PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON

SIMULATION OF MULTI PARAMETRIC NATURAL SELECTION

Name	SRN
S Nikhil Ram	PES1201801972
Vishnu R Dixit	PES1201801448
Anirudh S Ayya	PES1201800338

Branch & Section : Computer Science and Engineering, 4 - ‘B’

PROJECT EVALUATION

(For Official Use Only)

Sl.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor :

Signature of the Course Instructor :

Table of Contents

List of Figures 3

Abstract 4

I. Introduction 4

1.1 Natural Selection 4

1.2 Mathematical Simulation 5

1.2.1 Linear Algebra 5

1.2.2 Monte Carlo Method 5

II. Approach 6

2.1 Overview of Approach 6

2.2 Environment Development 6

2.3 Adaptive Landscape Framework 7

2.4 Evolutionary Stable Strategy (ESS) 7

III. Data Representation 8

3.1 Overview of Data Representation 8

3.1.1 Mini World 8

3.1.2 Epoch 8

3.1.3 Population 8

3.1.4 Resource 9

3.2 Population with the Mini World 9

3.2.1 Mini World Environment 9

3.2.2 Features 9

3.2.3 G-Matrix 9

3.2.5 Gauss-Random Favouritism 10

IV. Implementation 11

4.2 Types Of Representation 11

4.2.1 Representation-0 11

4.2.2 Representation-1 11

4.2.3 Representation-2 12

V. Graphs and Observation 13

5.1 Population Distribution Analysis 13

5.2 Population Distribution Analysis 13

VI. Conclusion 15

VII. Bibliography 16

List of Figures

I. **Introduction**

1.1 Natural Selection

Fig 1.1 - Natural Selection

II. **Approach**

2.2 Environment Development

Fig 2.1 - Determining PI-Monte Carlo plot

2.3 Adaptive Landscape Framework

Fig 2.2 - Phenotype space (2-D Adaptive Landscape)

III. **Data Representation**

3.2.3 G-Matrix

Fig 3.1 - 2 Trait G-matrix

3.2.5 Gauss-Random Favouritism

Fig 3.2 - Sampling From Random Generator

IV. **Implementation**

V. **Graphs and observation**

5.2 Population Distribution Analysis

Fig 5.1 - Generated Transformation Map (1000 generations)

Fig 5.1.1 - Origin Generation

Fig 5.1.2 - After 1000 Epochs

Fig 5.2 - Population vs Time Curve (I)

Fig 5.3 - Population vs Time Curve (II)

VI. **Conclusion**

SIMULATION OF MULTI PARAMETRIC NATURAL SELECTION

Abstract

Natural Selection is the qualifier on the basis of which the intricate process of Evolution has proceeded through and set up the Adaptive Landscape over several thousand millennia. It requires a lot of resource and computation power to simulate this complex process accounting all the parameters involved. Hence we need to simplify these factors into a small subset of parameters and set up a mathematical environment to visualise the outcomes in a broad sense with the available resources.

I. Introduction

1.1 Natural Selection

Natural selection is one of the central mechanisms of evolutionary change and is the process responsible for the evolution of adaptive features. Natural selection is a non-random difference in reproductive output among replicating entities, often due indirectly to differences in survival in a particular environment, leading to an increase in the proportion of beneficial, heritable characteristics within a population from one generation to the next.[1a]

Natural selection gradually leads to the evolution of new characteristics by eliminating unfit traits. By itself, Natural Selection is incapable of producing new traits, and in fact, most forms of natural selection *deplete* genetic variation within populations. Evolution by natural selection is a two-step process. The first step involves the generation of new variation by mutation, whereas the second step determines which randomly generated variants will persist into the following generations. Most new mutations are neutral with respect to survival and reproduction and therefore are irrelevant in terms of natural selection. A small percentage of new mutations will turn out to have beneficial effects in a particular environment and will contribute to an elevated rate of reproduction by organisms possessing them. Even a very slight advantage is sufficient to cause new beneficial mutations to increase in proportion over the span of many generations.[1a]

NATURAL SELECTION

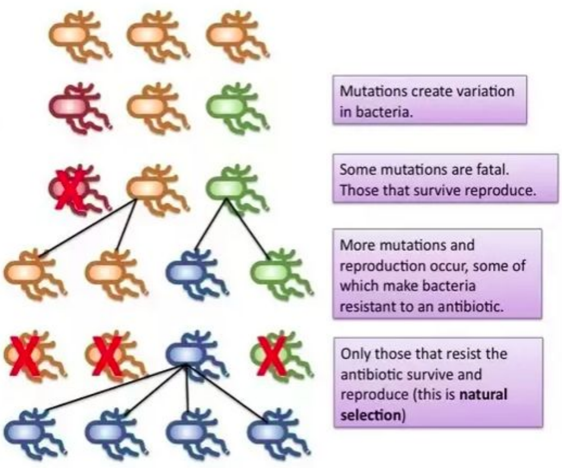

Fig 1.1

Beneficial mutations simply increase in proportion from one generation to the next because they happen to contribute to the survival and reproductive success of the organisms carrying them. Eventually, a beneficial mutation may be the only alternative left as all others have ultimately failed to be passed on. At this point, that beneficial genetic variant is said to have become “fixed” in the population. Over time, beneficial traits will become increasingly prevalent in descendant populations by virtue of the fact that parents with those traits consistently leave more offspring than individuals lacking those traits. Mutations are the source of new variation. Natural selection itself does not create new traits; it only changes the proportion of variation that is already present in the population. The repeated two-step interaction of these processes is what leads to the evolution of novel adaptive features.[1a]

1.2 Mathematical Simulation

1.2.1 Linear Algebra

Linear algebra operations play an important role in scientific computing and data analysis. With increasing data volume and complexity in the "Big Data" era, linear algebra operations are important tools to process massive datasets. The advent of modern high-performance computing architectures with increasing computing power has greatly enhanced our capability to deal with a large volume of data.[1b] These inherent benefits give us the ability to simulate real-life situations in a feasible manner.

1.2.2 Monte Carlo Method

Numerical methods known as Monte Carlo methods can be loosely defined in general terms to be any methods that rely on random sampling to estimate the solutions. Monte Carlo methods are often applied to problems which are either too complicated to be described by a mathematical model or whose parameter space is too large to be explored systematically.[1b]

II. Approach

Inspiration for adapting this approach was by this Youtube Channel - Primer[2a]

2.1 Overview of Approach

The end goal of the program is to have a working simulator which performs as close to a real world environment as possible. To achieve this, the amount of the environment under the control of the program can be modified as various parameters.

Several environment variables are decided for one instance of an ecosystem that allows for the controlling of the system. This simulation encompasses the integration of Matrix representations for a *Monte Carlo Simulation* so as to create the *Adaptive Landscape framework*.

2.2 Environment Development

The system of the simulation is based on *Monte Carlo simulation* where an inherently deterministic process can be modelled by a random process. This method of estimation is often used when there is no other alternative to arrive at the required answers to the requirements. This includes systems with several degrees of freedom such as a real biological process. A system such as the survival and the propagation of a species cannot be carried out experimentally as it is a very long term process. Such processes are often modeled using the Monte Carlo Method.[1c]

DETERMINING PI - MONTE CARLO PLOT

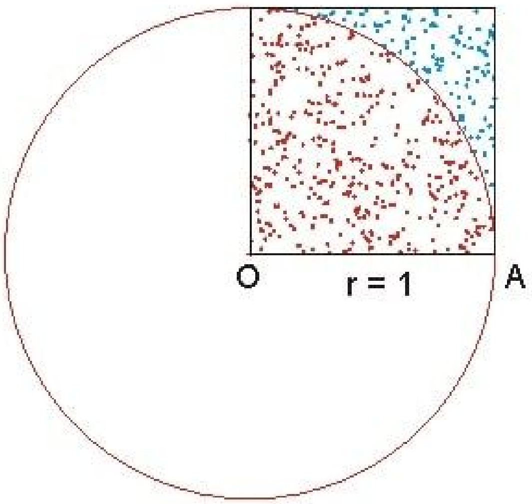

Fig 2.1

Methods of *Bayesian Inference* are used to make decisions about the behaviour of the system with an assumed prior and several observations of the process which can be carried out with a Monte Carlo Simulation

2.3 Adaptive Landscape Framework

The *adaptive landscape framework* was a tool developed to model evolutionary ideas. It is a central tenet of biology that a certain amount of genetic variability is common to all species (for if it were not, and all organisms in a species were exactly the same from one generation to the next and natural selection, and thereby evolution, could not occur). Humans, as an example, exhibit a wide range of heights, body types, eye colors. There are several other invisible traits which also greatly affect the performance of any form of life and that for the purpose of this simulation, will be ignored.[1d]

Natural selection seldom acts on genes themselves; it is, rather, these inherited traits, called *phenotypes*, that determine whether an organism dies young or lives to procreate and pass on its genes, and thereby its phenotypes as well, to another generation. Relating an organism's phenotypes to its propensity to procreate forms the basis of the adaptive landscape. We will investigate the possibility of plotting a population of organisms on such an adaptive landscape, and simulating the 'pull' of natural selection moving a population towards the peak.

A *phenotype* is a measurable trait of the being and the survival/superiority of the being is often a function of all the phenotypes of that species. Phenotypes can be measured numerically and are often unfavourable to the species if the value is in the extremes of the phenotype space. The phenotype space shows the fitness for each value of the feature and for each feature of the species and is representative of the ideal characteristics of the species.[1d]

PHENOTYPE SPACE
(2-D ADAPTIVE LANDSCAPE)

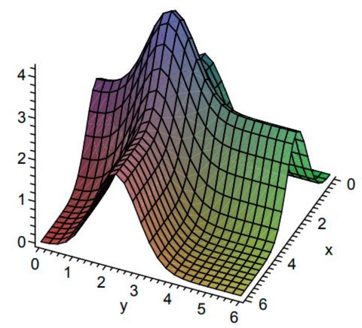

Fig 2.2

2.4 Evolutionary Stable Strategy (ESS)

ESS is one of the prominent types of strategies used in *Game Theory* to base a wide spectrum of naturally occurring events. The origin of this strategy is based on the fundamental principles of Evolution and hence, It is necessary to base the simulation on these fundamental principles. ESS is widely based on the concept of *Nash Equilibrium*. In essence Nash equilibrium requires a strategy profile that not only should each component strategy x_i be optimal under some belief on behalf of the i^{th} player about the others' strategies, it should be optimal under the belief that x itself will be played. Hence the player is not allowed to modify the strategy used and the game by itself shall result in an outcome.[1e]

III. Data Representation

Insights for the constructs is from the talk by Adam Jones[2b]

3.1 Overview of Data Representation

3.1.1 Mini World

Mini World shall define the fundamental parameters which shall remain consistent throughout the process of simulation. It shall act as the general frame-work for the simulation.

3.1.2 Epoch

changes in the fabric of life-forms resulted by evolution is studied and analysed by screening through the snap-shots between each generation. A *generation* can be defined as all of the people born and living at about the same arbitrary instance of time, regarded collectively.[3a] In the considered mini-world, each generation is considered as an *Epoch*. A series of epochs must be conducted sequentially in order to observe the changes and determine the result.

3.1.3 Population

The next step towards a more accurate model is to abandon the idea of treating a population as one representative point, and allow the model to portray an entire population of organisms spread out through phenotype space.[1d] Each entity in the population shall abide by the fundamental rules of the ESS.

3.1.4 Resource

The Resources of the mini world acts as the ceiling based on which the entities competing in the mini-world are limited. It acts as the population controlling factor in a broad sense. In the simulation interpreted, resources shall be vastly considered as *"food"* and shall vary for each generation over an average point attributing to the changes in the environment.

3.2 Population with the Mini World

3.2.1 Mini World Environment

The environment of the mini world is defined by a set of independent parameters. It shall act as the phenotypic space where in the adaptive landscape shall unfold.

3.2.2 Features

Every population entity in the system is defined by a finite set of *f* features. Every entity in the population is considered a distinct *species* based on the values of each feature it possesses.

3.2.3 G-Matrix

Chance matrix is the representation of how much each feature contributes to the survival of the being. This matrix can be modeled in two ways.

- It can be a column vector with the weights for each feature contributing to the survival of the species
- This matrix can also be represented in more than one dimension as the features are not necessarily independent of each other. The excellence of one feature might contribute positively or negatively to the survival chance brought about by some other feature.

The multi dimensional representation of such data is called *G-Matrix*. The G-matrix summarizes the inheritance of multiple, phenotypic traits. The stability and evolution of this matrix are important issues because they affect our ability to predict how the phenotypic traits evolve by selection and drift. This makes a large difference over several thousand generations and species, but for the simplicity, this was made constant for one instance of the environment and was only varied between environments for different results. This matrix more often than not undergoes transformation as the traits favourable to survival often change with time[1f].

2 TRAIT G-MATRIX

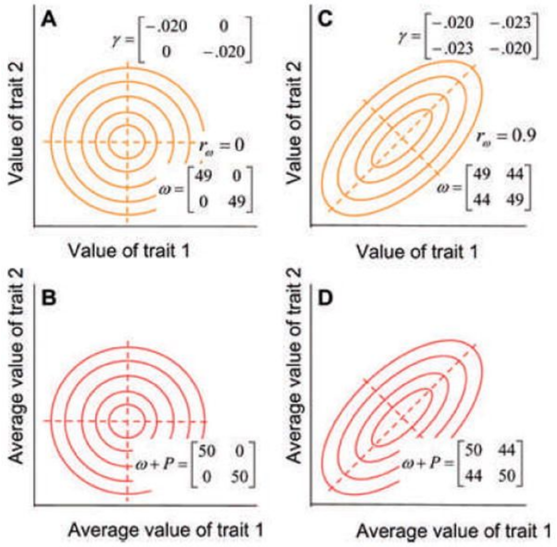

Fig 3.1

3.2.4 Population Matrix

The Population matrix *P* shall be the representative entity for the entire set of beings in the corresponding epoch. Assuming there are *n* beings and *f* features, the matrix P is of the order $(n+1)\times f$. Each row of the matrix corresponds to a vector with the set of corresponding features of a being. The 0th row consists of a *zero vector* which is beneficial during rescaling.

3.2.5 Gauss-Random Favouritism

The ability of a being to acquire *food*[ref] from the environment is the factor which determines the likelihood of persistence of the *species* to the following generation as well as the likelihood to replicate. A large amount of inheritable and diversified variability is favourable, but by giving a better chance for the appearance within any given period of *profitable variations*.[1g]

Hence to uphold “Survival Of the Fittest”, the beings better supported by the environment must have more likelihood to attain resources.

Considering the population matrix *P* and chance matrix *C*. The relative likelihood is given by row matrix *R*.

$$R = (P \bullet C) \bullet O$$

O - ones-matrix of order $f \times 1$

To statistically distribute the resources between the beings, we normalise R and rescale it within the range 0 and 1. Note that the 0th row consisting of a zero vector helps in uniform relative rescaling.

Left-Skewed Half-Gauss-Normal Curve based *Random Value Generator* is utilised to generate values for every “food” available and those beings with corresponding values in R closest to the generated value acquire a unit.

SAMPLING FROM RANDOM GENERATOR

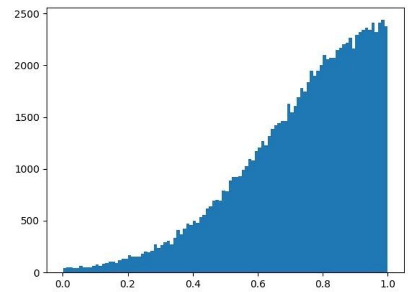

Fig 3.2

IV. Implementation

Complete Implementation - https://github.com/Rvdixit23/Environment-Simulation.git

4.1 Premise

There were multiple methods to implement such a system using matrices with various representations. Each implementation strategy has its own form of sufficiency in representation and shortcomings. The working implementation is coded in *Python 3.6*. The external modules used were - *Numpy, Matplotlib*.

4.2 Types Of Representation

4.2.1 Representation-0

This representation involved creating the entire population in a single vector with the value attached to each being as the value of a single feature of that being. This allowed for a simplicity but caused it to not be scalable to a larger number of features, and let alone the interdependence between features.

4.2.2 Representation-1

The scenario of two features, which were labeled as speed and intelligence were implemented as one vector for each being representing the excellence in each feature. This in product gives the *chance* of that being to survive in the surroundings. We are now left with a vector of the *chances* of each being in the population. This vector is transformed and normalized to contain *probabilities* between 0 and 1. The scaling down is done on a relative scale, to take into account competition between beings for survival.

The amount of food obtained is a random function of the probability of survival of that being. The survival/reproduction of the beings is further a function of the number of food they obtain.

There are several ways to model the survival and actions of the beings depending on the amount of food collected by them.

- Set a threshold for the amount of food needed for survival, for reproduction and for every child thereafter

- Create a food distribution system to simulate cooperation between beings

As the emphasis of the simulation was to apply mathematical concepts, a simple system of survival and reproduction was adopted

The being reproduces if it obtains two food, survives without reproducing with only one food, and does not survive with 0 food collected. This framework controls the fact that the beings have to compete for food, and that the successful beings have leeway for replication. The mutation model was created such that the being’s offspring has a chance of increase or decrease in its phenotypic score to bring randomness into the quality of the next generation. This distribution was chosen to be uniform to give equal opportunity for all the three types of offsprings, the identical, the positively influenced and the negatively influenced.

CLASS DEFINITIONS

```
class BeingBase(object):
        def __init__(self,
         featureValueList):
                """
                Creates a matrix to show the
                features of the being
                """
class Being(BeingBase):
        def endOfDayResult(self,
         environment):
                 """
                 Returns the action of the
                 being
                At the end of the day
                 depending on the food
                It had collected on that
                 day
                """
        def reproduce(self, mutChance,
         qualityMultiplier, chance):
               """
                Reproduces a mutated being
                 with a large chance of
                Increase in capability
                Chance of mutation is given
                 by mutationChance
                """
class Environment():

        def __init__(self,
               mutationChance,
                qualityMultiplier,
               \foodCountMean, chanceList,
               foodCountVariance=0):
                """
                Creates and
                 environment for
                 running a
               simulation in
                """

        def createPopulation(self,
                startingPopulation, \
                       feature=None,
               featureValues=None):
                """
                Creates a population in the
                 environment with the
               features  given the
               following data
                """
        def runDay(self):
                """
               Runs a day in the simulation
                Generates food for the day
                Each being gets either 0,
                 1, or 2 food
                 Beings which get 0 die
                  """
        def runSimulation(self,
        numberOfDays):
               """
                runs simulation for a
                 series of generations
                """
```

4.2.3 Representation-2

To generalise this to a complex ecosystem with beings not only with a few features but several features just like a real life scenario, another system with randomly generated features and G - Matrix was run, with similar ideas of mutation and food as the previous representation. A large number of subclasses of beings were created to represent a long term growth of the species compressed into a set number of epochs.

DRIVING CODE

```
envConfig = {
        "mutationChance" :  ...
        "qualityMultiplier" : ...,
        "foodCountMean" : ...,
        "chanceList" : ...
        }
env = Environment(**envConfig)
        populationConfig = {
    "startingPopulation" : …,
     "featureValues" : ...,
        }
    env.createPopulation(**popul
    ationConfig)
    env.runSimulation(...)
```

V. Graphs and Observation

5.1 Population Distribution Analysis

This simulation is developed based on some assumption about the distributions of the prior, in this case the functions from which all the quantities were derived. After making observations of the random process, inferences were made about the behaviour of the beings within the ecosystem. Some of the inferences made from the outcomes of the experiment are shown representatively as follows-

5.2 Population Distribution Analysis

Generated Transformation Map
(1000 Generations)

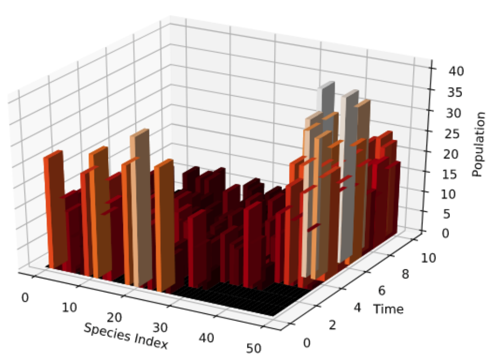

Fig 5.1

Fig 5.1 The population tends to move to a state with higher species index (Species Rank).

Origin Generation

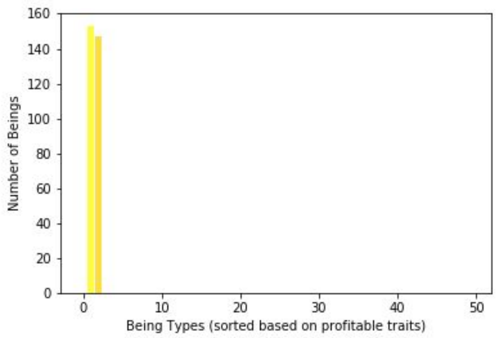

Fig 5.1.1

After 1000 Epochs

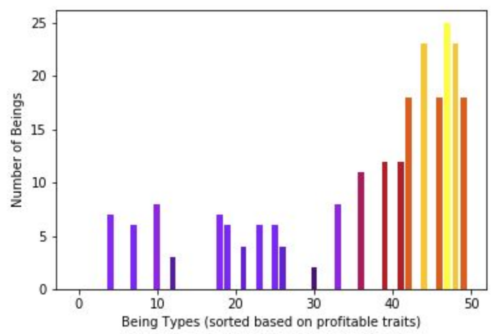

Fig 5.1.2

Fig 5.1.1 Shows the histogram of the population with the species rank as bin before the simulation.

Fig 5.1.2 The final population histogram ends up being composed of a large fraction of superior species. The existence of species of a lower rank shows Gaussian favouritism.

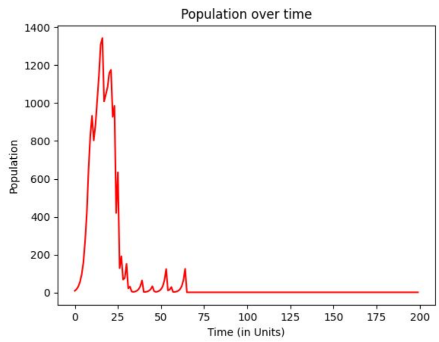

Fig 5.2

Fig 5.2 High mutation rates do not favour the survival of the species.

The ones positively affected by mutations often end up causing the downfall of a species unless the beings co-operative in nature.

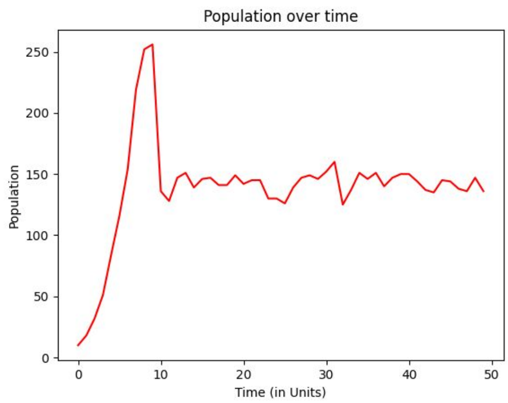

Fig 5.3

Fig 5.3 Lower mutation rates result in survival, a balanced environment and overall growth of the species.

Low mutation rates which can also be termed as evolutionary growth does not trigger extreme competition between members of the ecosystem as a large difference in capacities are not found often.

VI. Conclusion

The conclusion from the experiment conducted shows how multi parametric simulations can be carried out by representation as matrices and by their transformations. The positive outcomes of natural selection and how it works was also outlined by the outcome of the simulation. This is a perfect example for depicting the impact of applied linear algebra in the field of biology.

VII. Bibliography

1. Research Papers and Journals
    a. Gregory, T.R. "Understanding Natural Selection: Essential Concepts and Common Misconceptions". Evo Edu Outreach 2, 156–175 (2009). https://doi.org/10.1007/s12052-009-0128-1
    b. Ji, Hao. "Novel Monte Carlo Methods for Large-Scale Linear Algebra Operations" (2016). Doctor of Philosophy (PhD), Dissertation, Computer Science, Old Dominion University, DOI: 10.25777/pg8q-s553 https://digitalcommons.odu.edu/computerscience_etds/15
    c. Kroese, D. P.; Brereton, AT.; Taimre, T.; Botev, Z. I. (2014). "Why the Monte Carlo method is so important today". WIREs Comput Stat. 6 (6): 386–392. doi:10.1002/wics.1314.
    d. David Mummy "Mathematics Of Evolution" (2006). Mathematics Whitman College, Ph.D. Biomedical Engineering University of Wisconsin https://www.dgmummy.com
    e. Weibull, Jörgen (1997). Evolutionary game theory. MIT Press. ISBN 978-0-262-73121-8. Classic reference textbook.
    f. Arnold, Stevan J et al. "Understanding the evolution and stability of the G-matrix." *Evolution; international journal of organic evolution* vol. 62,10 (2008): 2451-61. doi:10.1111/j.1558-5646.2008.00472.x
    g. Darwin, Charles, and Leonard Kebler. On the origin of species by means of natural selection, or, The preservation of favoured races in the struggle for life. London: J. Murray, 1859. Pdf. Retrieved from the Library of Congress, https://www.loc.gov/item/06017473/
2. Inspiration
    a. Primer - Youtube Channel
    b. Adam Jones (Texas A&M)- Talk on Simulation-based approach to the evolution of the g-matrix
3. Terminologies
    a. "generation". *Dictionary.com Unabridged*. Random House, Inc. Retrieved 22 December 2016.