the value that bit contributes is 2^{29} . Previous bits will also contribute as well, but it is easy for the fitness function to be imprecise and select the wrong bits for example at the 29th position in the bit string, which is supposed to be 0. Given the enabling of elitism, this always preserves the ‘best’ chromosome, causing the fitness function to get stuck there. However, enabling crossover can introduce diversity as well as the ability to recombine more to get better choices.

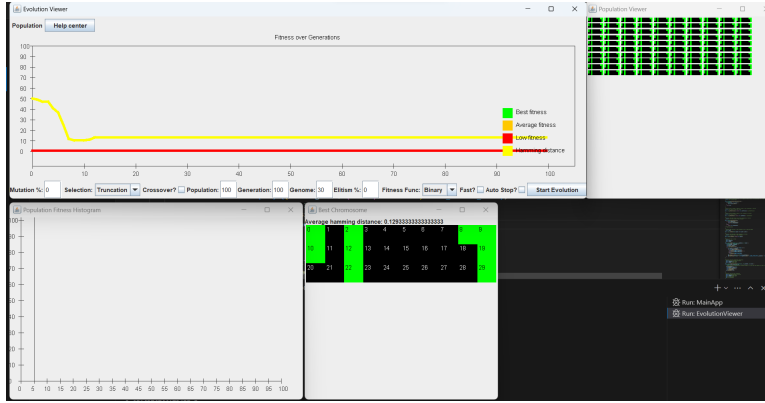


Figure 7: Simulation of Experiment 7

4.4 Experiment 8

Parameter	Value
Bit String Length	30
Population Size	100
Selection	Truncation Selection
Elitism	1% (i.e. Best Chromosome Cloned)
Mutation Rate	1% (i.e. $1.0/N$)
Crossover	True
Generations	100
Target Number	671613953

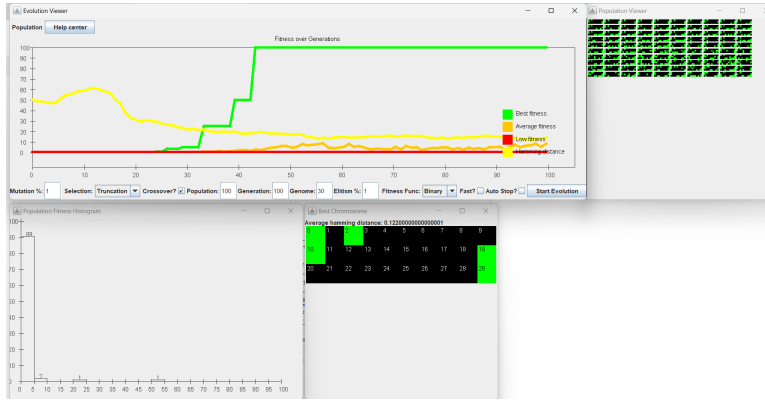


Figure 8: Simulation of Experiment 8

Trial	1	2	3	4	5	6	7	8	9	10
Generation	62	N/A	93	48	N/A	57	53	N/A	N/A	45

As demonstrated in the table above, the results align with our expectations, as a result of the introduction of diversity through crossover, potentially producing better chromosomes. Furthermore, elitism preserves the best chromosomes in case the next generation does not produce the most optimal either.

Nonetheless, in order to avoid failed trials, we believe that larger decimal numbers such as the current one will require a higher number of generations and a larger population to achieve the desired fitness score, i.e. a bit string that produces the right number when changed from binary to decimal. It is likely that enabling crossover and enabling elitism will assist in this as well.

5 Conclusion

In our genetic algorithm project, we simulated the evolution of chromosomes with the goal of attaining the optimal state—a condition where all gene bits are set to 1. Throughout this exploration, we explored the fundamental components of our genetic algorithm, focusing on chromosome representation, the integration of mutation mechanisms, and the optimization objective of achieving perfect chromosomes.

The heart of our genetic algorithm lies in the chromosomes. Our simulation sought to evolve these chromosomes towards the 'perfect' state, utilizing mutation to introduce diversity and simulate the evolutionary process. The optimization objective was to continually assess fitness and select the fittest individuals for reproduction.

In conclusion, our genetic algorithm project not only provided a platform for exploring the topic of evolution. The choices of chromosome representation, mutation methods, and elitism rates significantly influenced the trajectory of the evolving population. The project served as a valuable exploration of genetic algorithms, emphasizing the importance of achieving desired outcomes in the quest for optimal solutions.

References

- [1] Hinton, G.E., & Nowlan, S.J. (1996). How Learning Can Guide Evolution.
Complex Syst., 1.