

Antibiotics Resistance of Bacteria: Evolution in Action

Omri Mezger and Shira Ben Amotz

Supervisor: Daria Dubrovin

Introduction

Antibiotics were always considered one of the wonder discoveries of the 20th century since they can prevent various bacterial infections. However, it was proven that changes in the bacteria's genes enable them to resist the effects of antibiotics. Therefore, "Antimicrobial Resistance" became a worldwide concern, while efforts are being made to prevent the phenomenon.

Bacteria become resistant to antibiotics by developing a specific mutation. This occurs rarely, and the chances for that are different for each type of antibiotic.

In our research, we examined the way bacteria became resistant to antibiotics when they were gradually exposed to it. We have used different types of antibiotics and tried to find whether bacteria become resistant to the deadliest ones. We focused on the following aspects:

1. The connection between the bacteria's chance to resist a specific type of antibiotics and the number of resistant bacteria.
2. The ability of bacteria to become resistant to high doses of antibiotics through evolutionary processes.
3. How deadly should antibiotics be in order to prevent bacteria from becoming resistant to high amounts of antibiotics. (Antibiotics is considered more deadly as the chance of bacteria to develop resistance to it decreases)

The Research Method

Our research is based on an experiment done in Harvard University in which bacteria were exposed to five different amounts of antibiotics in a Petri dish. We created a numerical model, which imitates the real experiment.

The model describes a petri dish which is divided into five parts (see Figure 1). In the first part there is no antibiotic, the concentration of antibiotics in the second part is enough to kill non resistant bacteria. In the following parts the concentration of antibiotics is 10, 100, and 1000 time higher.

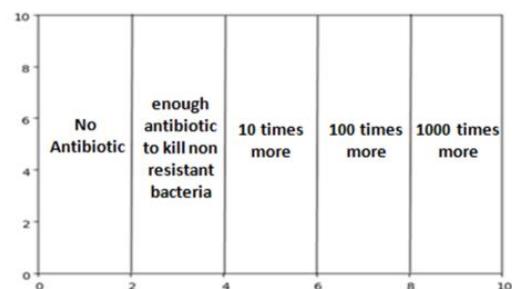


Figure 1

Initially non resistant bacteria appear in the first part of the petri dish, where they grow and spread. Once they reach the second part, they mostly die, but since there are billions of bacteria, a few become resistant by chance and are able to survive and spread in the second part. As bacteria reach the third, fourth and even the fifth part the process repeats itself. (Figure 2)

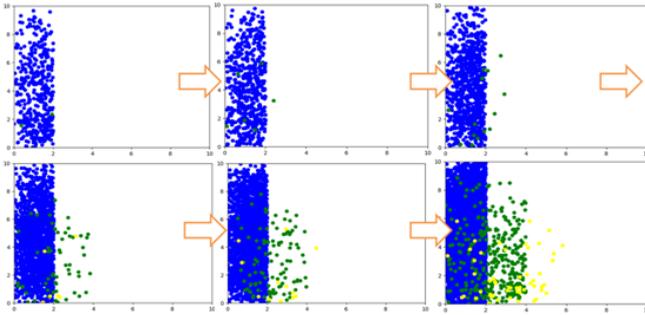


Figure 2 Each dot represents one million of individual cells.

In every experiment we have done, we changed the bacteria's chance to develop antibiotics resistance. We plotted the relation between this chance, and the number of bacteria that reached each part of the petri dish (Figure 3).

Results

The following graph shows the dependence between the chance of bacteria to resist antibiotics and the number of bacteria that reached the third part.

Most of the graphs we built, had shown a similar trend. Using a linear regression, we found the x-intercept for each graph which we interpret as the chance to resist antibiotics required to prevent bacteria from reaching parts with higher concentration. This means that if we use antibiotics that the chance to resist it is the x-intercept then no bacteria will reach the third, fourth and

fifth parts. For instance, we found that in order to prevent bacteria from reaching the third part, the chance to develop antibiotics resistance should be $6.5 \cdot 10^{-11}$, which is a very small number.

Conclusions

- **Through evolutionary processes**, bacteria can develop resistance to high amounts of antibiotics **when exposed gradually**. This means that in a “real life antibiotics treatment”, the patient should not change the dose he takes and should not stop the treatment before it is finished.
- There is a connection between the chance of bacteria to develop antibiotics resistance and the number of bacteria that reached each part of the petri dish. When we decreased this chance, less bacteria reached the parts with high concentrations of antibiotics.
- In order to prevent bacteria from becoming resistant to antibiotics when exposed gradually, we should use antibiotics that the chance to develop resistance to them tends to zero.

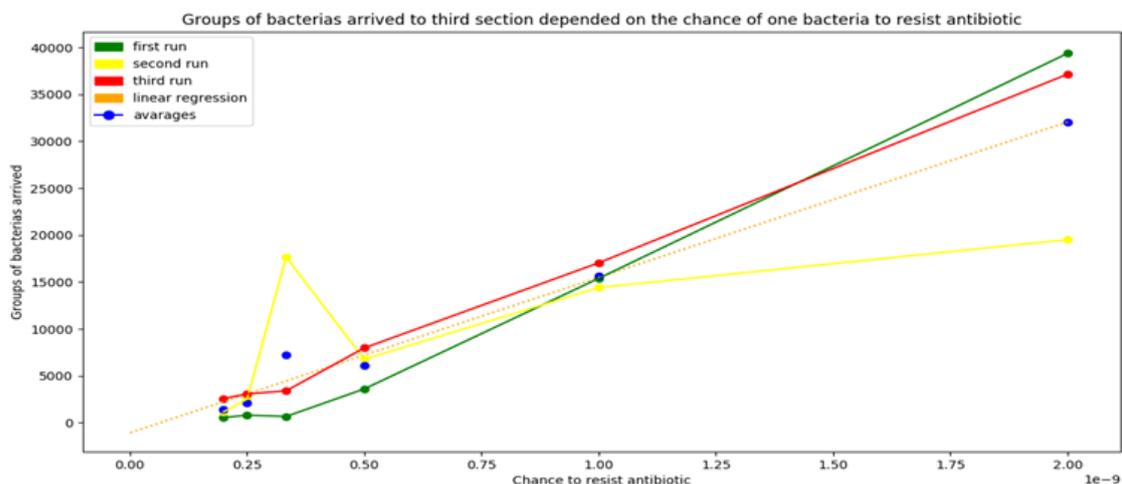


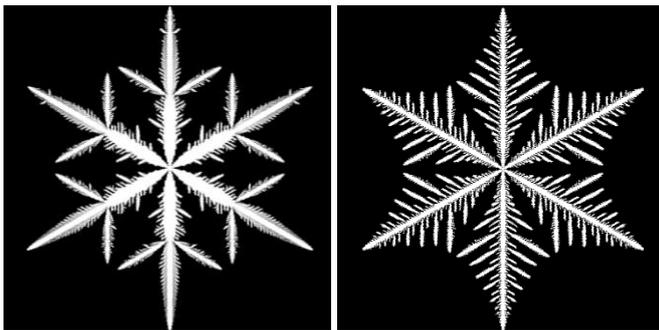
Figure 3

Growth of two dimensional snow crystal- snowflake

Introduction

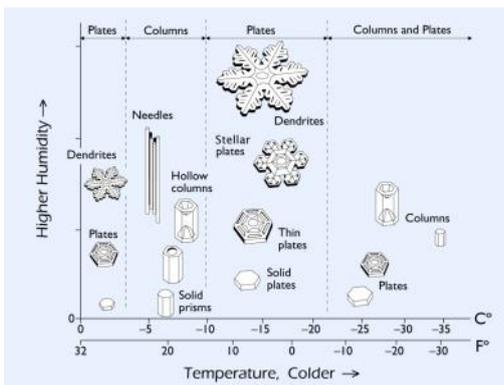
Snowflakes are dense water vapors which freeze around a nucleus (usually a speck of dust). The water molecules are organized in a precise hexagonal structure.

The snowflakes' structure depends on various environmental parameters like temperature, humidity, diffusion constant, cloud's height, wind etc. That is why there are no two identical snowflakes.



Our model results

In our research, we attempted to confirm Nakaya Diagram- a table that was developed by Ukichiro Nakaya in 1940. The diagram sorts the different kinds of snowflakes to different arrays, depending on the temperature and the humidity. Based on Nakaya, we tried to find proportion to the diffusion constant equation.



Nakaya Diagram

Model

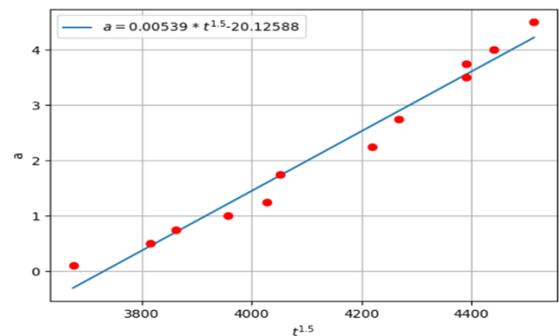
Our model simulates the growth of two-dimensional snow crystal depends on 4 parameters- humidity percentage in the air, diffusion constant, humidity percentage which is added from the air, and number of steps (which reflects the time of the snowflake in the air).

The snowflake is represented in the model as a matrix; in every step the matrix splits to a matrix

which performs diffusion between the snowflake's cells and a matrix which performs diffusion with the air. At the end of each step, the two matrices are combined. The model generates a picture of the resulting snowflake.

Results and conclusions

We fixed two parameters (added humidity, number of steps), and ran the model for various diffusion constants and humidity values. Then we built our own table, and compared it to Nakaya Diagram. We found the proportion between our diffusion axis and Nakaya's temperature axis (power 1.5). We proved Nakaya Diagram, and the diffusion constant equation.

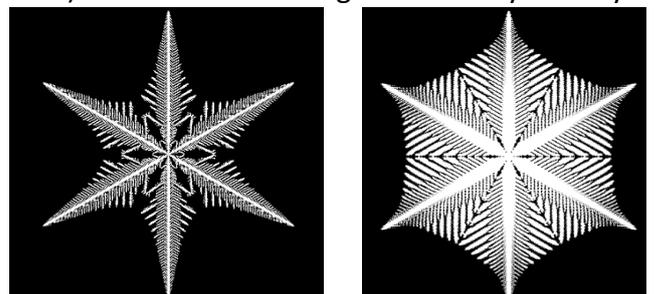


The proportion between diffusion constant and temp^{1.5}

Future uses

There are other factors that can influence the snowflake, like the weather condition, the wind and even dust.

We fixed some parameters, although they can change at any moment (temperature, humidity, pressure...). That is why our model is not completely realistic. In future research, we plan create a simulation where these parameters also vary, and also create 3 dimensional snowflake matrix to get a higher precision. Another interesting option we considered is to explore the influence of impurities (such as a speck of dust) on the snowflake's growth and symmetry.



Our model results

Developing an algorithm to create a snake that knows how to operate in the game world

Genetic algorithm VS Depth First Search algorithm

Presenter: Eran Halperin

Supervisor: Avi Ben Simon

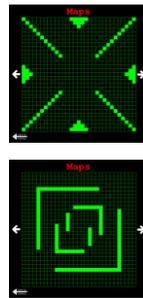
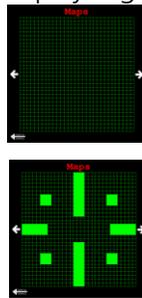
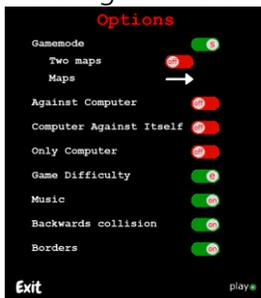
Introduction:

The Snake game was invented in the 1970's.

The game is simple – the player (the snake) has to eat as many apples as he can without eating himself or bumping into walls. Every time the snake eats an apple, his body lengthens.

Based on these rules, I built a game of my own, I added maps and new game mechanics. When the game was ready, I began working on the final project which was to create algorithms that could "play" against a human.

Based on these rules, I built a game of my own, I added maps and new game mechanics. When the game was ready, I began working on the final project which was to create algorithms that could "play" against a human.



Algorithms:

1. Depth First Search (DFS) algorithm - This algorithm grades all of the options (paths) which the snake can perform in a given position with a given number of steps. According to these grades, the algorithm chooses the most optimal path.
2. Genetic algorithm (GA) - This algorithm creates a large number of genetic lists before the start of the game. Each genes list contains the snake's chances of heading in a certain direction. Every genes list (snake) is run on a simulated game before the "real" game begins for a fixed period of time, and receives a score according to it's "behavior" in the game world. When all snakes finish running, a relative "pairing" (a snake with a higher score will get more genetically important) is done for each two options with the closest scores. This act of simulated games and matches is called a "generation", and repeats itself until one list remains. This is the best snake and it is the one that will play the real game.

Research process:

In my research I wanted to examine the preferred algorithm between the two. In order to put the algorithms into a fair "head-to-head" game, I had to first find the optimal value of each of their key components. For the DFS algorithm, this component is the number of steps that the algorithm looks forward. For the GA, this component is the number of "generations" which the computer should calculate before beginning the game.

To do this I conducted an experiment. For each component I checked a range of values (starting at the smallest possible and ending at the largest, which a normal computer should be able to run "smoothly"). For every value I examined these three dependent variables:

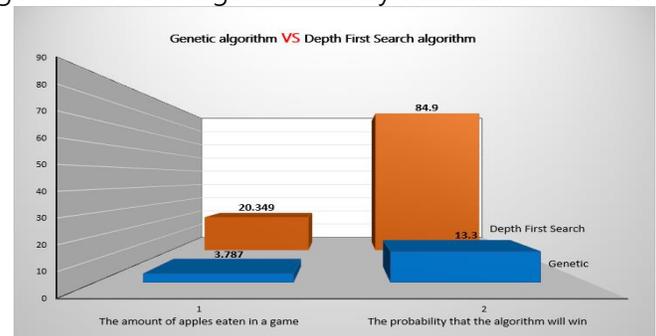
1. The number of apples the snake eats in his first thousand steps.
2. The number of apples which the snake eats throughout the game.
3. The number of steps the snake "walks" throughout the game.

The component value was decided when these variables reached a common peak (their best average state). After finding the optimal values for the components, the comparison was made. It included two dependent variables:

1. The chances (in percentages) of one algorithm to win the "head to head" game against the second algorithm.
2. The number of apples each algorithm eats throughout the game.

Results:

According to my tests, I found that the DFS algorithm gives a much better AI than the GA does. I showed that the DFS snake is winning approximately 6.38 times more than the Genetic snake. It was concluded that the DFS algorithm on average is better by 587.85% than the GA.



Conclusions:

According to my results, I concluded that the DFS algorithm's ability to analyze many of the future outcome is what makes it better than the GA.

Future research:

In the future it could be a great idea to examine the "behavior" of an algorithm which would be the unification of the two algorithms. For example, I could add a gene which chooses the number of steps a snake can look forward or, add genes which choose how to grade the paths (how many points to give (or) take, to (or) from a path score for a particular move).

Wood-Wide-Web - Models of Forest Development

Authors: Daniel Peled and Ori Vigman

Susan Simard, a forest researcher, wanted to investigate the hypothesis that the trees communicate through an underground root system. She conducted an experiment in the forest as follows: she covered 80 trees of three species in plastic bags – the paper birch, the Douglas fir, and the red-western cedar, and sealed them with thick tape. For each type of tree, she injected through the sealed sacks radioactive isotopes of carbon. She waited an hour, which is the time it takes for trees to absorb carbon dioxide and do photosynthesis. Then she activated the Geiger counter to trace the traces of isotopes. She discovered that the radioactive material injected into the fir trees had reached the birch trees, and the radioactive material injected into the birch trees reached the fir trees. The cedar did not transfer radioactive isotopes to the other trees and did not receive radioactive isotopes from them. Simard concluded that the fir trees and the birch trees communicated and cooperated through the root system. In contrast, cedar did not cooperate with the other trees. In other words, there are species of trees that cooperate and others that do not cooperate. Simard testified about her discovery: "I knew I found something big, solid evidence of the massive communication network of trees in the underground forests, which is called the "Wood-Wide-Web". Simard later discovered that each forest has large, old trees that have the most connections to the network. She called these trees "mother trees" or "Hub trees" because they supported young trees in their vicinity and increased their chances of survival.

In this work we used the model of "Conway's game of life" (and other models that differ from the original model in the rules of the game) to simulate the development of forests under different conditions and predict how the forest will look after a certain number of generations. We examined the effect of the connections between the trees in the forest on the prosperity of the forest and. We incorporated Simard's findings into our models.

Using the models, we attempted to reveal the secrets of the underground network. If we can understand the interactions in the forest it can help

planning effective forestry and to predict the map of a natural forest after several generations.

"Conway's game of life" is designed to simulate the development of biological cells or systems according to simple predefined rules. The cells are arranged as a two-dimensional array, so that each cell has 8 neighboring cells. Each cell will be defined as being at one of two states: "alive" or "dead". In each generation, cells are born and die according to the following game rules: A dead cell will be born if it has exactly three living neighbors. A live cell will die in two cases: If it has only one live neighbor or less, it will die of loneliness. If it has four live neighbors or more it will die from overcrowding. In case a cell is alive and has two or three neighbors, it will remain alive.

Because we wanted to simulate a natural forest, we assumed that the initial dispersion of the various species of trees in the forest was random. We gave ourselves the option to choose the following parameters: percentage of live trees of each type at the initial state; relative "power" of each type of tree ("strong" trees have an advantage over others in survival and chances of birth, thus in the model "strong" trees play according to preferred rules of game that we defined); existence or non-existence of "Mother Trees" and so forth.

We followed the development of the forest using an animation program and a quantitative graphical program. In the animation program you can visualize how the number of trees in the forest and their distribution changes through the generations. In some of the models we reached a significant reduction in the number of trees after several generations and an uneven distribution of trees, while in other models we reached a steady state and even an increase in the number of trees through the generations.

We discovered that some of the models we ran were not effective. For example, Conway's "game of life" model, results in a significant reduction in forest density over generations. Using this model we found

a steady state of single species forests of low density, less than 10 percent. On the other hand, in some of the models, we were able to simulate a stable forest from the very early generations, where the initial density of trees is preserved, and in other models we were even able to simulate a forest that develops over the generations with an increase in the number of trees.

We demonstrated through the models how tree cooperation increases the total amount of trees in the forest and preserves its diversity. On the other hand, competition reduces the total amount of trees and leads to extinction of the weak species (unless there are mother trees to support them and to prevent their extinction).

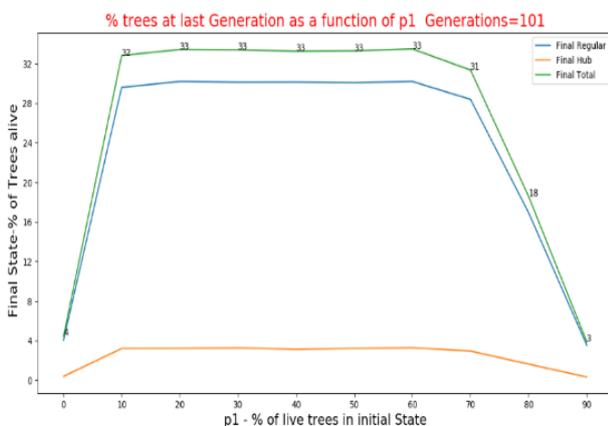
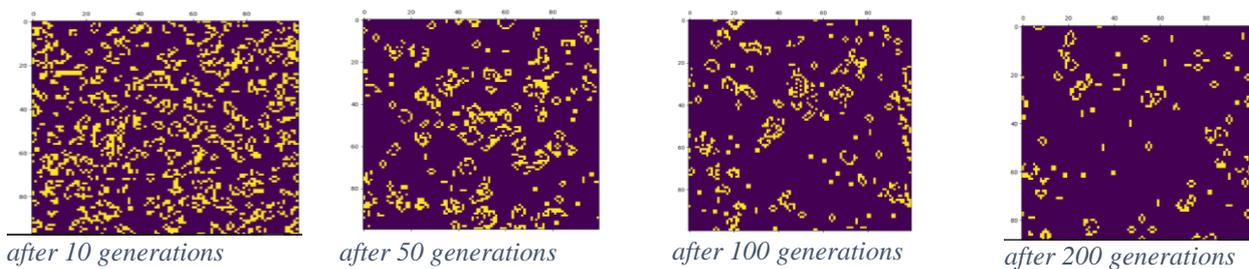
Using our models, we showed how very few "mother trees" in the forest (1%-2% of the trees) help to maintain forest density and even increase it,

contrary to the findings from the basic model from which we started - Conway's game of life.

In order to test the model empirically, experiments should be carried out in forests, such as mapping of the Wood-Wide-Web system.

Our models can explain the existence of forests with a density of up to 40%. They do not explain forests of higher densities, even though they exist in the world. However, it is not straight forward to relate the concept of "density" as used in our models to the concept of density in practical mapping of a forest. Furthermore, the models we have developed are simple and ignore significant factors such as the environmental and climatic changes taking place over the generations, or the unequal conditions that various trees receive within the forest itself.

Examples of the results of models we ran: Animation Program – Development of forest of initial 30% density Conway's game of life model.



Quantitative graphical program:

Model of single type tree forest, containing "regular" trees and "mother" trees:

The graph shows the final density of trees in the forest, depending on the initial percentage of the regular trees. Selected parameters: The percentage of initial mother trees: 2%. Probability that a regular tree will become a "mother tree": 2%. Probability of mother tree to die: 10%.

We can draw an interesting conclusion from this graph: For an initial density of trees between 10% and 70%, after 100 generations a final tree density of about 32% is reached, regardless of the initial density.

In contrast, in extreme situations (80% and 90% initial density of regular trees), the number of trees sharply decreases.

A Computational Model That Describes The Dynamics of a Body on an Inclined Rotating Plane

Daniel Yakoby

Supervisor: Shlomo Rozenfeld

In our paper we studied the dynamics of a body on an inclined plane with friction. In addition, the body is attached to a spring and the plane rotates. It is important to note that we examine the system in a rotating reference frame when the observer is located at the rotation axis. The basic system presented in our work consists of a body attached to a spring on an inclined frictionless plane. From here we gradually expanded our system: we added friction to the plane and then we added the fact that the plane rotates. Throughout each step, the new system will be examined. Moreover, we studied a few physical phenomena such as conservation of energy and harmonic motion.

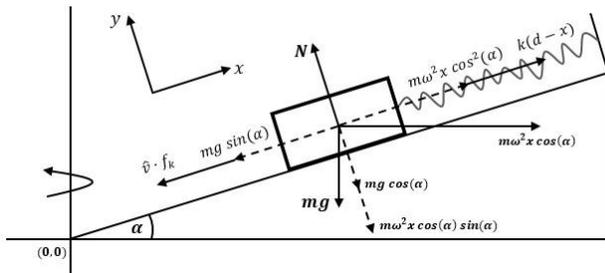
Our work contains a complementary computer program that creates a graphical simulation of our system and outputs the relevant graphs for studying the system's dynamics. In addition, the program allows one to choose one's configuration of the desired physical system. This is possible due to the fact that almost all the parameters (for example, the spring constant, the angular velocity, the kinetic friction coefficient) are modifiable.

In general, the goal of our work was to study the system we described above. More specifically, we wanted to find the position of the body as a function of time and to study the body's dynamics from it.

Results

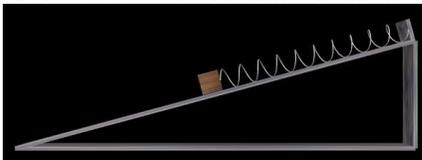
Describing The System

The figure below depicts the force diagram of the body. In the abstract we described the system briefly. For a full description please refer to the written paper.



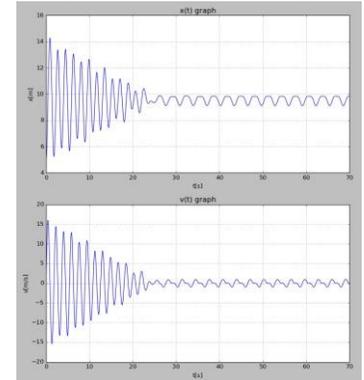
The Simulation

For the visualization we need to find the position of the body as a function of time. In order to do so, we used the differential equations that were derived in the theoretical background of the written paper. We solved the equations using the Runge-Kutta 4 method, an iterative method for approximating a solution of ordinary differential equations. Below we see a screenshot from the simulation.



In our written paper we study a lot of different aspects regarding our system. Here we will present only one interesting result.

Assume that the angular velocity is a periodic function with respect to time. Then, after the system stabilizes, the position and the velocity of the body will be periodic functions with respect to time. We can see this phenomenon in the figure above.



Summary

Although we have not conducted an experiment we verified the model from three different points of view: the friction force, conservation of energy and simple harmonic motion. We approved that our model agrees with the theoretical results that we expected.

Despite that, the model has a lot of boundaries. When "unreasonable" parameters are inserted the model would not act as physically expected. The model is also limited in it's scope due to it's restraining postulations. In the future we hope to expand this work to determine if the system is physically acceptable. The written paper contains more information concerning those subject.

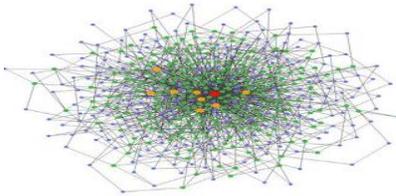
The Spread of a Pandemic on a Network

The use of networks as a model to describe human society to pandemic prediction purposes.

Noam Niv and Raz Swissa

Supervisor: Shlomo Rozenfeld

Introduction:



A network is a mathematical structure which describes vertices (nodes) which are connected to each other by edges (connections).

The research of network is an old field of knowledge, but before the use of computers there were no efficient tools to study networks.

Researchers had been using mainly visual and maps of the networks they wanted to study, and they relied on the human brain's visual abilities as their tool to analyze those networks. The main limitation of this method is that big scales of networks cannot be researched with it, as there is no human who can draw maps to describe them and there is no human who can analyze them by his eye.

Today, when we have computers and their great computational power in our pockets, we investigate networks at very large scales (dozens of thousands to dozens of billion points). The network is an efficient model to describe the human society and for this purpose we will use them. We will investigate the spreading process of an epidemic over different types of networks and compare them, and analyze the effect of a vaccine upon the spread of the epidemic depending on the type of net on which it is spreading over.

Simulation:

Our simulation uses networks to simulate the society. With this assumption, we had introduced two basic functions. In the basic function, as the simulation begins, we infect one vertex and then there is a certain chance for it to infect other nodes at each time step. An infected node also has a chance to recover and then it is vaccinated and can no longer be infected. It then produces a plot of the number of infected vertices as a function of time.

The second simulation is a loop of the basic function in which, a growing percentage of vertices is vaccinated for each run of the basic loop and a graph of the maximal percentage of infected vertices as a function of the percentage of vaccinated vertices is made.

Results:

- In the basic simulation we can see consecutive results in multiple runs and in each run networks which had few centers (vertices which were connected to many other vertices) had a higher peak: a higher maximal number of infected vertices.
- In the second function we got an interesting result- the maximal percentage of infected vertices is approximately linear. Obviously, the more vaccinated vertices as the simulation begins, the maximal percentage of infected vertices gets lower.

For Further Research:

- In further research there are few more parameters to consider- first, opening conditions and connections can be changed during the simulation, second, the size of the population can change not only directly by the infection but also by other factors. Moreover, in developed countries sick individuals are usually being isolated to prevent infection so we would expect the chances of infection to go lower as time passes.
- More futuristic methods such as a vaccine, which passes as a virus between vertices, can also be researched with this simulation.

The colonization of Mars

Sharon Sorin, Omer Lev

Rozenfeld

Supervisor: Shlomo

Introduction

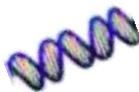
Today, many believe humanity's future lies in other planets, rather than just Earth. To be specific, Mars. This lead to many companies working on the option of sending people to live on Mars. However, even when we can make Mars a habitable planet (or at least a portion of it), we wouldn't be able to send many people.



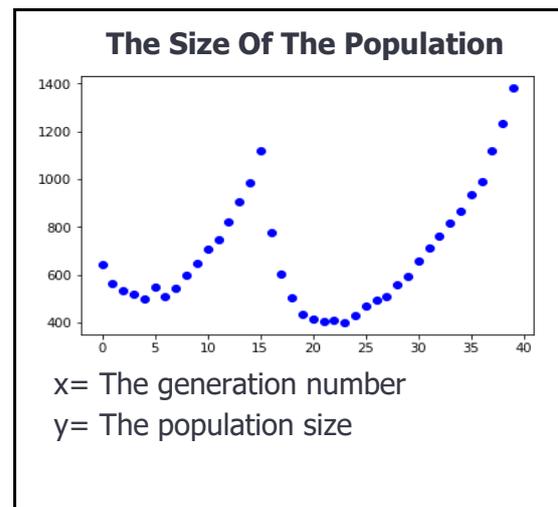
Research methods

The project emulates a human population on Mars for a number of generations. In order to simulate humans, we used a known genetic model and its properties such as: the importance of genetic diversity in a population's ability to survive a changing environment. In addition, we used social studies principles, and ecological reproduction ideas.

The term 'minimal population' exists in biology under the name 'MVP' (minimum viable population). Through the years scientists tried to associate it to certain species using their own models and rules. One of these, concerning humans, is the '50/500' rule. This idea refers to humans' ability to survive considering the needed genetic diversity in order to sustain through an environmental change while keeping a functioning society. This rule and others are the basis of the validity of these results.



For a survival chance of 99%, the initial population should be around 1720 unrelated people. Another result we noticed is along the passing generations, the genetic diversity lowers as expected. This process imitates the natural selection.



In our model we also looked at the model's behavior in extreme cases. We checked how it reacts to different timings of the environmental change, and what happens after 40 generations.

Our model's results were, in essence, similar to the existing theories.

For further research



In the future we hope to research:

- A scenario where more than one environmental change occurs.
- An expansion to the algorithm which includes a better genetic model (not only mendelian properties).

And more...

Results

According to our model, in order to keep a population survival chance at 95%, the required initial population should be 810 unrelated people.

To conclude, we can say that maybe one day we'll get to see 810 people wearing space suits taking off to Mars.

Lagrangian Points

By: Tamir Asfa, Boaz Montilya

Supervisor: Shlomo Rozenfeld

Introduction:

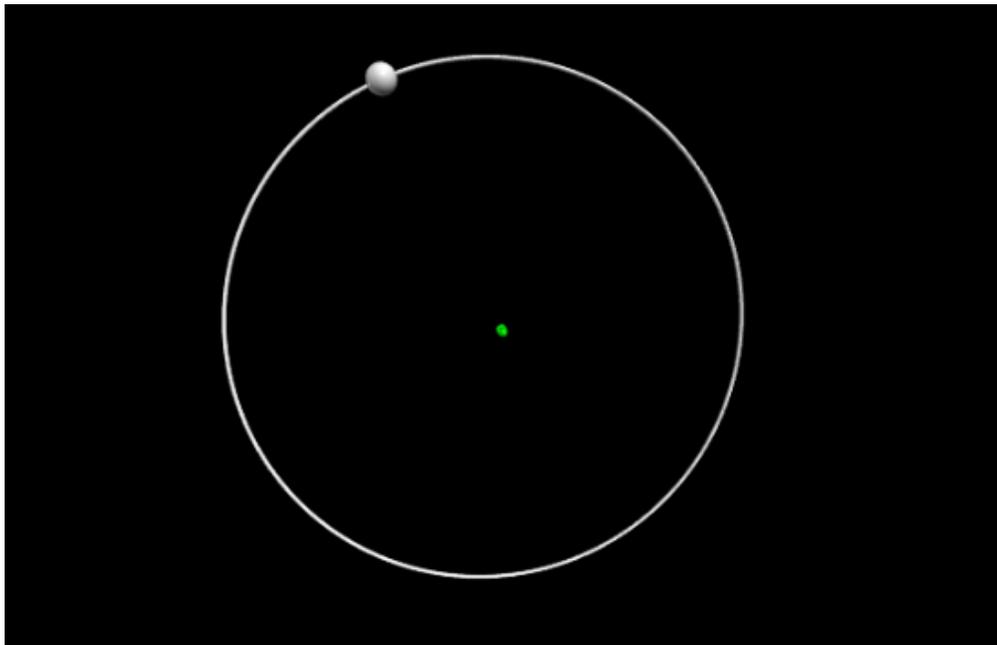
The Lagrangian points theorem claims that for every two-object system in which one small object is orbiting a second, massive object, there are five points in which a third object with neglectable mass (a satellite) can be placed and it will preserve its orbit with a 1:1 resonance (same orbital frequency) with the satellite. The theorem also indicates two points which should be stable, so small errors will fix themselves and three points that are not stable, so small errors will be magnified over time, causing the satellite to either crash or reach the escape velocity and leave the planetary system. In our project we simulated the Lagrange points in the earth-moon system, and checked their stability.

Simulation:

We've built a Python simulation that lets the user create a few objects and map their movements in a 3D space assuming gravity is the only force acting upon them. Using this simulation, we created a 2-body simulation of planet Earth and the Moon and added a satellite to be able to test the Lagrangian Points.

Validating Our Tools: To validate our simulation, we simulated the earth-moon system, and saw that the moon repeats the same orbit precisely every time, which indicated our simulation is

(The

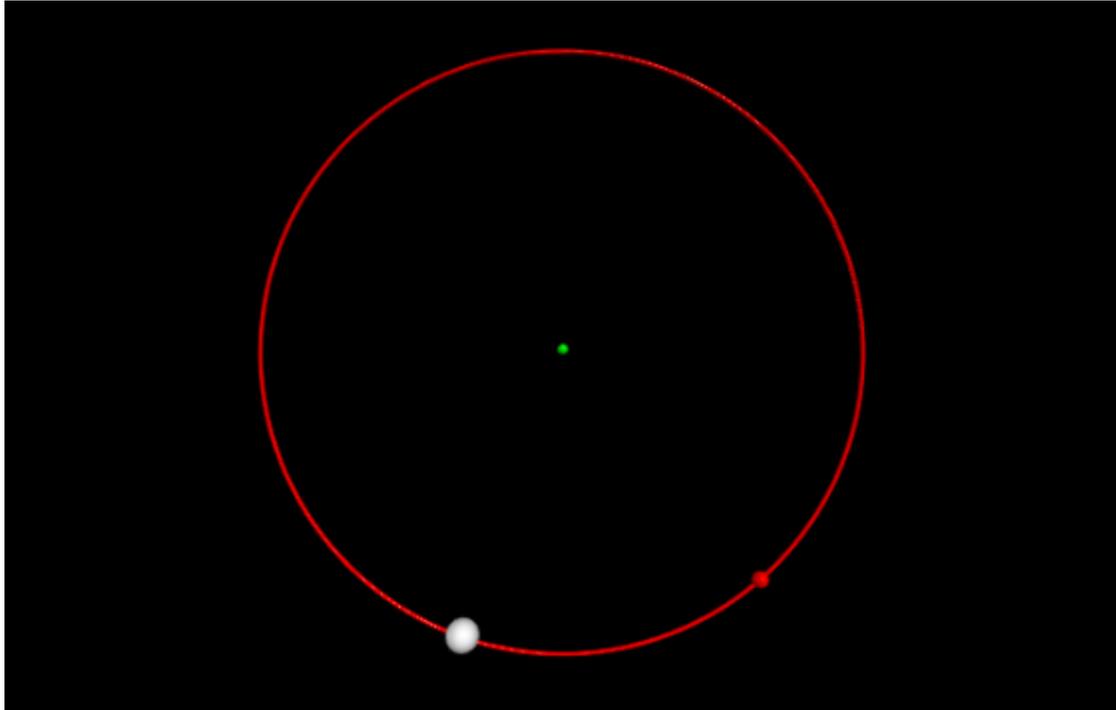


precise
Moon is

exaggerated for visual clarity).

Results:

We saw that the unstable Lagrangian Points were indeed unstable, and either drifted away or crashed quickly when we miss-positioned them, while the stable points' satellites have gone periodically around their point when we miss-positioned them. We also found an interesting periodical movement in point L1 and we found out that the point L3 was much more stable compared to the other unstable points.



Picture of L4 circling earth in the simulation

Conclusion:

Our conclusion was that the Lagrangian points theorem is correct, and that the Lagrangian points do exist. With this proof a new possibility of safe space expeditions is presented. A drone / spaceship / satellite could orbit around a two-body system and conduct research while saving fuel entering a stable Lagrangian Point.

Future research:

More research can be made on the better stability of L3, whether to position there satellites or to check what causes this stability and using this principle in other fields. One could also test the Lagrangian theorem in a different 2 body system with a bigger mass discrepancy (i.e. Sun-Earth system) and check for any difference in results compared to the Earth-Moon system.

Abstract: Long bone tumor segmentation and imaging for lesion Analysis

First Author: Roy Rosman – Nathanson, Hemda
Correspondent Author: Mr. Shlomo Rozenfeld, Hemda

Keywords: Unsupervised Machine-Learning, K-means Algorithm, K-means Segmentation, Lesion Analysis Segmentation.

The purpose of this document is to present a novel approach for Bone Tumor Segmentation using Unsupervised Machine learning.

Introduction

The grievous cancer disease has been long known to cast death and sorrow upon the human kind. The constant fight for life against the disease have yet to find a cure. But nowadays with proper identification, the condition of a patient with a tumor could be significantly improved by physicians who can in some cases save the life of the patient using the right treatment. In order to determine the proper treatment the patient should receive, most cases require surveillance and analysis of the tumor and identification of its potential malignancy based on its behavior. The recent incorporation of computers and constantly evolving algorithms to the fight against cancer may tip the scales.

Purpose

The method presented in this work is purposed to be used as a tool for the Radiologists. It analyzes the CT scans of a patient's long bone over time in order to determine the characteristics of the lesion and give a fast notice in case the tumor underwent a malignant transformation.

Method

Initially the program requires the patient's CT scans data in order to analyze the progression of the tumor. After these have been uploaded to the program's data base by the doctor, the program runs the following major steps:

(1) Normalization: The CT scans are being edited into a new format, with a unified contrast. This is being done by a histogram equalization method.

(2) Segmentation: The equalized images are segmented by applying an unsupervised machine-learning algorithm, K-means, to find the best clustering of the pixels in the image.

(3) Lesion area calculation: Prior to the calculation, the segment of the lesion is identified by the supervisor (i.e. the Radiologist). The area of the lesion is being calculated by the program using a pixel mapping method.

(4) Area comparison: The maximum area of the lesion in each CT scans group (e.g. Initial scans & Past 6 months scans) are being compared in order to set the scans on the same level.

(5) Determining the lesion characteristics: After the program has built a set of two images of the exact same infected area scanned over time, it makes the decision about the characteristics of the lesion based on the change of the area of the lesion.

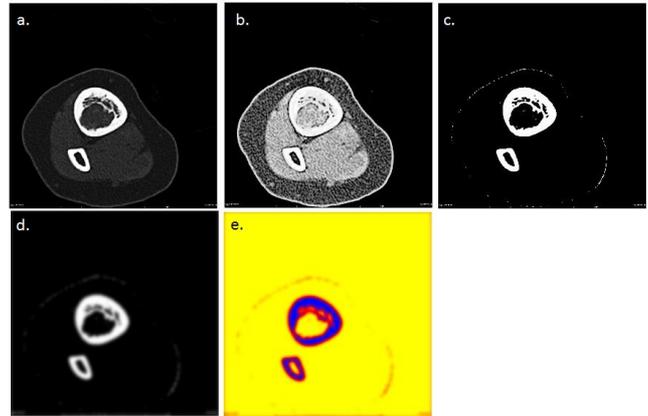
These 5 major steps ideally do not require many imaging techniques, however some image processing was required.

A total of 5 image filtering processes were used:

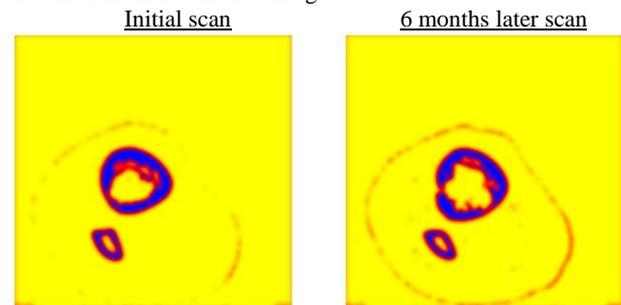
- a. RGB image -> greyscale image (using luminosity conversion)
 - c. greyscale contrast adjustment (using histogram equalization)
 - c. greyscale image -> binary image (using Thresholding)
 - d. binary image -> blurred greyscale image (using Gaussian filter)
 - e.* blurred greyscale image -> RGB image (coloring the segments)
- this conversion is made for the comfort of the user based on the segmentation process.

Results

A summary of the image processing of a patient:



The maximum lesion area images set:



The decision: "the lesion area has increased by 7.2%"

Conclusion

The program is capable of analyzing the progression of a lesion over time based on a main characteristic, the area. Although the program is able to identify and analyze most inner bone tumors, it cannot recognize and apply the segmentation on CT Scans of a tumor on the outer layer of the bone as of yet.

Discussion

This program is a cornerstone in the development of a larger, much more complete program once an extensive data base will be available. The program could be improved to apply supervised machine learning in order to classify the tumor. The two major future additions that will take the program to the next level are:

- 1) Adding more parameters to the analysis. By increasing the vector of parameters of the progression of the lesion (e.g. the shape of the lesion) will result in more precise decisions.
- 2) Applying machine learning classification algorithm. With the ability to analyze a vast data base, the program could learn how to scale to lesion progression parameters in order make a decision about the type and behavior of the lesion and even suggest the treatment.

Ice cream – won't make me fat

Do different microbial compounds have different sugar-breaking capabilities?

Noa Klein

June 2017

Introduction

The microbial is an array of micro-organisms, including bacteria and fungi, which lives in a specific area of the body that serves as its habitat. The human microbial consists of the microbial of the mouth, the microbial of the intestine, the microbiome of the skin and more.

Bacteria and fungi compete for food, and most foods that humans cannot digest are used as their energy sources. They take advantage of sugars in the host intestine, and during the process of cellular respiration they break them down. Variability in the bacterial genomes, is responsible for variability in energy source preferences, and for microbial contribution to host (the person harbor this specific microbiome) digestion process. Moreover, since no two persons contain the exact identical microbiome, differences in microbial population dictates differences in the effectiveness of food consumption and food digestion.

In this project we wanted to explore possible differences in sugar-breakdown related genes in human intestinal bacteria. Furthermore, I wanted to demonstrate the existence of variability in fungi responsiveness to different types of sugar and sweeteners.

Research method

The study used two research methodologies: experimental and computational. First, by utilizing computational methods I examined the array of genes of intestinal bacteria associated with the breakdown of sugar. The difference in the arrays of genes responsible for sugar breakdown between the different bacteria, indicates on a different sugar consumption pathways. Second, we examined the effect of four sugars and artificial sweeteners on yeast, which represented the fungi of the intestine.

The computational experiment utilized the human microbiome project database, which lists the genes responsible for the breakdown of sugars in each intestinal bacteria. Our algorithm compared, the number of different sugar-breakdown related genes between different bacterial families, and classified them accordingly. I showed that there

is a difference in the number and type of genes responsible for the breakdown of sugars between different bacteria.

In this experiment, the algorithm studies and categorizes the genes of each bacterial family. It should be noted that even within a bacterial family there are slightly different gene combinations, and the genome is not exactly the same. The algorithm develops artificial intelligence, through which it learns how to classify the different bacterial families.

The experimental part of the project deals with fungi, and the effect of sugars of various types and sources on the rate of cellular respiration carried out by yeast. I asked whether there is a different effect of sugars on the rate of cellular respiration in yeast.

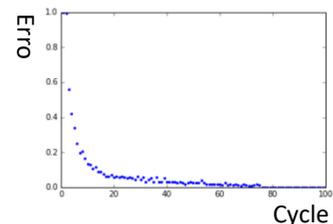
To test the effect of sugars on yeast, and whether each sugar has a different effect, I performed an experiment on four sugars / artificial sweeteners - sucralose, glucose, stevia and agave syrup. When the yeast carries out the cellular respiration process in which they break down sugars, carbon dioxide is released into the air in the form of bubbles, forming foam. A greater amount of foam means high cellular respiration rate, and vice versa. Carbonic acid is also formed following the melting of carbon dioxide in the water. Carbonic acid lowers acidity in vitro. In other words, the lower the acidity level, the higher the rate of cellular respiration. Thus, it is possible to determine which sugars or artificial sweeteners supports a more efficient cellular respiration in the yeast cells.

Results

The computational analysis

The ability of the algorithm to succeed during the cycles is evident, and the error decreases, as can be seen in the graph.

The graph also shows the number of cycles required for the algorithm to stabilize the learning process - about 77 cycles - then the error in learning tends to 0.



In addition, the algorithm returns the percentage of error in the application of the legality of classification of untested items, according to the bacterial families studied previously. The calculated error is 0.073, which means about 93% success.

The yeast experiment

I measured the foam formation and the acidity of the yeast media, incubated in a growth medium containing three sugars / artificial sweeteners - sucralose, glucose and agave syrup. - The cellular respiration was evaluated by measuring the height of the foam. In addition, the pH level was tested before and after the experiment. When stevia was added to the growth media, no foam was formed at all. Stevia is an artificial sugar that the yeast cannot break down, meaning that it lacks calories for them.



Experimental different test tubes

For all sugars there is a decrease in acidity. In the test tubes of sucralose, glucose and agave syrup, the decrease in acidity was explained by cellular respiration. In these test tubes, cellular respiration took place, and I know this also because of the formation of the foam.

In the test tube with the stevia there is no foam, so I understand that there was no cellular respiration. And yet, the acidity in it has declined. This is probably due to another chemical reaction between the stevia and the water.

Conclusions

From the ability of the algorithm to classify bacterial families, it can be understood that there is a fundamental difference in their bacterial genome. A change in the composition of the intestinal bacteria means a change in the genes found in the microbiome. The genes are related to the breakdown of sugars, confirming the conclusion that each sugar has a different effect on different people and their microbial

microbes, and that each person must consume food that can be broken down by his intestinal bacteria.

In the yeast experiment, the height of the foam and the acidity level are an indicator of the breakdown of sugars and the existence of cellular respiration. It can be understood that some of the sugars caused cellular respiration and some did not. This suggests a different effect of each sugar on yeast. The yeast has certain genes, which allow for the dissolution of specific sugars, and this is probably the source of the difference in their response. Microbial fungi are also affected by sugars. Some sugars can break down and others cannot. The composition of the intestinal fungi varies according to the sugars that each person consumes, and thus the fungi are able to break down the sugars from which they feed. One can conclude that each sugar affects people and their fungal microbial in a different way.

The combination of the two methodologies - experimental and computational - shows that each sugar has a different effect on bacteria and fungi. The human microbial is made up of bacteria and fungi, so the effect of each sugar on it is different and depends on its composition. For each composition of bacteria and fungi there are various sugars that can be efficiently break down. The microbial population varies among humans, so each sugar or sweetener has a different effect. It can be concluded that for each person with a different microbial composition there are foods that allow optimal utilization.

The microbial composition changes as a result of the food we eat. After eating a large amount of a particular type of food, the intestinal microbiome varies accordingly, and bacteria and fungi that can break down its components, multiply. In this way, the population of bacteria and fungi in the intestines changes from normal assembly, affecting the entire host organism. The usual microbial composition of each person has specific food that is most effectively broken down and utilized optimally. This food, which are broken down and are utilized by the intestinal population, are the best food for the consumption by the host organism.

Application of Genetic Algorithms on Artificial Snakes

Eli Gross & Amir Livny

Introduction

Genetic Algorithms are algorithms which address problems by imitating the process of Natural Selection. The basic principles of Natural Selection are: Variation of the starting population, Selection of the fittest, and heredity – passing the genetic information to the next generation.

For our project, we created a game, similar to the famous game “Snake”. The goal of the game is to stay alive as long as possible by avoiding collision with a snake or a wall. Each snake is controlled by its brain – a neural network. The brain gets the snake’s view of the board as input, and outputs the decision of the next move. The options are going straight ahead, turning right or turning left.

Research Method

First, we created a starting population of snakes; Each snake received random values as the weights in its neural network. Then we repeatedly produced the next generation of brains using evaluation, crossover and mutation. This means that in each generation we measured the fitness of each brain. Next, we created new brains using the relatively fitter brains from the last generation. Finally, we slightly mutated the weights of the new brain.

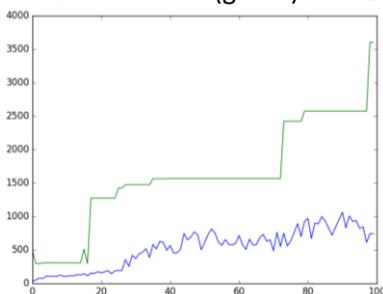
We used Python with the following packages:

- Numpy - Making matrix manipulations easily.
- Matplotlib – Displaying the gameboard’s state and plotting graphs.
- Pickle - Saving interesting populations of snakes.

Our code is divided into several classes. Snake – creates a snake object. Brain – the three-layer neural network. Evolution – contains functions which mimic the process of Natural Selection.

Results

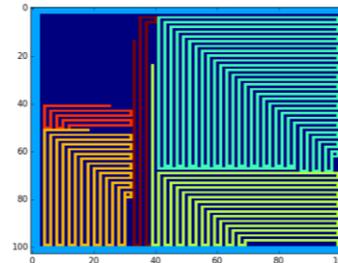
Training of most populations was successful. The next graph shows the average fitness (blue) and the maximum fitness (green) over 100 generations.



The graph indicates classic genetic algorithm behavior – a breakthrough every a few generations.

Supervisor: Shlomo Rozenfeld

We researched the effect of various parameters on the success of the snakes, and the different shapes formed by them. For each parameter, we ran the simulation with one snake, as well as multiple snakes playing at the same time. Snakes with square vision managed to learn at a higher percentage than snakes with rectangular vision. In addition, snakes with square vision created more complex shapes such as spirals or zigzags.



Five snakes forming zigzags

Furthermore, the size of the starting population had an influence on the learning speed and quality of the snakes. The larger the starting population was, the faster the snakes learned.

Unexpectedly, the larger the depth of the snakes’ vision was, the worse they performed. This is caused probably due to the overflow of information.

When trying different number of neurons in the hidden layer of the neural network, we didn’t find a clear correlation between the size of the hidden layer and the success of the snakes.

Conclusion

Overall, the snakes learned well how to avoid walls and themselves. Nevertheless, it seems the ability to avoid more advanced obstacles wasn’t developed.

Variation of the starting population was significant to the process of learning; In order to see learning, the starting population had to be large enough.

The type of the snakes’ vision also had an effect on the learning. Snakes with square vision learned faster and created more complex shapes because their vision was more consistent while turning.

On the other hand, we observed that increasing the size of the vision didn’t improve the results. In fact, doing this hindered the learning process.

Further Uses

This type of algorithm can be applied on robotic vacuum cleaners or lawn mowers.

Lastly, it could be interesting to see how the snakes will perform after adding memory to our model, i.e. passing the previous vision in addition to the current vision as input to the neural network.