INFO 6205

Program Structure & Algorithms

Summer 2021

Viral Evolution-Covid-19 Report

<u>Team Members:</u>

Heni Bhungalia (001544075)
Ravi Patel (001533801)
Sachin Dhumne (001066158)

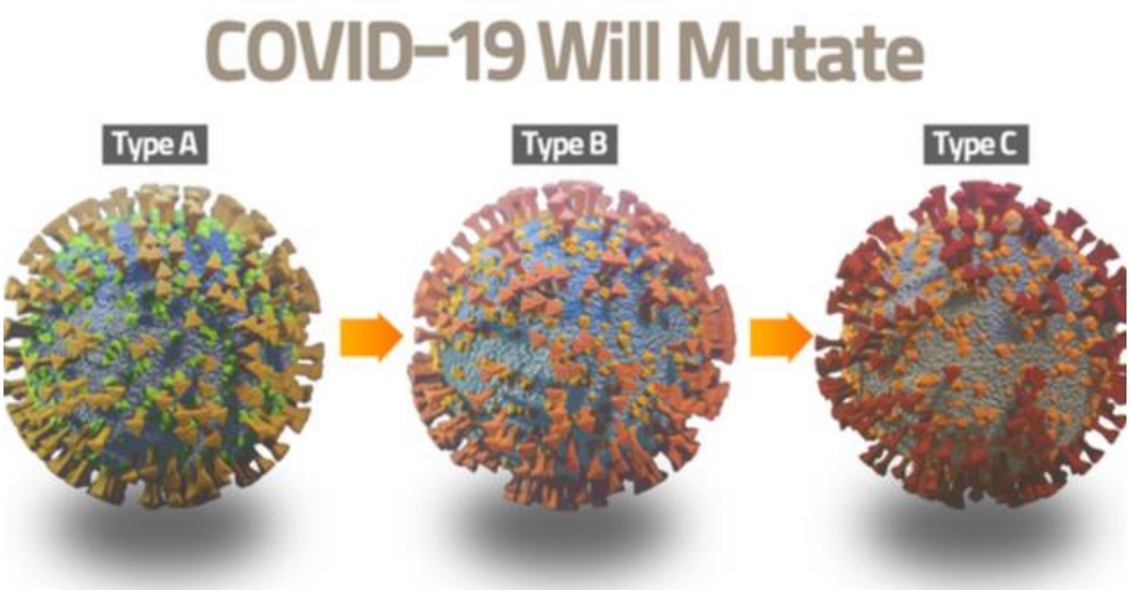


Table of Contents

INTRODUCTION

All viruses – including SARS-CoV-2, the virus that causes COVID-19 – evolve over time. When a virus replicates or makes copies of itself, it sometimes changes a little bit, which is normal for a virus. These changes are called “mutations”. A virus with one or more new mutations is referred to as a “variant” of the original virus

Most viral mutations have little to no impact on the virus’s ability to cause infections and disease. But depending on where the changes are located in the virus’s genetic material, they may affect a virus’s properties, such as transmission (for example, it may spread more or less easily) or severity (for example, it may cause more or less severe disease).

AIM OF THE PROJECT

Our objective is to study the evolution of variants of the SARS-CoV-2 virus having a genome consisting of almost 30,000 bases (i.e. A, C, G, or T) with 10 individual genes, on these potential hosts will be either (a) naive (i.e. not infected, not vaccinated), (b) previously infected, (c) vaccinated, (d) previously infected and vaccinated. Implementing the genetic algorithm to evolve new variants and simulating the virus evolution on the host population. And to simulate the virus transmission of SARS-CoV-2, the pathogen behind COVID-19. The simulations take into account different measures taken throughout the pandemic including no restrictions, lockdown, quarantine, social distancing, mask-wearing, testing, and vaccinations.

COMPLETE PROJECT DETAILS

- **Genetic Algorithm:** Genetic algorithms are designed to solve problems by using the same processes as in nature — they use a combination of selection, recombination, and mutation to evolve a solution to a problem. Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence.
  An algorithm starts with a set of solutions (represented by viruses) called population. Solutions from one population are taken and used to form a new population, as there is a chance that the new population will be better than the old ones are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. An algorithm starts with a set of solutions (represented by individuals) called population. Solutions from one population are taken and used to form a **new variant**, as there is a chance that the new population will be better than the old one.

- **Fitness Function:** We have measured the fitness defined for virus and host population to explain disease transmissibility and immunity, which indicates how non-infected people were infected by the different variants of viruses. Assuming fitness between (300 – 1000). We have 12 fitness functions for each kind of host and genome type.

- **Mutation:** In certain new offspring formed, some of their genes can be subjected to a **mutation** with a 1/3 of 30,000 probability. This implies that some of the genes in the genotype string can be flipped. Mutation occurs to maintain diversity within the population and prevent premature convergence.

COMPLEXITY AND DATA STRUCTURES

Given the usual choices (point mutation, one-point crossover, roulette wheel selection) a Genetic Algorithms complexity is **O(g(nm + nm + n))** with g the number of generations, n the population size and m the size of the individuals.

Fitness Function = 12 * O(n)

Therefore, the complexity is on the order of **O(g(nm)).**

HASH TABLE

We are using this data structure for saving the virus genotype as key and value as a list of 12 fitness functions. Thus, it becomes a data structure in which insertion and search operations are very fast irrespective of the size of the data. Hash Table uses an array as a storage medium and uses a hash technique to generate an index where an element is to be inserted or is to be located from.

Hash tables suffer from **O(n)** worst time complexity:

1. If too many elements were hashed into the same key: looking inside this key may take O(n) time.
2. Once a hash table has passed its load balance - it must rehash

However, it is said to be **O(1)** average and amortized case because:

3. It is very rare that many items will be hashed to the same key [if you chose a good hash function and you don't have too big a load balance.
4. The rehash operation, which is O(n), can at most happen after n/2 ops, which are all assumed O(1):
5. Thus, when you sum the average time per operation, you get:

(n*O(1) + O(n)) / n) = O(1)

IMPLEMENTATION (CHARTS/ALGORITHM)

In this project, we will be doing Viral Simulation using a Genetic Algorithm. We will be initializing the population, calculating fitness. We will be showing various evolution like Naïve, Infected, Recovered, Vaccinated, Died, and Delta Variant. Along with that we are also showing Graph1 for Infected, Recovered, Vaccinated, Died individuals. Graph2 and Graph3 will be showing Fitness generation for variants 1 and 2 respectively and Graph 3 is showing fitness generation for the Delta variant.

We have studied the modelling of the COVID-19 epidemic and the implementation of steps to combat the virus at the scale of the population. Our model consists of we have considered the following stages:

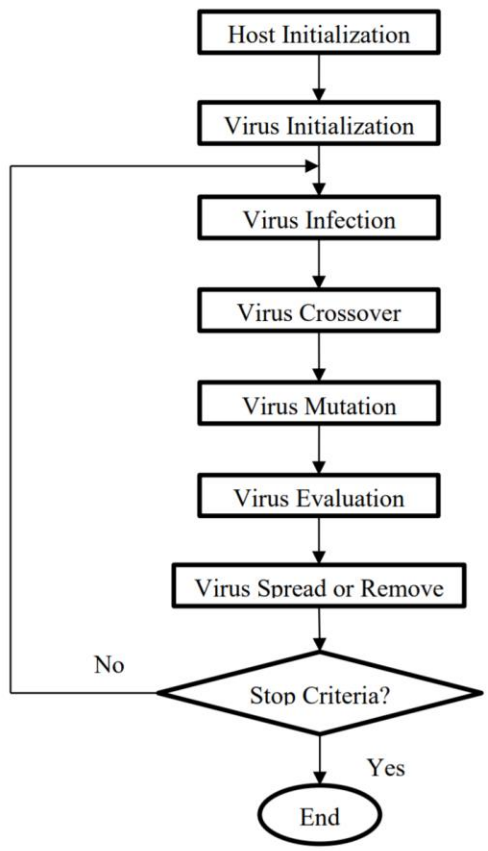


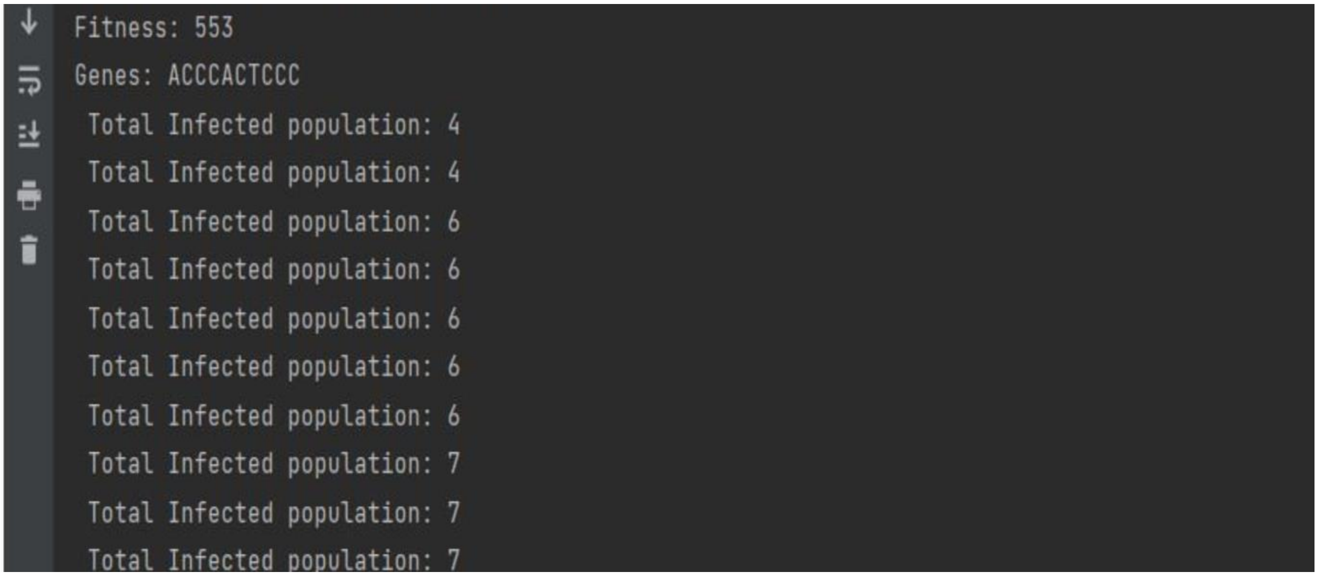
Figure 1 Flow Chart

1. GENERATION 1

ASSUMPTIONS:

We start with running a genetic algorithm by randomly creating a virus using 4 genotypes (A, C, G, T). Initializing the host population (1000) and randomly assigning them the fitness and 4 genes (A1, A2, B1, B2). Assume a minimum proximity distance of 10 units between two people to become infected. There is a high risk of disease transmission when an infected individual comes into contact with another within this radius of the infected, as seen in the graph.

CONSOLE OUTPUT:



```
Fitness: 553
Genes: ACCCACTCCC
 Total Infected population: 4
 Total Infected population: 4
 Total Infected population: 6
 Total Infected population: 6
 Total Infected population: 6
 Total Infected population: 6
 Total Infected population: 6
 Total Infected population: 7
 Total Infected population: 7
 Total Infected population: 7
```

Figure 2 Generation 1 Infection

FOR FIRST GENERATION:

Generation: 0 Fittest: 504
Generation: 1 Fittest: 485
Generation: 2 Fittest: 485
Generation: 3 Fittest: 485
Generation: 4 Fittest: 485
Generation: 5 Fittest: 485
Generation: 6 Fittest: 439
Generation: 7 Fittest: 410
Generation: 8 Fittest: 410
Generation: 63 Fittest: 486
Generation: 64 Fittest: 419
Generation: 65 Fittest: 419
Generation: 66 Fittest: 419
Generation: 67 Fittest: 419
Generation: 68 Fittest: 503
Generation: 69 Fittest: 503
…….
. …..
…….
. …..
…….
Generation: 93 Fittest: 419
Generation: 94 Fittest: 486
Generation: 95 Fittest: 534
Generation: 96 Fittest: 534
Generation: 97 Fittest: 553
Generation: 144 Fittest: 465
Generation: 145 Fittest: 503
Generation: 146 Fittest: 503
Generation: 147 Fittest: 532
Generation: 148 Fittest: 486
Generation: 149 Fittest: 486
Generation: 150 Fittest: 486
Generation: 151 Fittest: 553

Solution found in generation: 151
Fitness: 553
Genes: ACCCTCCCAC

CHART (First new Variant)

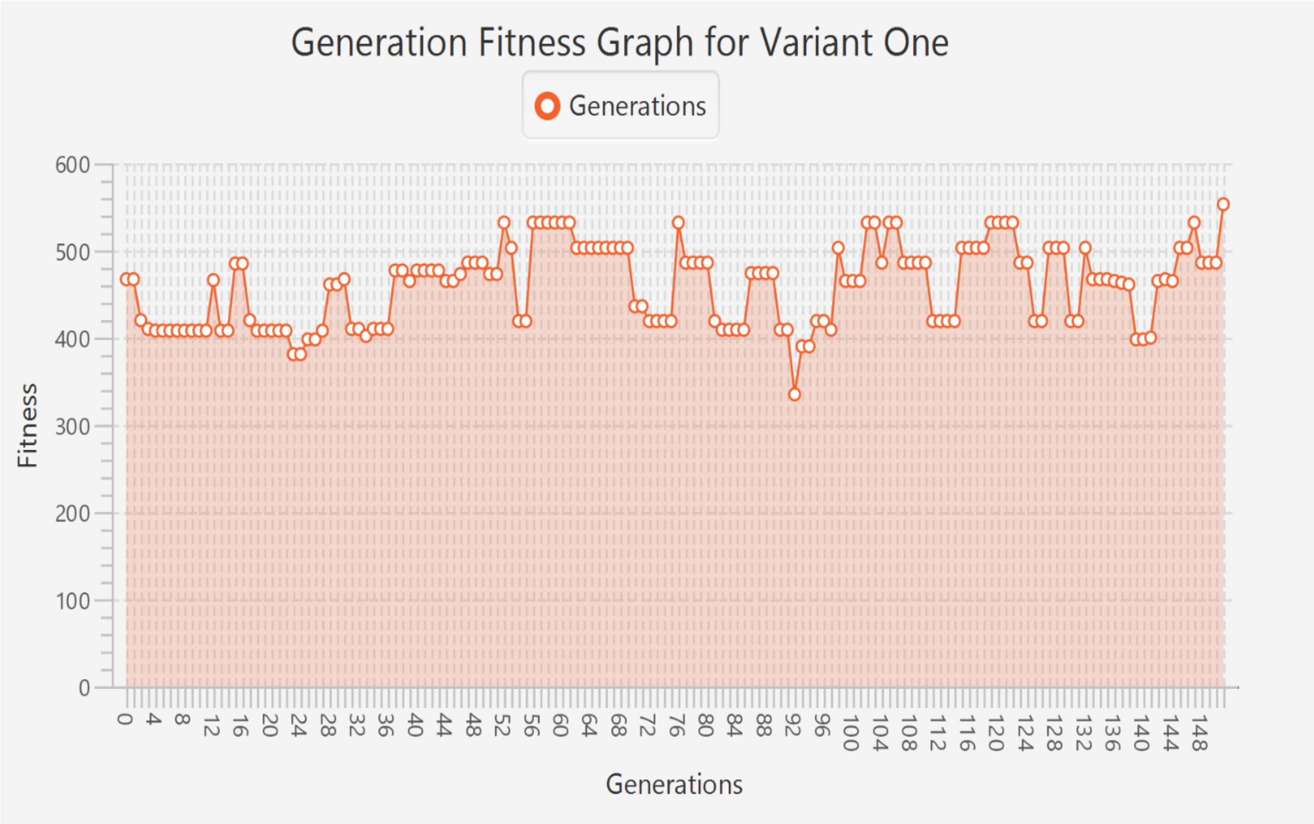


Figure 3 Number of Generation

SIMULATION

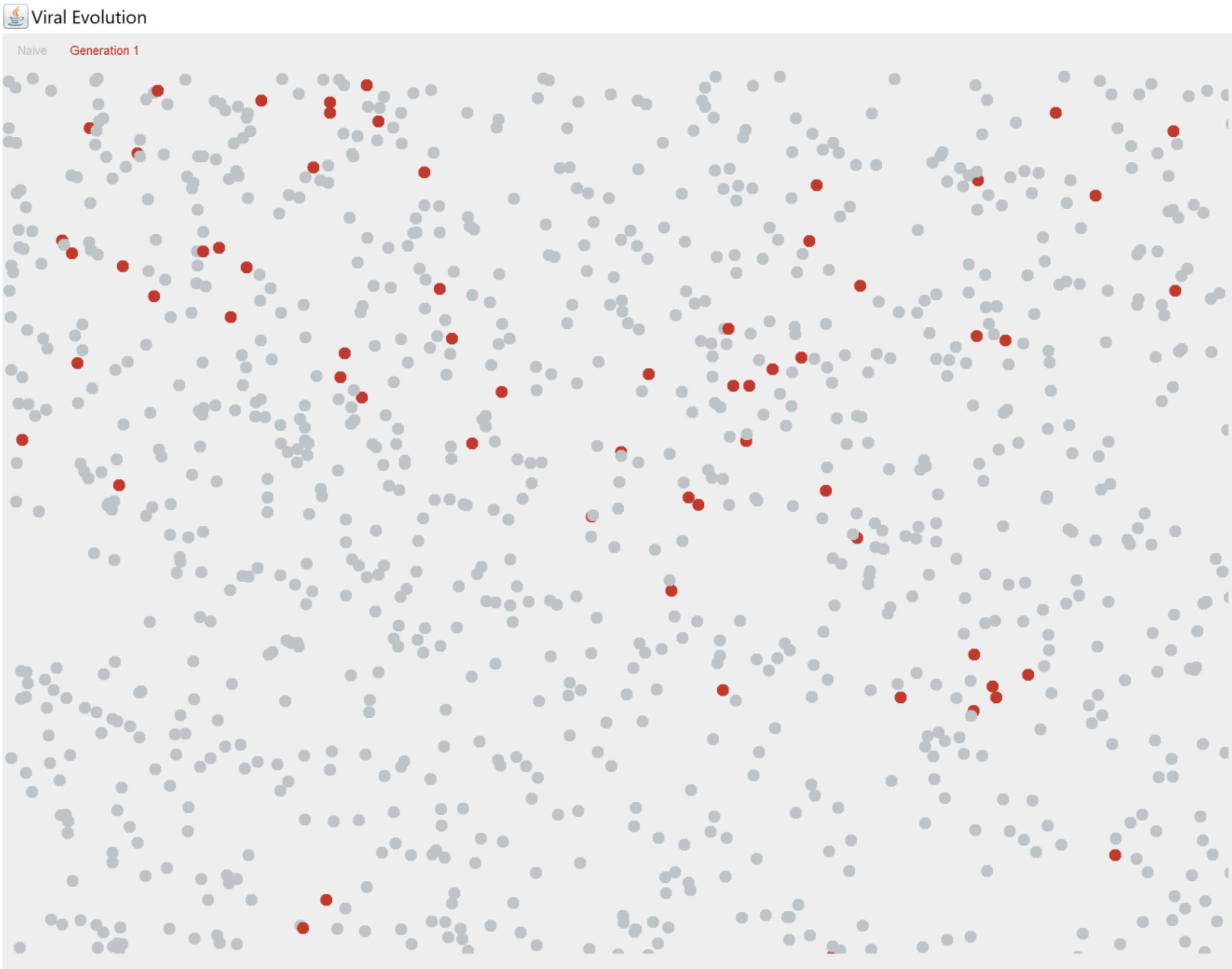


Figure 4 Generation One Infection

CONCLUSION GENERATION 1 VIRUS

The genetic algorithm found the first fittest virus on the 97th generation that can be visualized in the Generation Fitness graph of variant one. The host (naïve) population shown in grey and host population whose fitness is less than the virus will get affected and shown in brown. We can see in the simulation that 7-10 % host population is getting affected by generation 1.

2. GENERATION 2

ASSUMPTIONS:

We assume that 10% of the population would be affected by generation 1. After a span of 3 months, we call a genetic algorithm to perform mutation and evolve the second generation that is fitter than generation 1.

OUTPUT

CONSOLE OUTPUT

Generation: 0 Fittest: 485
Generation: 1 Fittest: 491
Generation: 2 Fittest: 485
Generation: 3 Fittest: 485
Generation: 4 Fittest: 463
Generation: 5 Fittest: 436
Generation: 6 Fittest: 430
Generation: 7 Fittest: 422
Generation: 8 Fittest: 422
Generation: 9 Fittest: 422
Generation: 10 Fittest: 422
Generation: 11 Fittest: 422
Generation: 12 Fittest: 422
Generation: 13 Fittest: 422
Generation: 14 Fittest: 418
Generation: 15 Fittest: 408
Generation: 16 Fittest: 408
Generation: 17 Fittest: 408
………
Generation: 108 Fittest: 532
Generation: 109 Fittest: 467
Generation: 110 Fittest: 467
Generation: 111 Fittest: 553

Solution found in generation: 111
Fitness: 553
Genes: CCCATCACCC

CHART

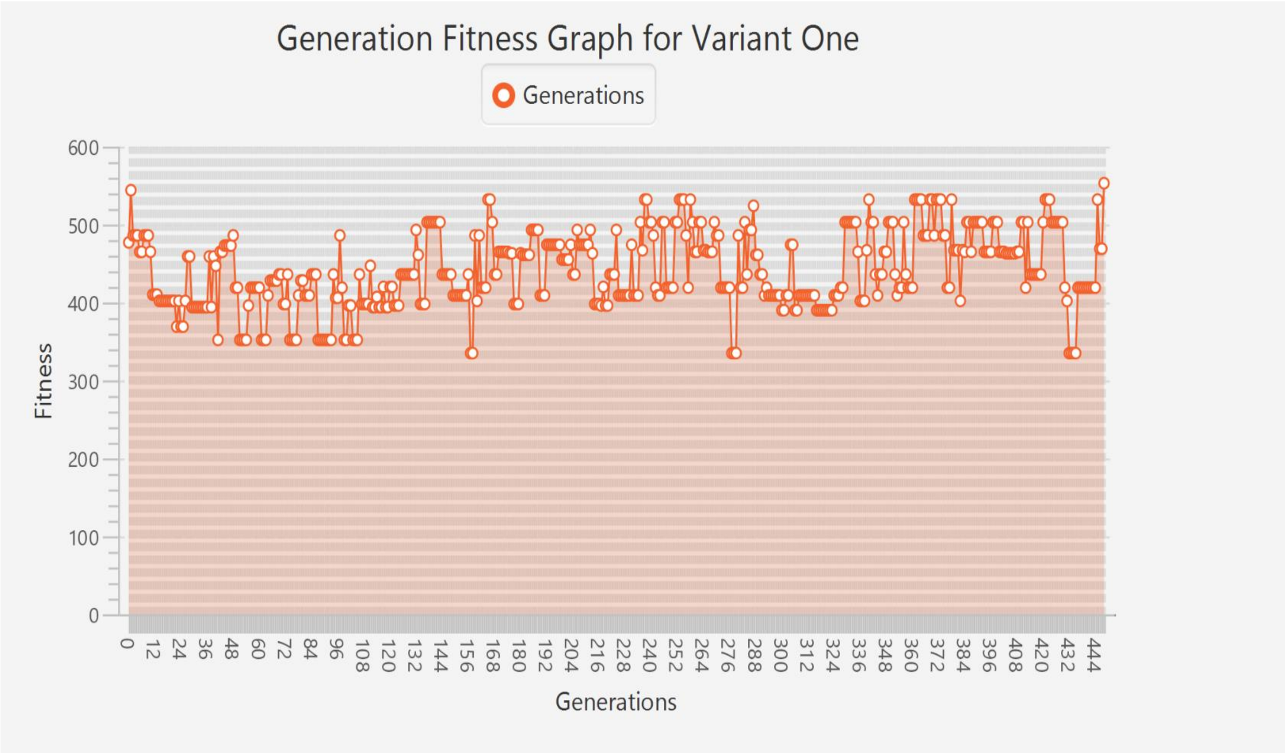


Figure 5 Number of Generation

SIMULATION

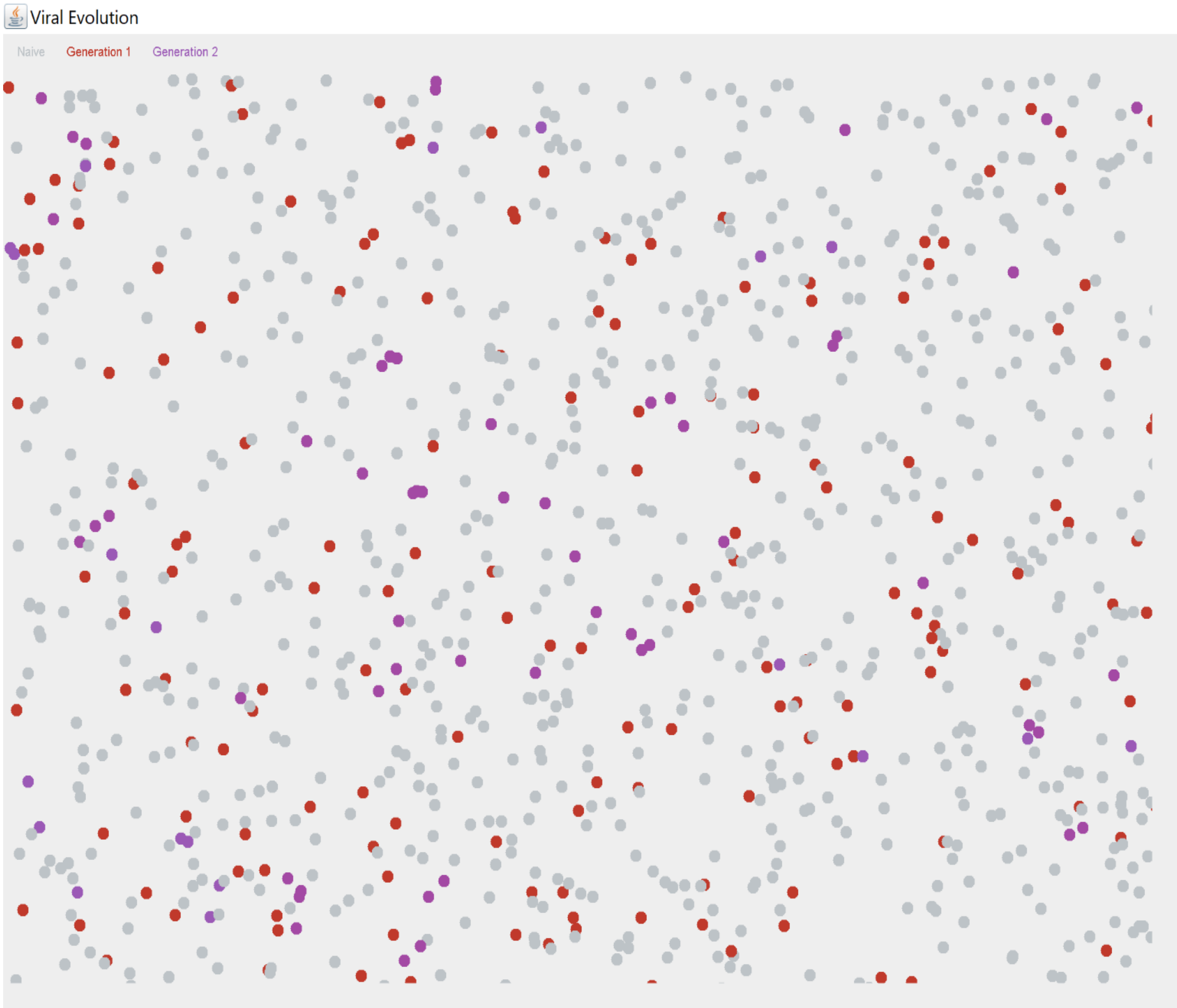


Figure 6 Generations Infections

CONCLUSION GENERATION 2

After generation 2 viral evolution it starts affecting the host population whose fitness is less than generation 2 woold get affected and start getting converted in purple. We can see in the chart that at 111th generation we found generation 2. In the simulation we can see about 15% of the host population is affected by 2nd covid variant.

3. RECOVERED

ASSUMPTIONS:

After getting affected by 2 different variants some of the host population who has good immunity would start recovering.

OUTPUT

CONSOLE OUTPUT

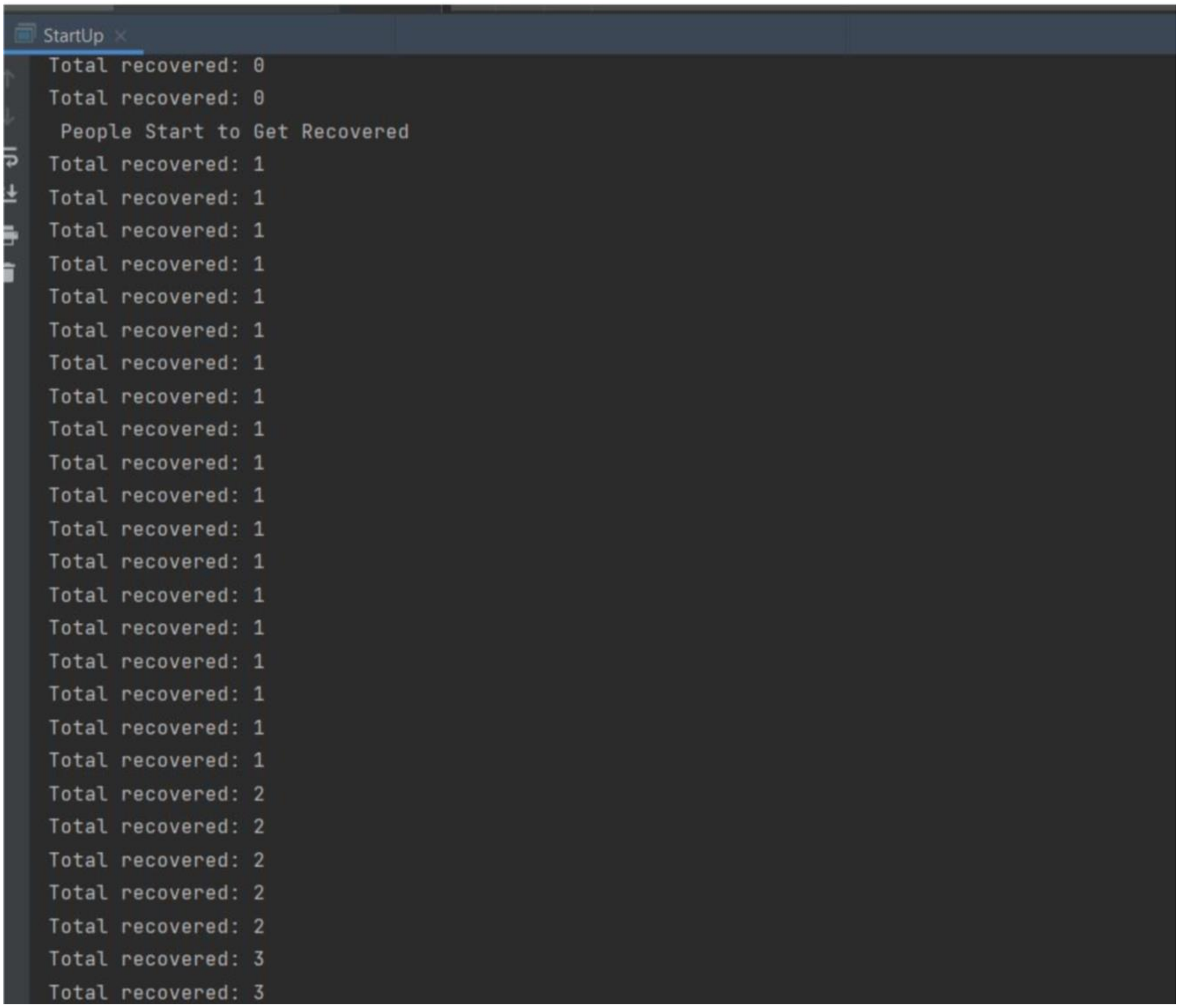


Figure 7 Recovered

CHART

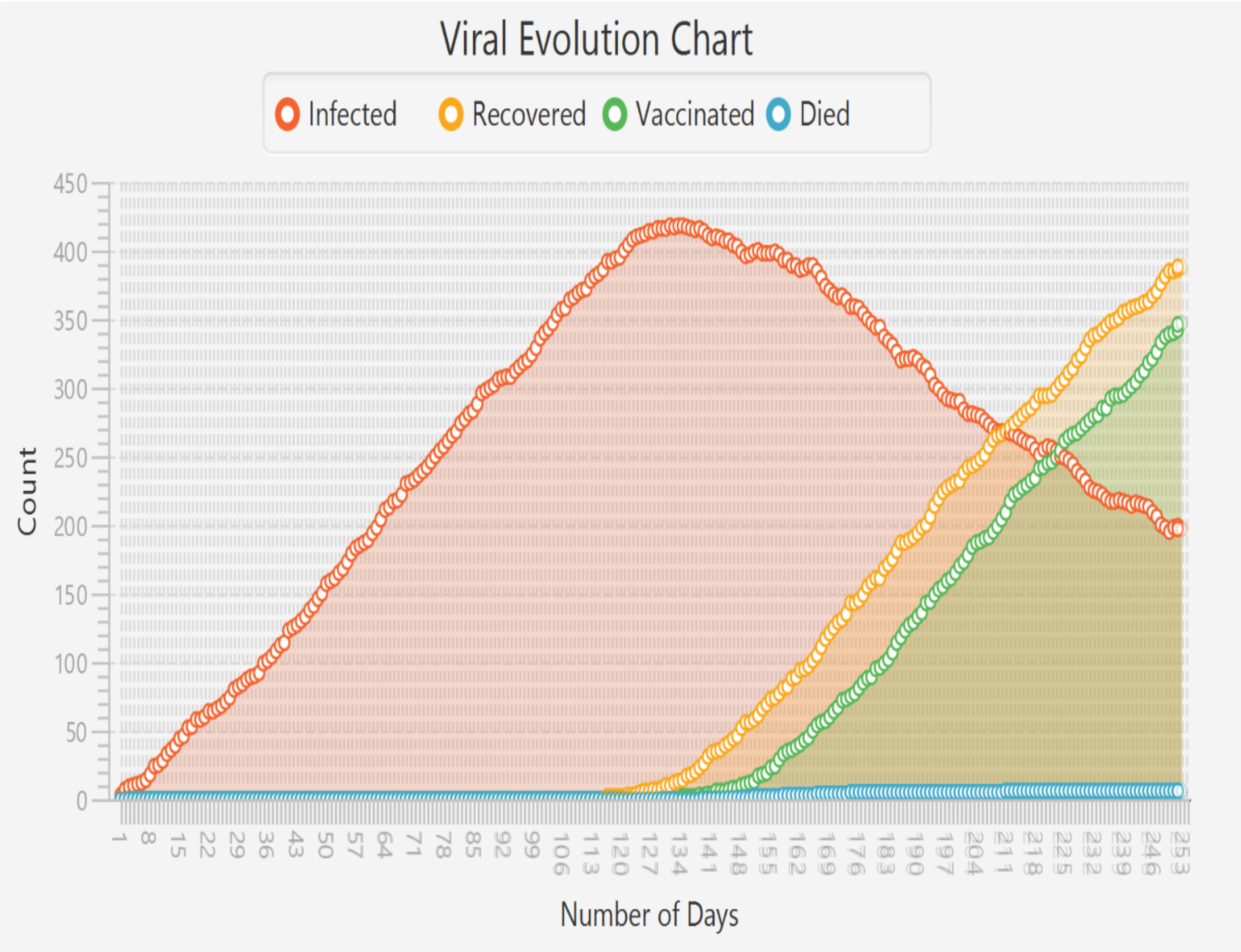


Figure 8 Viral Evolution

SIMULATION

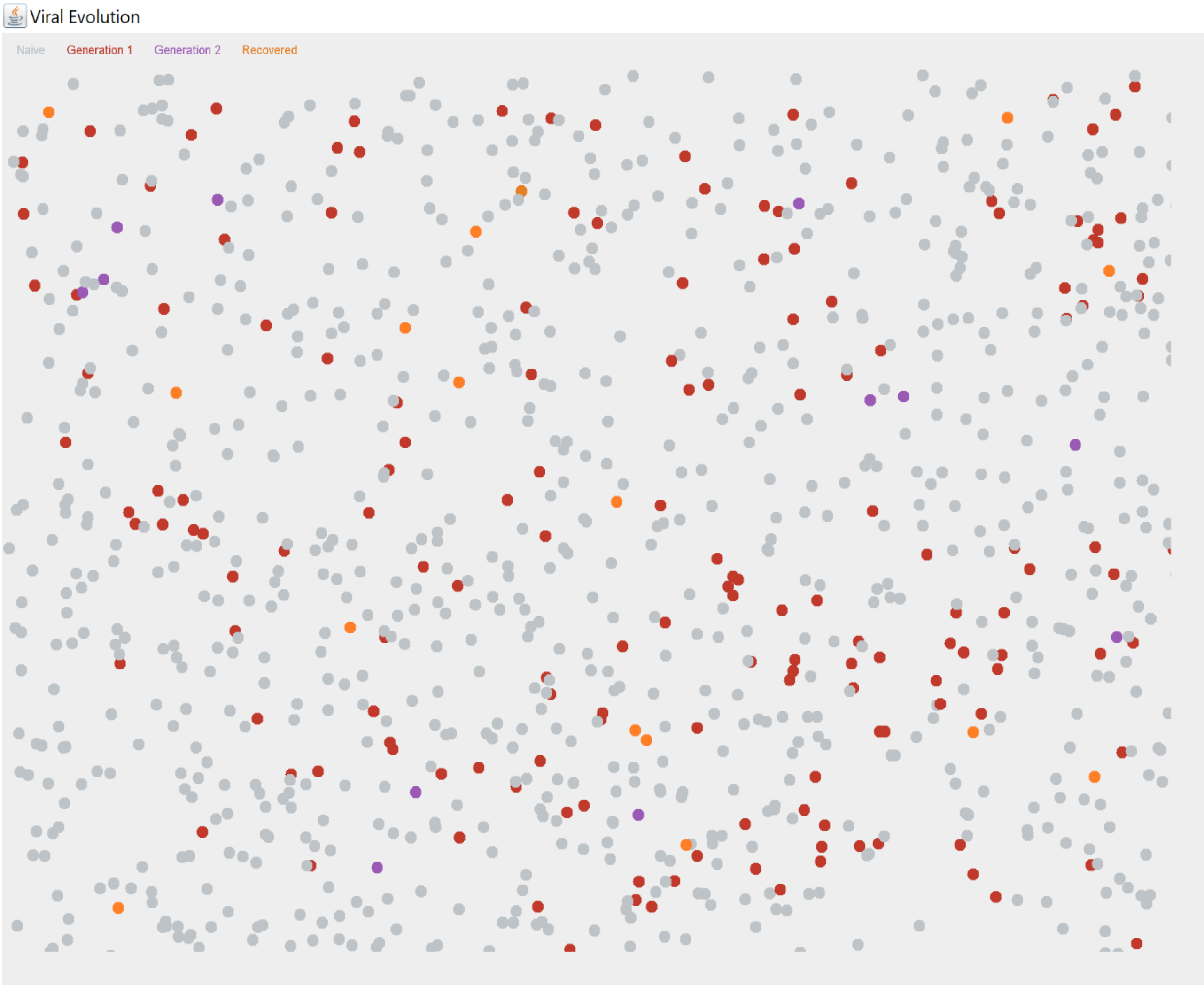


Figure 9 Recovered Population with Infected

RECOVERED CONCLUSION

Over the period of time, we can see some of the host population is getting recovered because of their good immunity. We can see in the chart that the recovery graph has started making linear progress and forming a curve. In the simulation we can see about 5% for of host population is getting recovered.

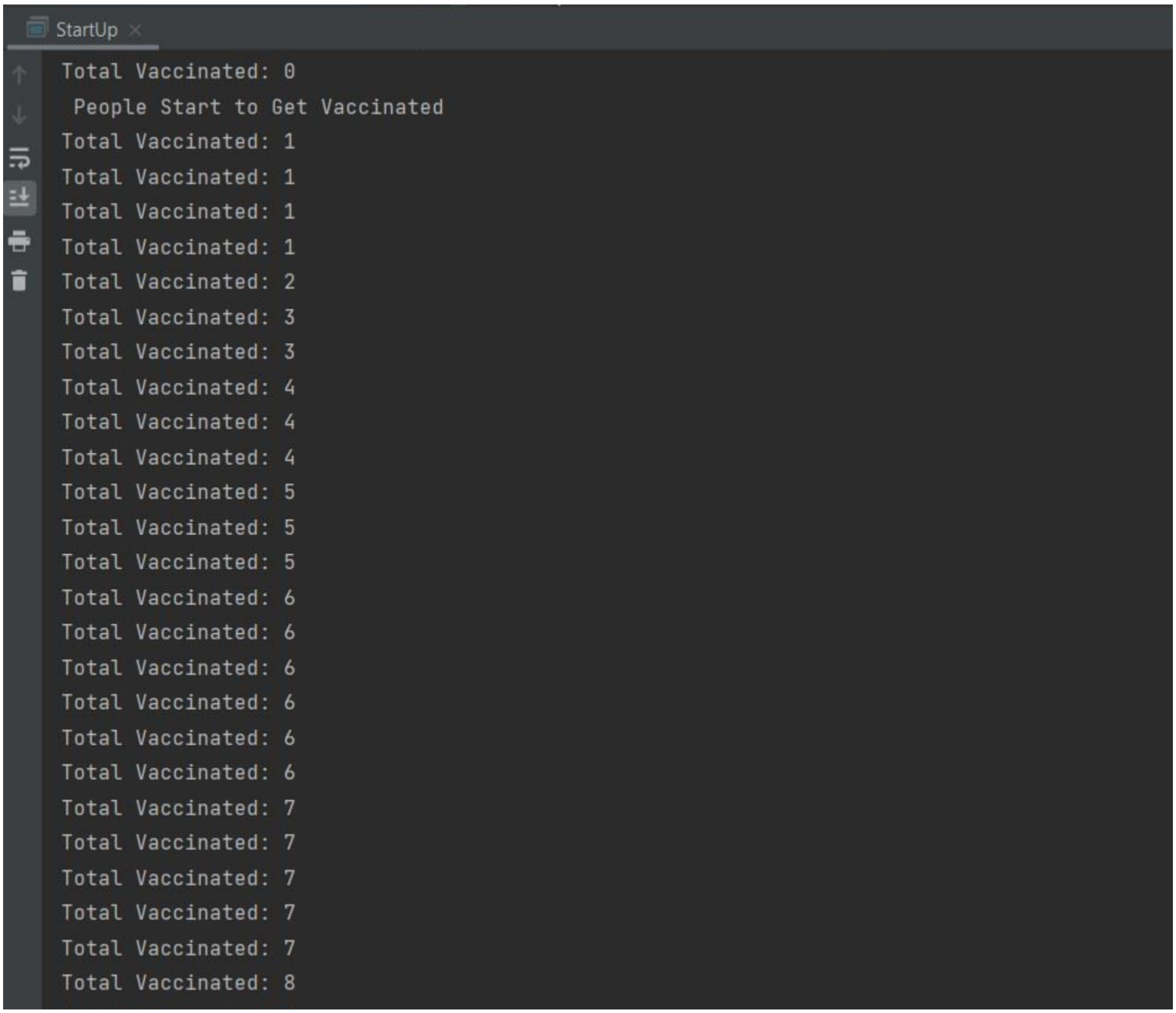
4. VACCINATED

ASSUMPTIONS:

After a span of one year (300 days), scientists and WHO were able to figure out the vaccine. Gradually people started getting the vaccine and they are becoming immune to Covid-19 by boosting their immunity and increasing their fitness above the virus.

OUTPUT

CONSOLE OUTPUT



```
StartUp
Total Vaccinated: 0
 People Start to Get Vaccinated
Total Vaccinated: 1
Total Vaccinated: 1
Total Vaccinated: 1
Total Vaccinated: 1
Total Vaccinated: 2
Total Vaccinated: 3
Total Vaccinated: 3
Total Vaccinated: 4
Total Vaccinated: 4
Total Vaccinated: 4
Total Vaccinated: 5
Total Vaccinated: 5
Total Vaccinated: 5
Total Vaccinated: 6
Total Vaccinated: 6
Total Vaccinated: 6
Total Vaccinated: 6
Total Vaccinated: 6
Total Vaccinated: 6
Total Vaccinated: 7
Total Vaccinated: 7
Total Vaccinated: 7
Total Vaccinated: 7
Total Vaccinated: 7
Total Vaccinated: 8
```

Figure 10 Vaccinated

CHART

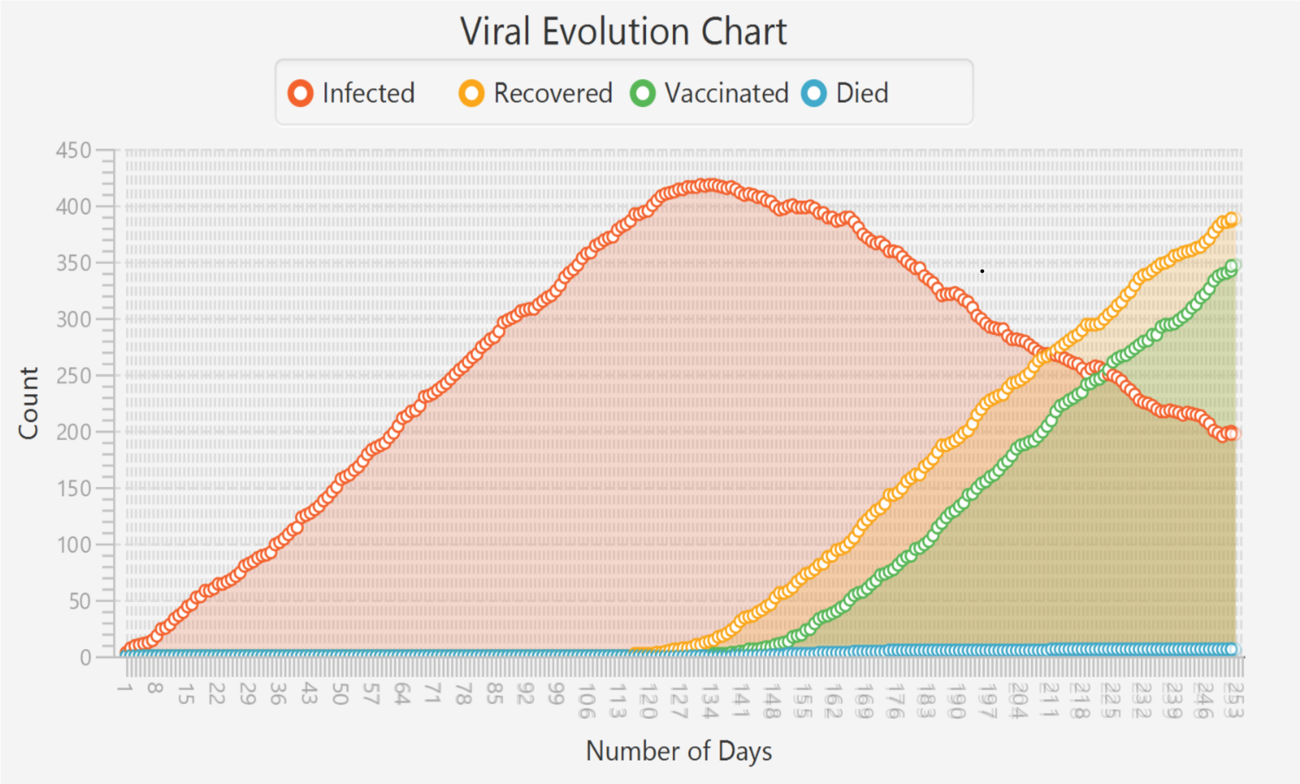


Figure 11 Viral Evolution

SIMULATION

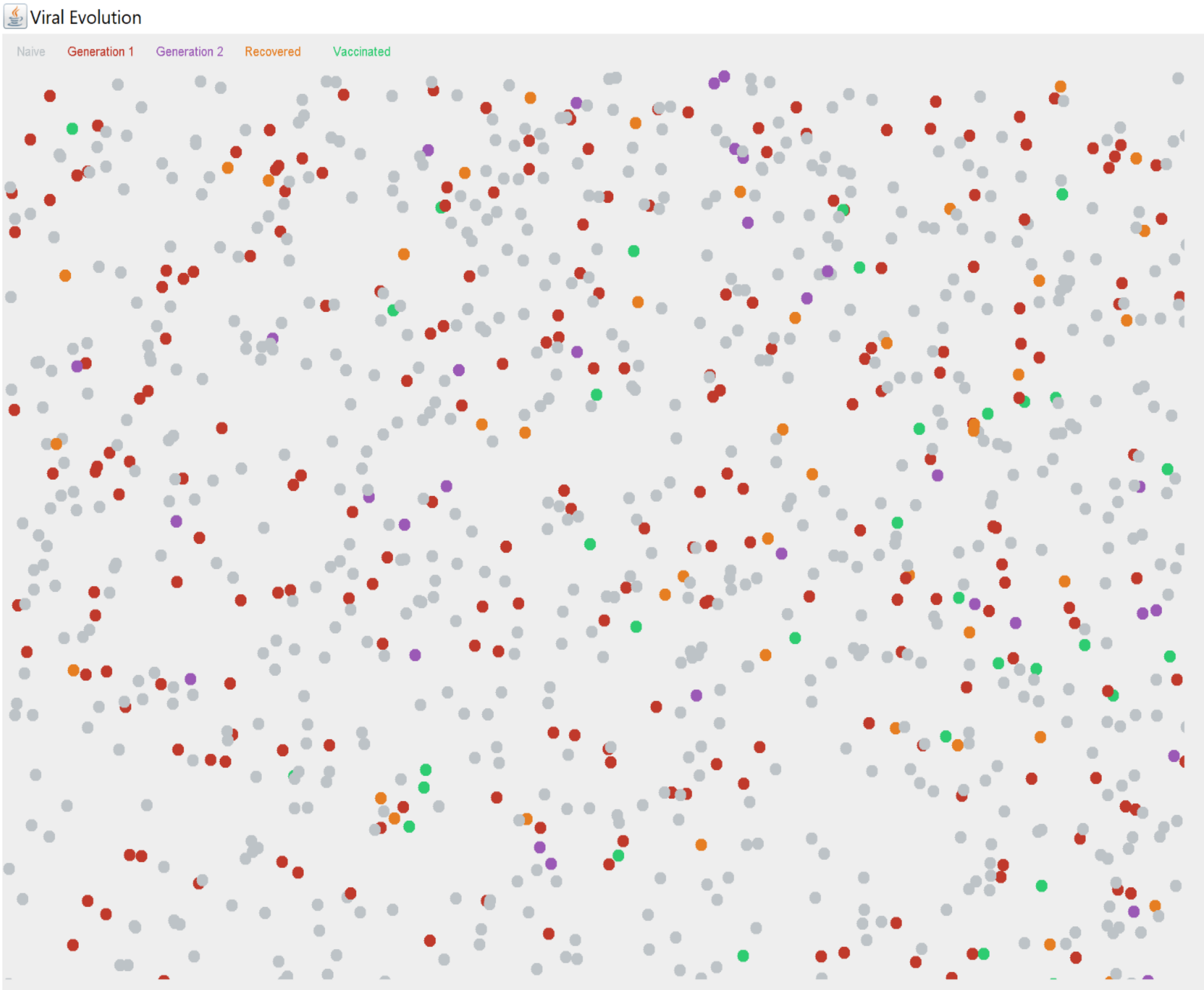


Figure 12 Vaccinated, Recovered with Infected

VACCINE CONCLUSION

With some of the host population getting recovered, and by start giving the vaccine to host population we can see that people are becoming more immune. In the chart we can see that with recovered vaccination is also improving gradually and making a linear growth. In simulation, we can see that brown and purple (infected host) were getting converted to orange(recovered) and green (vaccinated).

5. DIED

ASSUMPTIONS:

After being affected by Covid-19, some the host population who were having less immunity because of some already suffering diseases were died.

OUTPUT

CONSOLE OUTPUT

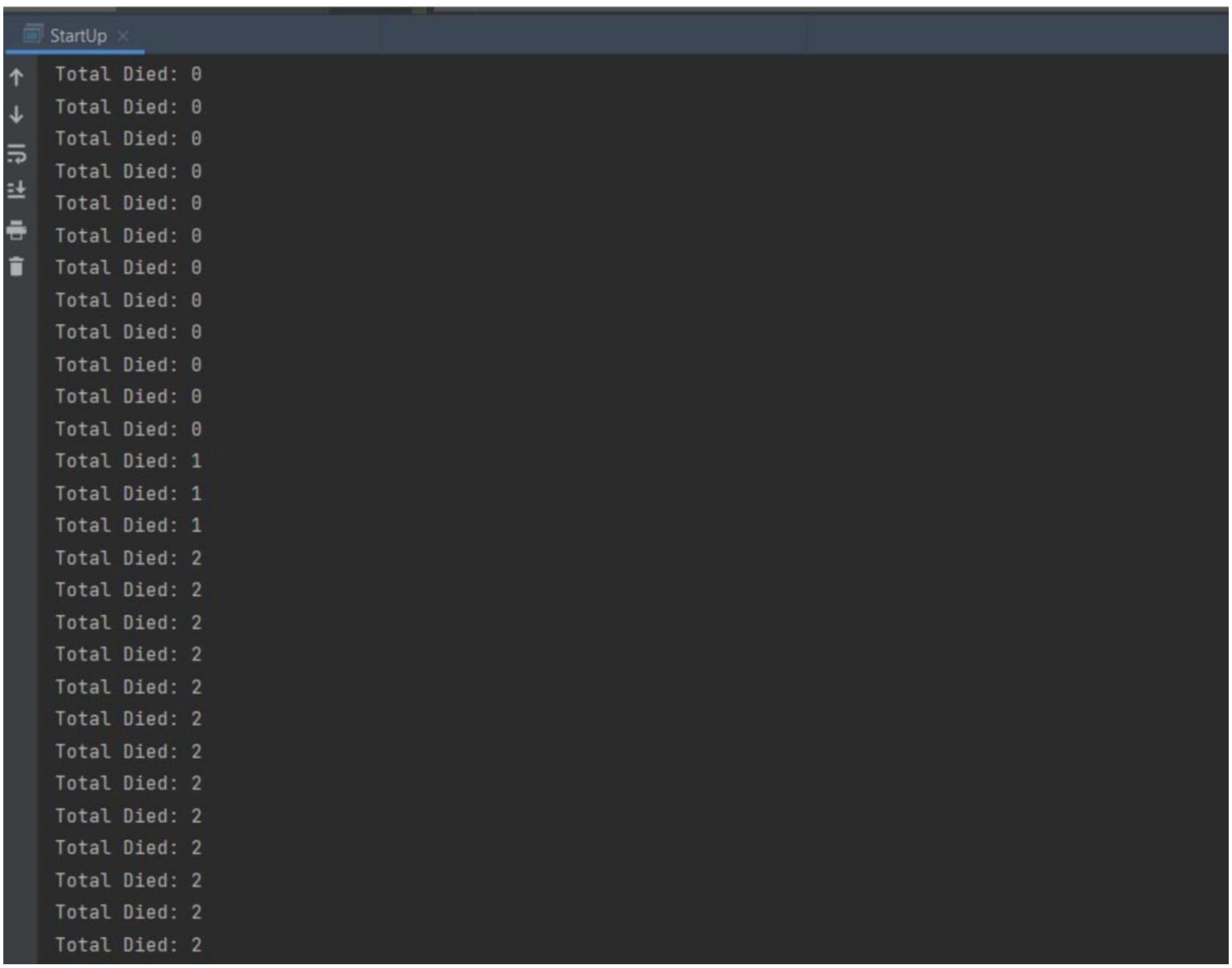


Figure 13 Died

CHART

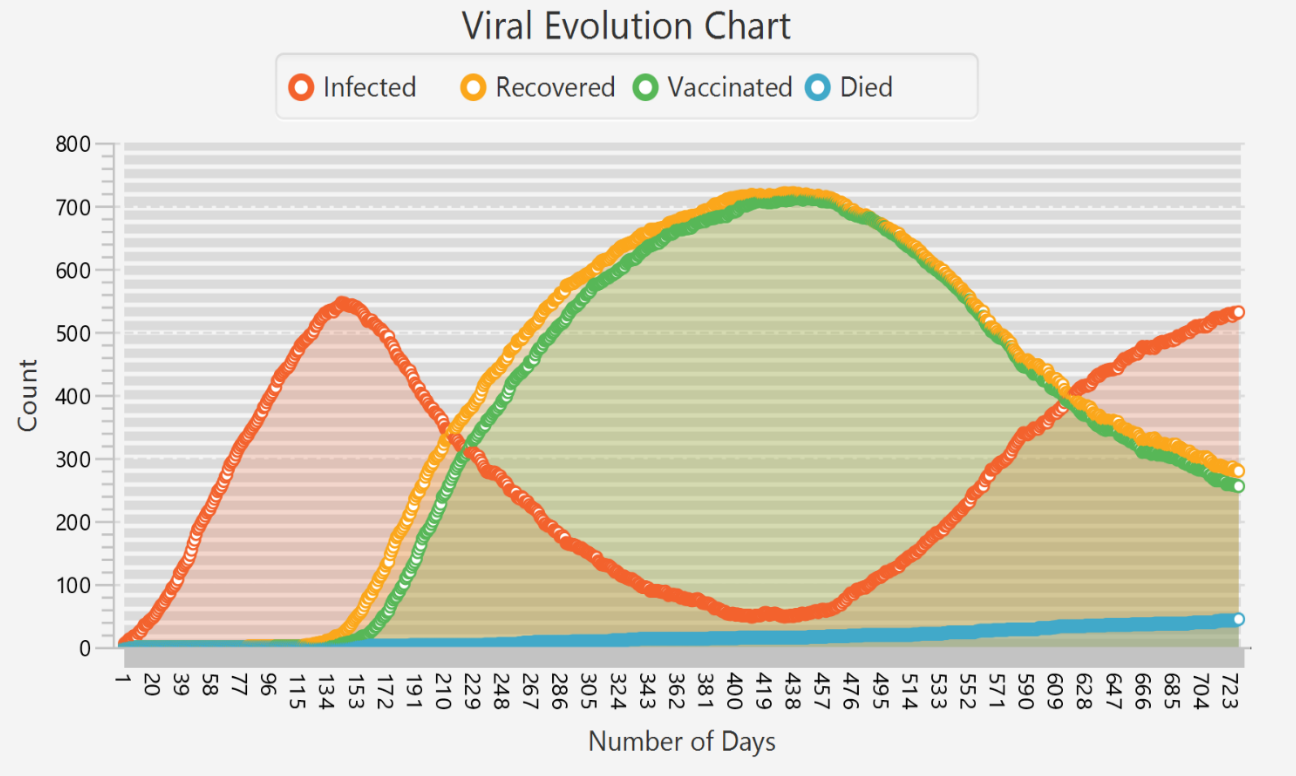


Figure 14 Viral Evolution

SIMULATION

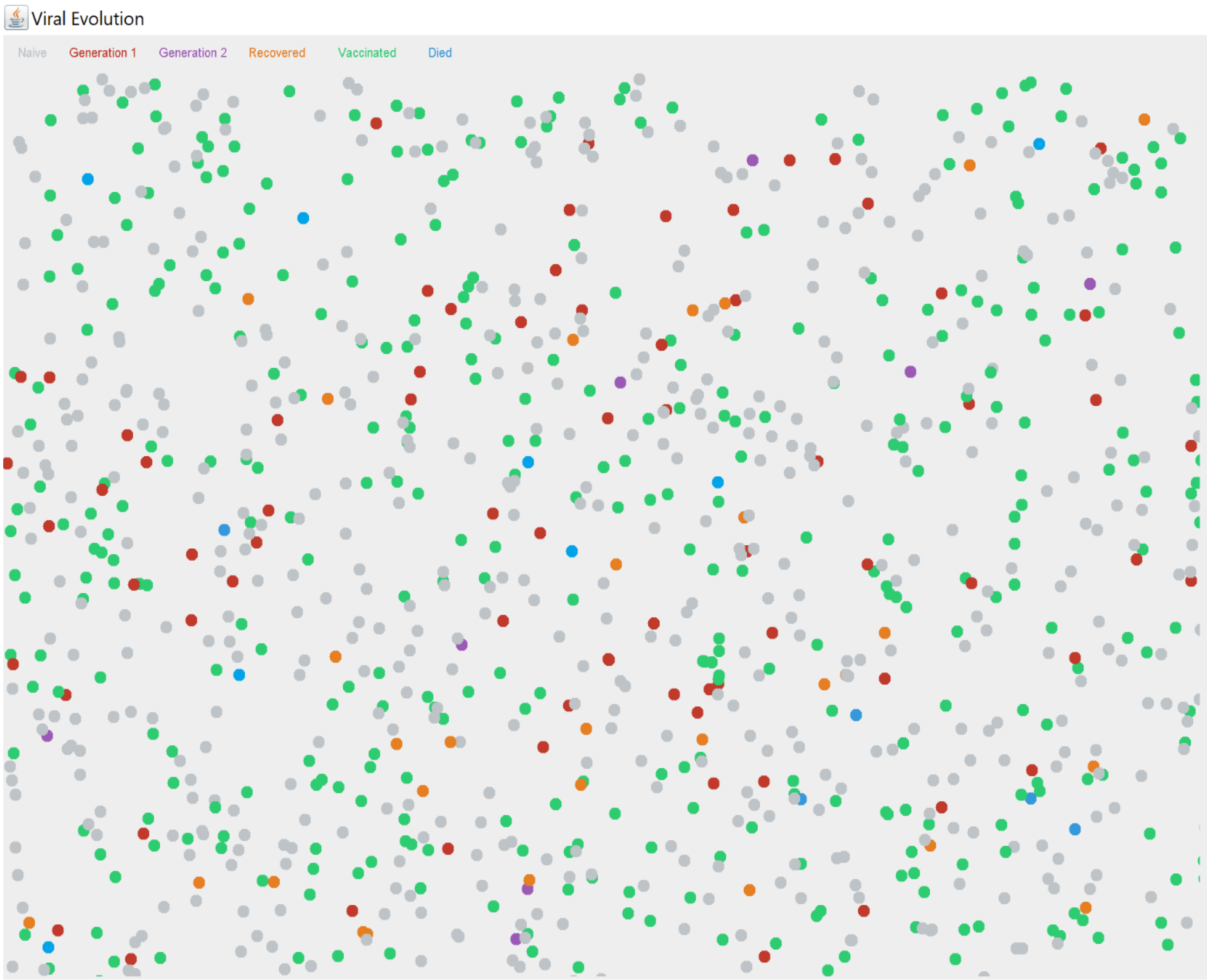


Figure 15 Died with Recovered, Infected, Vaccinated

DIED CONCLUSION

Two Covid-19 generations are killing 2% host population of low immunity. In the chart, we can see the blue graph has started making slow progress. In the simulation we can that the blue colour is not moving, and they are dead.

6. DELTA VARIANT

ASSUMPTIONS

After a year (around 400 days), there is the evolution of a new variant that has more fitness that could affect the vaccinated people as well. It has more mortality rate, more spreading rate.

OUTPUT

CONSOLE OUTPUT

Generation: 0 Fittest: 467
Generation: 1 Fittest: 428
Generation: 2 Fittest: 493
Generation: 3 Fittest: 493
Generation: 4 Fittest: 493
Generation: 5 Fittest: 493
Generation: 6 Fittest: 493
Generation: 7 Fittest: 493
Generation: 8 Fittest: 428
Generation: 9 Fittest: 428
Generation: 10 Fittest: 428
Generation: 11 Fittest: 428
Generation: 12 Fittest: 409
Generation: 13 Fittest: 474
Generation: 14 Fittest: 474
………..
Generation: 4038 Fittest: 534
Generation: 4039 Fittest: 534
Generation: 4040 Fittest: 534
Generation: 4041 Fittest: 534
Generation: 4042 Fittest: 534
Generation: 4043 Fittest: 534
Generation: 4044 Fittest: 601

Solution found in generation: 4044
Fitness: 601
Genes: ACCCCCCCCT

CHART

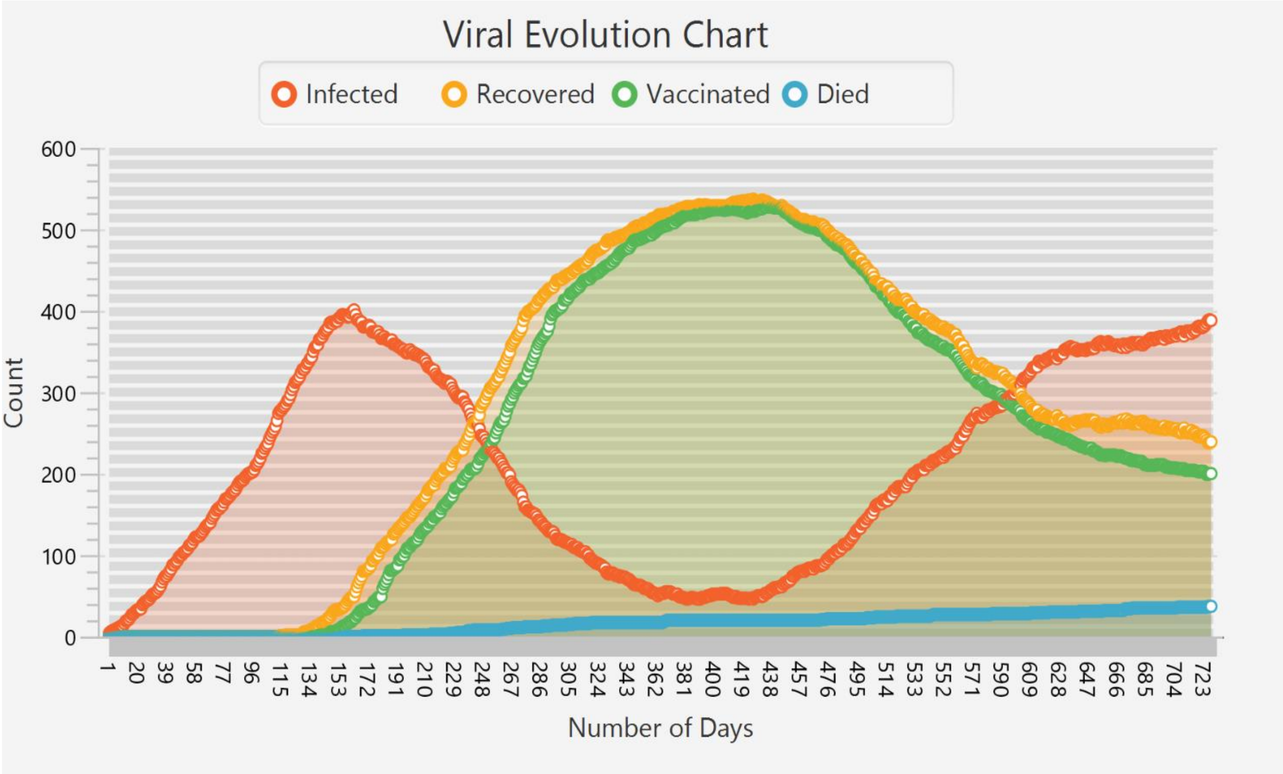


Figure 16 Viral Evolution

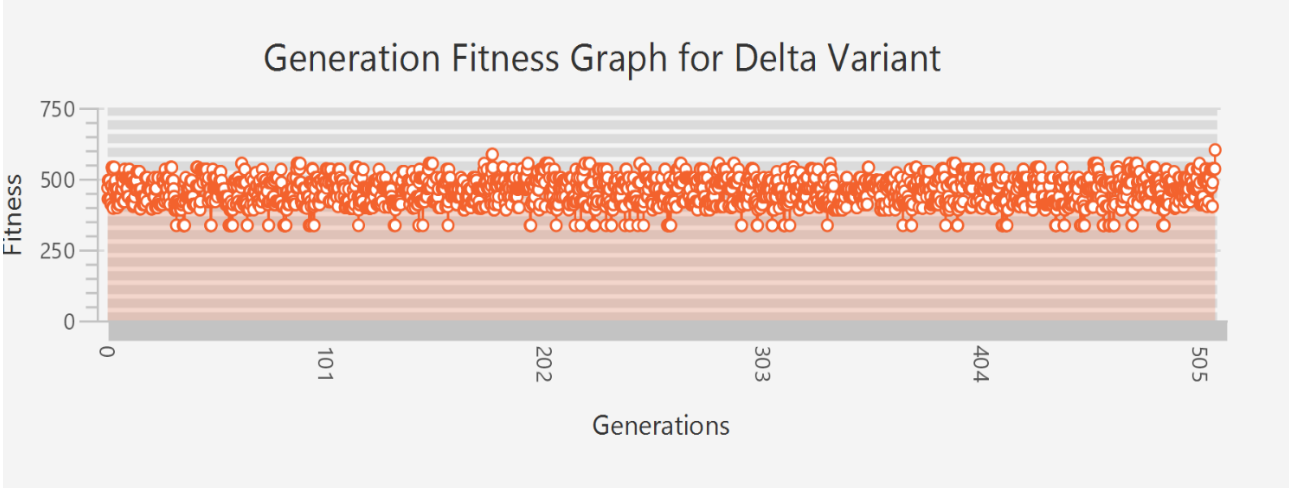


Figure 17 Number of Generation for Delta

SIMULATION

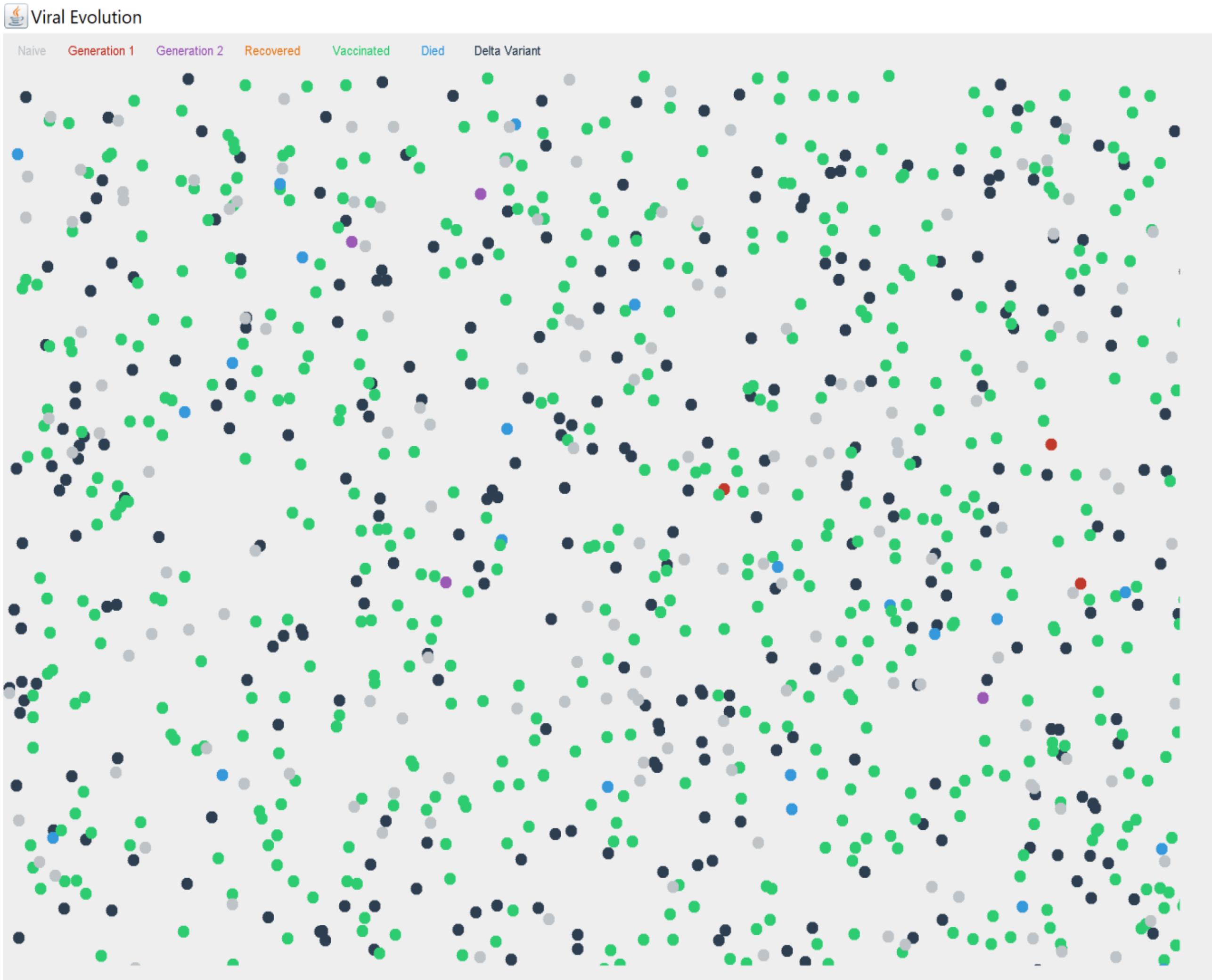
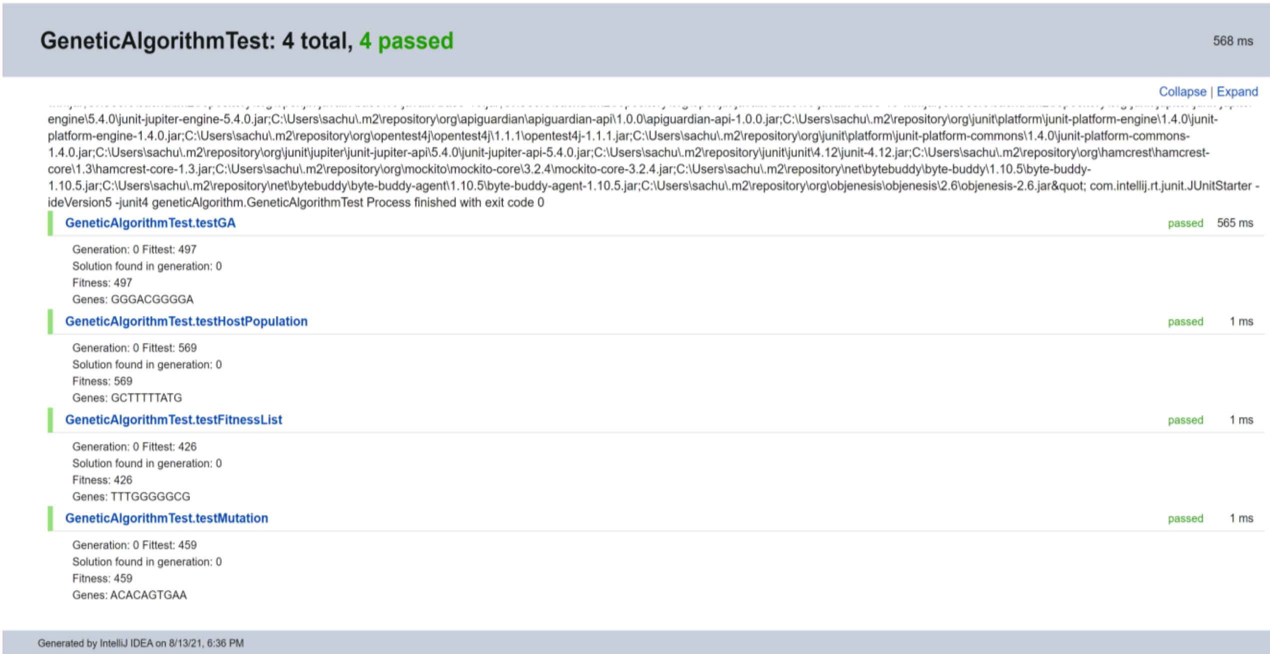


Figure 18 Delta Variant with Others

DELTA VARIANT CONCLUSION

Hence, we can see the delta variant of Covid-19 is becoming more dominant and causing infection to even vaccinated host population. We can see in the graph the infection rate is again going up and recovered/vaccinated is going down. In the simulation, we can see that the black colour (delta variant) is affecting all kinds of genotypes of the host population including vaccinated and recovered.

UNIT TESTS CASES



GeneticAlgorithmTest: 4 total, 4 passed 568 ms

Collapse | Expand

```
engine\5.4.0\junit-jupiter-engine-5.4.0.jar;C:\Users\sachu\.m2\repository\org\apiguardian\apiguardian-api\1.0.0\apiguardian-api-1.0.0.jar;C:\Users\sachu\.m2\repository\org\junit\platform\junit-platform-engine\1.4.0\junit-
platform-engine-1.4.0.jar;C:\Users\sachu\.m2\repository\org\opentest4j\opentest4j\1.1.1\opentest4j-1.1.1.jar;C:\Users\sachu\.m2\repository\org\junit\platform\junit-platform-commons\1.4.0\junit-platform-commons-
1.4.0.jar;C:\Users\sachu\.m2\repository\org\junit\jupiter\junit-jupiter-api\5.4.0\junit-jupiter-api-5.4.0.jar;C:\Users\sachu\.m2\repository\junit\junit\4.12\junit-4.12.jar;C:\Users\sachu\.m2\repository\org\hamcrest\hamcrest-
core\1.3\hamcrest-core-1.3.jar;C:\Users\sachu\.m2\repository\org\mockito\mockito-core\3.2.4\mockito-core-3.2.4.jar;C:\Users\sachu\.m2\repository\net\bytebuddy\byte-buddy\1.10.5\byte-buddy-
1.10.5.jar;C:\Users\sachu\.m2\repository\net\bytebuddy\byte-buddy-agent\1.10.5\byte-buddy-agent-1.10.5.jar;C:\Users\sachu\.m2\repository\org\objenesis\objenesis\2.6\objenesis-2.6.jar&quot; com.intellij.rt.junit.JUnitStarter -
ideVersion5 -junit4 geneticAlgorithm.GeneticAlgorithmTest Process finished with exit code 0
```

GeneticAlgorithmTest.testGA passed 565 ms

Generation: 0 Fittest: 497
Solution found in generation: 0
Fitness: 497
Genes: GGGACGGGGA

GeneticAlgorithmTest.testHostPopulation passed 1 ms

Generation: 0 Fittest: 569
Solution found in generation: 0
Fitness: 569
Genes: GCTTTTTATG

GeneticAlgorithmTest.testFitnessList passed 1 ms

Generation: 0 Fittest: 426
Solution found in generation: 0
Fitness: 426
Genes: TTTGGGGGCG

GeneticAlgorithmTest.testMutation passed 1 ms

Generation: 0 Fittest: 459
Solution found in generation: 0
Fitness: 459
Genes: ACACAGTGAA

Generated by IntelliJ IDEA on 8/13/21, 6:36 PM

Figure 19 Test Case 1

VirusPopulationTest: 6 total, 6 passed 14 ms

Collapse | Expand

```
&quot;C:\Program Files\Java\jdk1.8.0_231\bin\java.exe&quot; -ea -javaagent:C:\Users\sachu\.IntelliJIdea2019.3\system\testAgent\intellij-coverage-agent-1.0.508.jar=C:\Users\sachu\AppData\Local\Temp\coverageargs -
Didea.test.cyclic.buffer.size=1048576 &quot;-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.1\lib\idea_rt.jar=54432:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.1\bin&quot; -Dfile.encoding=UTF-8 -classpath
&quot;C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.1\lib\idea_rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.1\plugins\junit\lib\junit5-rt.jar;C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.1\plugins\junit\lib\junit-
rt.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\charsets.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\deploy.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\access-bridge-64.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\ext\cldrdata.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\dnsns.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\jaccess.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\jfxrt.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\ext\localedata.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\nashorn.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\sunec.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\ext\sunjce_provider.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\sunmscapi.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\ext\sunpkcs11.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\ext\zipfs.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\javaws.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\jce.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\jfr.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\jfxswt.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\jsse.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\management-agent.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\plugin.jar;C:\Program
Files\Java\jdk1.8.0_231\jre\lib\resources.jar;C:\Program Files\Java\jdk1.8.0_231\jre\lib\rt.jar;C:\SandBox\PSA\FinalProject\INFO6205-ViralEvolution\target\test-classes;C:\SandBox\PSA\FinalProject\INFO6205-
ViralEvolution\target\classes;C:\Users\sachu\.m2\repository\org\openjfx\javafx-controls\13\javafx-controls-13.jar;C:\Users\sachu\.m2\repository\org\openjfx\javafx-controls\13\javafx-controls-13-
win.jar;C:\Users\sachu\.m2\repository\org\openjfx\javafx-graphics\13\javafx-graphics-13.jar;C:\Users\sachu\.m2\repository\org\openjfx\javafx-graphics\13\javafx-graphics-13-
win.jar;C:\Users\sachu\.m2\repository\org\openjfx\javafx-base\13\javafx-base-13.jar;C:\Users\sachu\.m2\repository\org\openjfx\javafx-base\13\javafx-base-13-win.jar;C:\Users\sachu\.m2\repository\org\junit\jupiter\junit-jupiter-
engine\5.4.0\junit-jupiter-engine-5.4.0.jar;C:\Users\sachu\.m2\repository\org\apiguardian\apiguardian-api\1.0.0\apiguardian-api-1.0.0.jar;C:\Users\sachu\.m2\repository\org\junit\platform\junit-platform-engine\1.4.0\junit-platform-
engine-1.4.0.jar;C:\Users\sachu\.m2\repository\org\opentest4j\opentest4j\1.1.1\opentest4j-1.1.1.jar;C:\Users\sachu\.m2\repository\org\junit\platform\junit-platform-commons\1.4.0\junit-platform-commons-
1.4.0.jar;C:\Users\sachu\.m2\repository\org\junit\jupiter\junit-jupiter-api\5.4.0\junit-jupiter-api-5.4.0.jar;C:\Users\sachu\.m2\repository\junit\junit\4.12\junit-4.12.jar;C:\Users\sachu\.m2\repository\org\hamcrest\hamcrest-
core\1.3\hamcrest-core-1.3.jar;C:\Users\sachu\.m2\repository\org\mockito\mockito-core\3.2.4\mockito-core-3.2.4.jar;C:\Users\sachu\.m2\repository\net\bytebuddy\byte-buddy\1.10.5\byte-buddy-
1.10.5.jar;C:\Users\sachu\.m2\repository\net\bytebuddy\byte-buddy-agent\1.10.5\byte-buddy-agent-1.10.5.jar;C:\Users\sachu\.m2\repository\org\objenesis\objenesis\2.6\objenesis-2.6.jar&quot; com.intellij.rt.junit.JUnitStarter -
ideVersion5 -junit4 geneticAlgorithm.VirusPopulationTest ---- IntelliJ IDEA coverage runner ---- sampling ... include patterns: geneticAlgorithm\..* exclude patterns: Class transformation time: 0.0196642s for 428 classes or
4.5944392523364485E-5s per class Process finished with exit code 0
```

VirusPopulationTest.compareFitnessTest	passed	13 ms
VirusPopulationTest.testSecondFittest	passed	0 ms
VirusPopulationTest.testLeastFittest	passed	1 ms
VirusPopulationTest.calculateFitnessTest	passed	0 ms
VirusPopulationTest.testFittest	passed	0 ms
VirusPopulationTest.initializePopulationTest	passed	0 ms

Generated by IntelliJ IDEA on 8/13/21, 6:37 PM

Figure 20 Test Case 2

PopulationFrameTest: 5 total, 5 passed

435 ms

Collapse | Expand

PopulationFrameTest		435 ms
testHeightWidth()	passed	409 ms
totalDiedTest()	passed	15 ms
totalVaccinatedTest()	passed	2 ms
totalRecoveredTest()	passed	4 ms
totalInfectedTest()	passed	5 ms

Generated by IntelliJ IDEA on 8/13/21, 6:37 PM

Figure 21 Test Case 3

CONCLUSION

From the simulation, we concluded that SARS-CoV-2 virus is highly infected but then too its infection depends on host type fitness. We ran Genetic Algorithm to create a different virus variant and show how it spread in different host. We also notice vaccination play big role to help stop spread virus infection. Eventually after a while the virus increase its fitness after infecting the host creating a more powerful virus variant which we labeled as Delta variant, even vaccinated people can get infected by delta variant.

Genetic Algorithm

- ***Invariant***
  - A genetic algorithm is invariant with respect to a set of representations if it runs the same no matter which of the representations is used. We formalize this concept mathematically, showing that the representations generate a group that acts upon the search space. Invariant genetic operators are those that commute with this group action. We then consider the problem of characterizing crossover and mutation operators that have such invariance properties. In the case where the corresponding group action acts transitively on the search space, we provide a complete characterization, including high-level representation-independent algorithms implementing these operators.

- ***Termination***
  - A solution is found that satisfies minimum criteria
  - Fixed number of generations reached
  - Allocated budget (computation time/money) reached
  - The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
  - Manual inspection
  - Combinations of the above

REFERENCES

- https://www.ai.uga.edu/sites/default/files/inline-files/Weixin_Ling_Thesis_1116.pdf
- https://en.wikipedia.org/wiki/Genetic_algorithm
- https://genome.ucsc.edu/cgi-bin/hgTracks?db=wuhCor1&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=NC_045512v2%3A1%2D29903&hgsid=1133869119_bR5bzYzNJAGr0pkslyjYXazsVtAZ
- https://towardsdatascience.com/how-to-define-a-fitness-function-in-a-genetic-algorithm-be572b9ea3b4